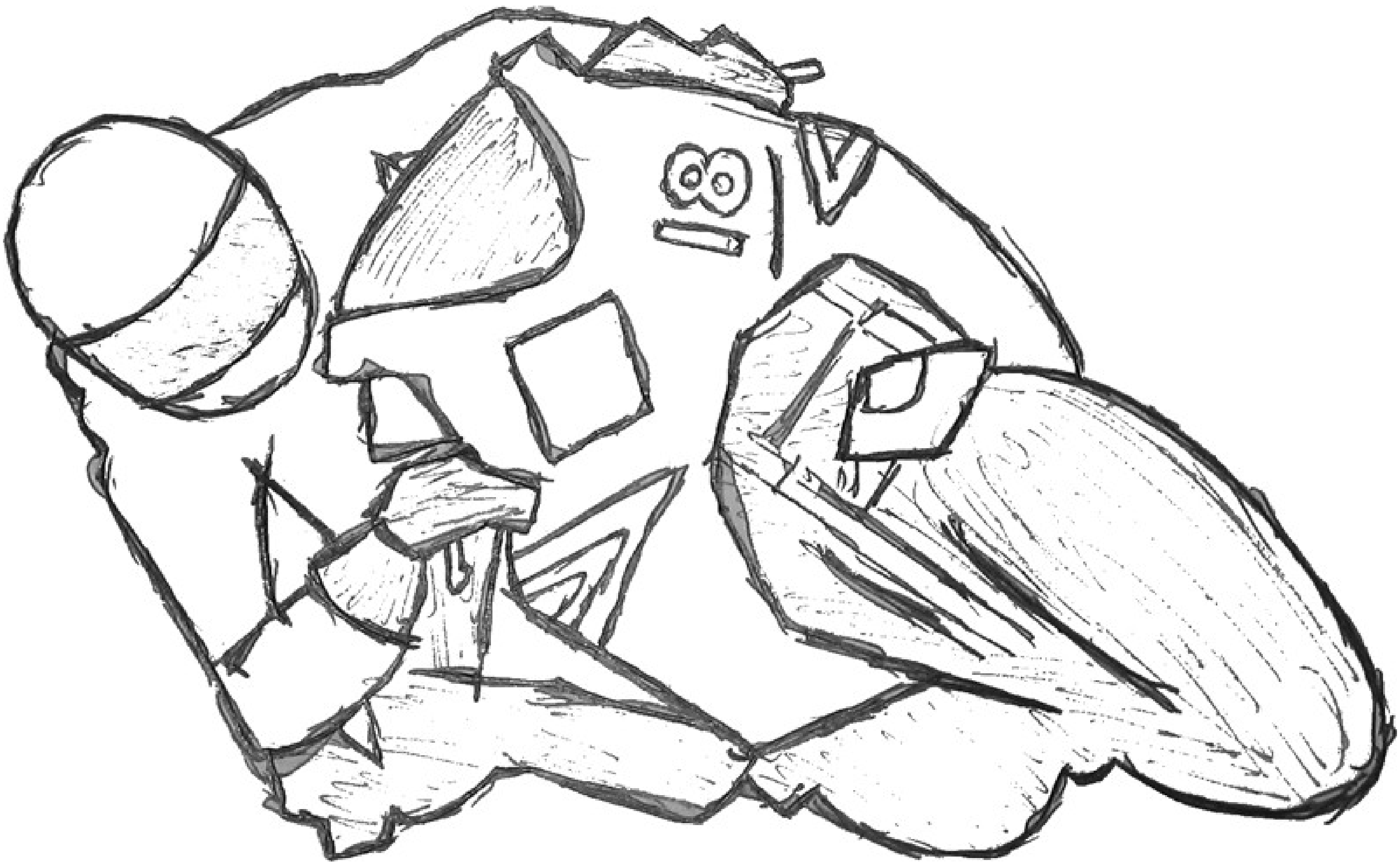
Motorcycle Data Acquisition System

(MODAQ)



MSC Mobile Computing

Masters Degree Program

Jonathan Guy Williams – WN210309

wn210309@student.staffs.ac.uk

A thesis submitted in partial fulfillment of the requirements of Staffordshire University for the degree of Master of Science

March 2009

Supervised by Mr David Thomas

Abstract

The purpose the project is to investigate a "theoretical" system solution enabling the acquisition of both locational and statistical data measurements from a high speed mobile object (*a motorcycle*) within a set boundary and environment (*a racetrack*) through the use of mobile and wireless technologies.

During motorcycle racing thousands of interesting measurements are produced throughout each lap of a race. The ability to acquire and analyse these measurements can be used to identify, improve and even solve any existing problematic areas.

Examples acquired measurements types include *Speed, Revs Per Minute (RMP), Throttle Position, Acceleration and Braking forces, Cornering Speeds and Forces, Fuel Level, Selected Gear and even Lean and Drift Angles...* Additionally a motorcycles *Locational and Longitude, Latitude and Altitude* measurements. All these measurements are "time" based in relation to the location of a motorcycle at a particular point (*section*) of a racetrack.

The proposed systems main objectives and deliverables are to investigate how such data measurements can be acquired from a motorcycle during a lap of a racetrack, and create a "theoretical" artifact detailing the complete **D**ata **A**c**Q**uisition (DAQ) process including the required and combined **H**ard**W**are (HW), **S**oft**W**are (SW), technologies and Methods as a complete Back-End DAQ system solution.
To enable this solution, relevant literatures will be reviewed with beneficial and required areas, systems and solutions being highlighted and used for further more in depth investigation.

The ability to viksualise and analyse the acquired data will be further investigated, and an "analytical" prototype application will be designed and implemented as a **P**roof **O**f **C**oncept (POC). The application will be used to plot and render the acquired data in a meaningful and understandable manner that can be used for further data analysis and factor identification.

The complete system, both Back-End (*theoretical*) and Front-End (*practical)* will be tested and evaluated accordingly using a relevant test plan(s), highlighting the projects overall success and achievements in relation to its set requirements.

The proposed project system could be used as a learning and teaching aid for riders, visually highlighting at what points they are fast and slow, with additional possibilities of comparing the data against "favorable" measurements highlighting where improvements can be made, and how they can be achieved.
Another potential the system use could be within racing teams to analyse both rider and machine performance identifying weak areas. For example highlighting a riders "riding" style (inputs and technique) or lines taken, as well as motorcycle setup issues, including when power delivery or speed is low, or even where noticeable and frequent drifts occur during cornering. Overall highlighting very valuable areas that could potentially "make or break" race results.

Example uses for the application are almost endless within the subject domain, with the overall system goal being to reduce lap times performed by both rider and machine, in the fastest and safest method possible.

After the project completion, and depending on the projects success, the established "theoretical" solution could taken further and developed at a "practical" level, that in conjunction with the Front-End Application providing a complete fully functioning Motorcycle DAQ system.

Table of Contents

Table of Figures

1 - Abbreviations

- (2WT) - **2**-**W**ay **T**elemetry
- (3D) – **3**-**D**imensional
- **A**
  - (ADE) - **A**pplication **D**evelopment **E**nvironment
  - (AS) - **A**ction-**S**cript
  - (ASF) - **A**dditional **S**econdary **F**actor
  - (ASW) - **A**pplication **S**oft**W**are
- **B**
  - (BCS) - **B**ritish **C**omputer **S**ociety
  - (BS) - **B**ase **S**tation
- **C**
  - (CBS) - **C**entral **B**ase **S**tation
  - (CEM) - **C**airngorm **E**vent **M**odel
  - (CP) - **C**onnector **P**anel
- **D**
  - (DAB) – **D**rain-**A**nd-**B**alance
  - (DAPS) - **D**ata **A**cquisition and **P**ositioning **S**ystem
  - (DAQ) - **D**ata **A**c**Q**uisition
  - (DAS) - **DA**Q **S**ystems
  - (DB) – **D**ata**B**ase
  - (DCGPS) **-** **D**ifferential **C**arrier-Phase **GPS**
  - (DDA) - **D**isability **D**iscrimination **A**ct
  - (DFP) - **D**ata-**F**low **P**rogramming
  - (DGPS) - **D**ifferential **G**lobal **P**ositioning **S**ystem
  - (DMA) - **D**ata **M**emory **A**ccess
  - (DoD) - **D**epartment **o**f Defense
  - (DOP) - **D**ilution **O**f **P**osition
  - (DP) - **D**ata **P**rovider
  - (DSRC) - **D**edicated **S**hort **R**ange **C**ommunication
  - (DSW) - **D**river **S**oft**W**are
- **E**
  - (EMS) - **E**ngine **M**anagement **S**ystems
  - (ERM) - **E**ntity **R**elationship **M**odels
- **F**
  - (F1) - **F**ormula **O**ne
  - (FP) - **F**ront (interface) **P**anel
- **G**
  - (GP) - **G**raphical **P**rogramming
  - (GPRS) - **G**eneral **P**acket **R**adio **S**ervice
  - (GPS) - **G**lobal **P**ositioning **S**ystems
  - (GPRS) - **G**eneral **P**acket **R**adio **S**ervice
  - (GSM) - **G**lobal **S**ystem for **M**obile Communications
  - (GUI) - **G**raphical **U**ser **I**nterface
  - (GUIDE) - **G**raphical **U**ser **I**nterface **D**evelopment **E**nvironment
- **H**
  - (HCI) - **H**uman **C**omputer **I**nteraction
  - (HD) - **H**ard **D**rive
  - (HTA) - **H**ierarchical **T**ask **A**nalysis
  - (HW) – **H**ard**W**are
- **I**
  - (IGS) - **I**nertial **G**uidance **S**ystems
  - (IMU) - **I**nertial **M**easurement **U**nit
  - (ISM) - **I**ndustrial, **S**cientific and **M**edical
- **J**
  - (JAD) - **J**oint **A**pplication **D**evelopment
- **L**
  - (LABVIEW) - **LAB**oratory **V**irtual **I**nstrumentation **E**ngineering **W**orkbench
  - (LBS) - **L**ocation **B**ased **S**ystems
  - (Loran) - **LO**ng **R**ange **A**id to **N**avigation
  - (LPM) - **L**ocal **P**osition **M**easurement
  - (LR) - **L**iterature **R**eview
  - (LTM) - **L**inear **T**ask **M**odel

- **M**
  - (MATLAB) - **MAT**rix **LAB**oratory
  - (MDI) - **M**ultiple **D**ocument **I**nterface
  - (MT) - **M**achine Mo**T**ion
  - (MVC) - **M**odel-**V**iew-**C**ontroller
- **N**
  - (NASCAR) - **N**ational **A**ssociation of **S**tock **C**ar **A**utomotive
  - (NAVSTAR-GPS) - **NAV**igation **S**ystem with **T**iming **A**nd **R**anging **G**lobal **P**ositioning **S**ystem
  - (NGWN) - **N**ext **G**eneration **W**ireless **N**etworks
  - (NI) - **N**ational **I**nstruments
  - (NIC) - **N**ational **I**nstruments **C**orporation
  - (NLTM) - **N**on-**L**inear **T**ask **M**odel
  - (NM) - **N**avigational **M**odel
- **O**
  - (ODB) - **O**bject-relational **D**ata**B**ase
  - (OOP) - **O**bject **O**riented **P**rogramming
  - (OS) - **O**perating **S**ystems
  - (OSSS) - **O**lympic **S**lalom **S**kiing LPM **S**ystem
- **P**
  - (POC) - **P**roof **O**f **C**oncept
  - (PP) - **P**roject **P**roposal
  - (PPS) - **P**recise **P**ositioning **S**ervice
  - (PS) - **P**roposed **S**ystem
  - (PSP) - **P**recise **S**ingle **P**ositioning
  - (PVA) - **P**osition/**V**elocity/**A**cceleration
- **R**
  - (RAD) - **R**apid **A**pplication **D**evelopment
  - (RIA) - **R**ich **I**nternet **A**pplications
  - (RPM) - **R**evs **P**er **M**inute
  - (RSI) - **R**epetitive **S**train **I**njury
  - (RT) - **R**ace **T**echnology
  - (RT) - **R**eference **T**ransponder
  - (RTC) - **R**eal **T**ime **C**lock
  - (RU) - **R**over **U**nit
- **S**
  - (SA) - **S**elective **A**vailability
  - (SatNav) - **Sat**ellite **Nav**igation
  - (SBS) - **S**ensor **B**ased **S**ystems
  - (SDLC) - **S**ystems **D**evelopment **L**ife-**C**ycle
  - (SNR) - **S**ignal to **N**oise **R**atio
  - (SNS) - **S**atellite **N**avigation **S**ystem
  - (SPS) - **S**tandard **P**ositioning **Si**gnal
  - (SSADM) - **S**tructured **S**ystems **A**nalysis and **D**esign **M**ethod
  - (STM) - **S**hort **T**erm **M**emory
  - (STUN) - **S**caleable **T**racking **U**sing **N**etworked **S**ensors
  - (SU) - **S**ensor **U**nit
  - (SW) - **S**oft**W**are
- **T**
  - (TBS) - **T**elemetry **B**ase **S**tation
  - (TDM) - **T**ime **D**ivision **M**ultiplexing
  - (TDR) - **T**eam **D**omination **R**acing
  - (TM) - **T**raq**M**ate
  - (TRIuMPh) - **T**ask **R**equirements formulation, **I**nteraction, **M**edia integration and **P**ractical implementation
- **U**
  - (UI) - **U**ser **I**nterface
  - (UTC) - **U**niversal **T**ime **C**oordinate
- **V**
  - (VDU) - **V**isual **D**isplay **U**nits
  - (VI) - **V**irtual **I**nstruments
- **W**
  - (WLAN) - **W**ireless **L**ocal **A**rea **N**etwork
  - (WNS) - **W**ireless **S**ensor **N**etwork

2 - Preamble Section

The aim of this project is to research into and design a prototype application in the area of motorcycle riding and racing using mobile and wireless technologies. The aim of the system is to be used for both teaching and learning aspects, improving rider skills (*technique, control, confidence and safety*), set within a racing environment (*race track*).

Reducing rider lap times is the overall aim of the system, fundamentally allowing a rider to complete a lap of a racetrack in both the safest and fastest method.
The system is intended to read, monitor and track individual laps covered by a single motorcycle and rider of a (*particular*) racetrack, logging the relevant and necessary factors and requirements that are needed. These can be further used to aid the described progression (*learning process/curve*) successfully achieving the expected results.

Both interest and participation in motorcycles and motorcycle racing has increased dramatically in recent years accompanied by the growth and technological advances within the mobile and wireless domain, continually offering more solutions to many problems, therefore, it is thought to combine the two areas relating and using them together to create a solution (*a system meeting the project requirements*).

The fundamental aims and objectives of the project are to generate both a good understanding and knowledge of the relevant development and subject area domains in order to enable a successful project completion. A detailed plan will be generated to aid the process, breaking down the overall project goal into smaller more manageable deliverables, piecing them together thus helping to meeting the final project goal. Along with the knowledge and understanding gained related to the best methods of tracking, collecting and storing data from high speed mobile objects, a prototype **U**ser **I**nterface (UI) is expected to be delivered demonstrating how such collected data collected can be used in terms of analysis needed during the improvement and learning process.
Above all a great learning enhancement within the mobile and wireless computing domain is not just s desired quality, but additionally an expected project outcome and deliverable.

The project plan can be found in **[Appendix 1]-(Summarised Project Plan)**

Due to the time-scale required to create such a system (*for a single developer meeting all system requirements, it is estimated that development of such a system would be anything up to and exceeding 1 year*) and the costs that would be incurred (*hardware, software and development time*) it has been decided that for this project, in depth research and investigation will be completed in all areas required for such a system, however only a prototype system will be provided using and running on test data using a selectively chosen UI areas (*screens*) demonstrating how such a system could work, painting a larger picture of such an applications potential and need in production.

The nature of the proposed system incurs certain challenging areas that need to be resolved to enable a successful project completion. Such challenges include both academic and intellectual challenges in terms of the identification and understanding of the best methods and technologies required to effectively and efficiently capture, track and collect the movements of a high speed mobile object, translating and communicating this data into a relevant format for storage, plus the additional challenge of developing a prototype UI design using a relevant programming environment, language and architecture pushed by the collected data (*acting as the applications data provider*) and displaying the data as both relevant and meaningful information that can be used for further analysis and meeting the application needs and requirements.

The project will be carried out in a professional and ethically correct manner following all relevant acts and guidelines related to the domains and subject areas throughout each stage of the development cycle. A more in depth study of the project is detailed in the Introduction Chapter.

3 - Introduction Chapter

Project Title: Motorcycle Data Acquisition System (MODAQ)
Project Area: Mobile Computing
Project Environment: Outdoor - Racetrack / circuit

3.1 - Project Introduction

The proposed system is primarily aimed at motorcycle riding and racing in the form of a learning assistant that can be used to aid both teaching and learning.
The system is intended to read and monitor individual laps of a racetrack covered by a single motorcycle and rider, logging and collecting relevant factors that can be used by the system to help the development (*teaching and learning*) progress.
To achieve the project goal, mobile technologies are not only thought to be the best suited, but also the only possible technologies (*within reason*) to both successfully and effectively achieve the required and relevant statistics and results.

Mobile and wireless technologies have seen a remarkable growth, advancing and evolving continually becoming more powerful and consistently achieving and maintaining higher levels, creating and improving solutions to various problems within many different domains. This growth is expected to continue during the foreseeable future, expecting additional improvements potentially presenting immense possibilities and results.

"*Mobile devices and wireless technologies seem destined to make a large and continuing impact on our lives. Mobile devices especially have been widely used in the last few years and recent developments in wireless technologies have provided new solutions to problems...***"** **[1]-(*J Chen et al, 2004*)**

The motorcycle interest, lifestyle and market in the United Kingdom (UK) are continually growing with more people showing interest within the domain each year. **"***There are almost 900,000 motorcycles on the road in the UK today. Growth has continued at approximately 7 to 10% since 1997. ...... The growth of the market from 2002 to 2006 is estimated to increase from £489m to £592m representing a 5% value increase for the future year on year.*" **[2]-(*Norwich Union, 2007*)**

This growing interest has also fueled interest in motorcycle racing with more "race-meet" tickets being sold each year and the addition of more race/riding schools opening across the country. **"***... to respond to the rising popularity and tremendous growth of motorcycle racing***"** **[3]-(*AmaPro racing, 2006*)**, with attendance records of racing schools rocketing with one of the largest and successful schools within the UK (*Handa's – Ron Haslam Race School*) recently celebrating **"***... the attendance of its 30,000th customer since being set-up back in 1996***"** **[4]-(*MotorbikeNews.co.uk, 2006*).**

As with most projects, this project is aimed at fulfilling a need. Projects tend to be undertaken to fill a gap in the market, or to advance/improve on areas within particular markets leading to deliberation in order to satisfy particular problems or a translated/perspective problematic situation. The problem situation for this project is described in the following section.

3.2 - Problem Situation

Due to the discussed growth and interest in motorcycles and motorcycle racing, more people are attracted to, and want to learn to both ride and race. Riders want to know how to ride well and get the full potential out of their machines with the best and safest environment in which to learn and practice those skills is on the racetrack.
Motorcycle racing is far from being a cheap sport or hobby, however, in comparison to other motor sports, motorcycle racing costs are reasonably moderate, this again a reason to appeal to more people. Along with the increasing interest and focus has come increased television coverage and sponsorship. This brings both exposure and revenue to the sport allowing it to blossom, presenting a very marketable area for such a system to exist being an (*always growing and obvious*) requirement for exponential riders, furthering a greater demand and marketable area for riding schools offering them the chance to achieve their full potential.

Combining both the advances in mobile technologies with the requirement for the enhancement and learning of motorcycle racing, the **P**roposed **S**ystem (PS) will fulfill the necessary brief of requirements resolving the discussed fundamental issues and in doing so provide a sufficient solution to the problem situation.

3.3 - Proposed Solution

At this stage of the process not all technologies or principles are completely understood as to which is the best to use/way to go as the proposed solution compromises of comprehensive ideas thought to be required for a complete and functioning system.
The primary theory at this stage of development is to install a mobile device such as a tracker on a moving object (*a motorcycle*), which in turn communicates with a **C**entral **B**ase **S**tation (CBS) using relevant conversing methods and technologies. For example sensors or satellites (*depending on which is best suited to the PS*) that can be used to send and receive (*transmit*) collected data from the mobile device to the CBS and fundamentally translate this data into relevant and meaningful information allowing it to be used for analysis and enabling the identification of areas for improvement.

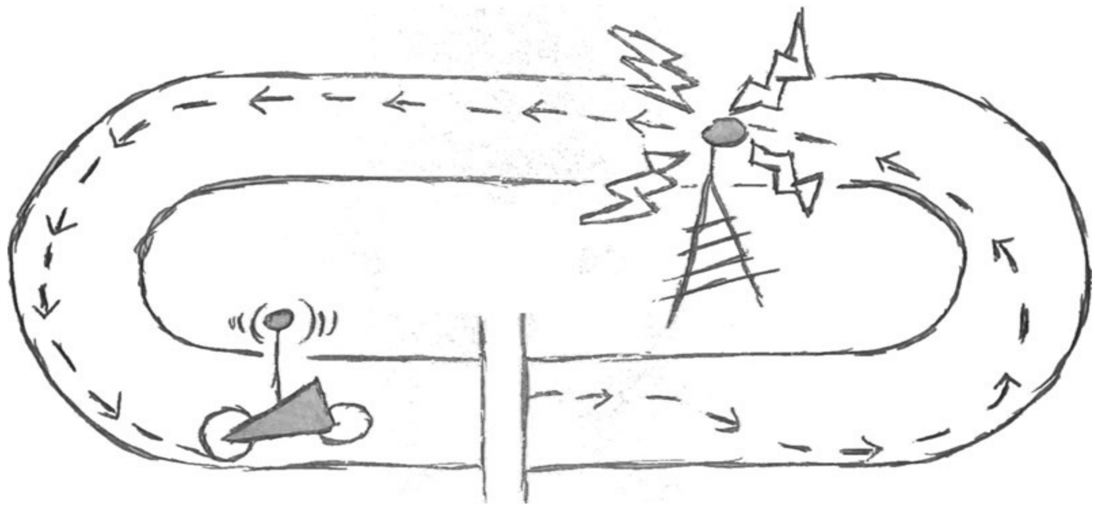
Such example data that could be collected and used for relevant measurements, features and analysis may include the following fundamental information:

- Overall racetrack lap times.
- Object positioning at different sections of a racetrack.
- Identification of racing lines used/taken during individual laps.
- Different speeds at different sections of a racetrack.
- Average Speed calculation.
- Fastest and slowest speed detection.
- Individual corner speed detection.
  - Speeds when entering corers.
  - Speeds maintained mid corner.
  - Speeds when exiting corners.
- Braking detection, including braking points and strength.
- Acceleration detection, including acceleration points and strength.

The above listed fundamental areas could of course be expanded upon and additional important and interesting factors could be calculated from these results. In cases with the correctly selected hardware (*and unlimited funds*) the possibilities are almost endless, mainly due to the discussed technological advances within the area. For example, devices can be connected to **E**ngine **M**anagement **S**ystems (EMS) and can send machine information (***R**evs **P**er **M**inute (RPM), throttle, brakes and clutch positions, currently selected gears and fuel consumption to name a few*), packeting this data and transmitting it to a CBS where it could potentially be synchronised and translated into further analysis information.

currently there are factors in such a system that are either obvious or have already been established. It is important that these be documented and used within further research as they may cause challenging or problematic hinderance to further project progression, including:

- The system will be operational in an outdoor environment.
- The system will (*potentially*) have a centralised architectural structure.
- A high speed object (*motorcycle*) movements are required to be tracked.
- A small lightweight device is required to be installed on the mobile object (*motorcycle*) that will not affect handling or performance in any way.



[Figure 1]-(*Proposed prototype system architecture*)

Once collected and communicated, the data is intended to be stored relevantly in (*as currently believed*) a **D**ata**B**ase (DB) allowing it to be universally available and useable. Once the data has been stored, a large and challenging factor of the system is believed to have been accomplished.
The data can be then pumped into a usable GUI where it can be converted and displayed as understandable (*meaningful*) information visually showing a performed lap(s) maybe even using simulations as an additional feature and learning aid. This enables the collected data to be used for lap analysis and furthermore highlights areas where room for improvement exists, overall and principally reducing lap times and improving riding in general.

3.4 - Project Background

The tracking of a mobile object is a very large and vast subject that has played and proved a primary objective within the mobile and wireless domain from its existence, growing and evolving with technologies advancing to the high standards and achievements of current object tracking today, with impressive foreseeable enhancements for the future.

There has been a reasonable amount of work, research and projects carried out within the same/similar domain as the PS illustrating successful results. Although in many cases this work is not always in the same area, it is however believed relevant for discussion.

Tracking mobile objects does not just relate to the tracking of a motorcycle (*or any motorised vehicle*) around a racetrack, it can relate to anything that moves within any environment in which the movement of the object is required to be tracked for whatever reason(s).
As mentioned, this is a large subject area resulting in the existence of many types of both previous and current work within the area. Some examples include:

- The tracking of civilians (*soldiers*) working within and to aid military operations.
- The tracking of animals to aid in biology research.
- The common tracking of vehicles within **Sat**ellite **Nav**igation (**SatNav**) systems.

These examples show the wide range of environments and work within the subject areas detailing the usages of how the technologies can be used within the different environmental settings - from urban/rural areas to built up and busy areas. This also encompasses different scales ranging from a large/wide area to much smaller ones.

Work within more relevant areas to the PS include systems with similar foundations to those believed necessary regarding the PS, thus proving the project to be so far traveling in the correct direction.
Vehicle tracking within motor racing which is mainly and fundamentally currently performed within the **F**ormula **O**ne (**F1**) domain, however is additionally used within other popular motor racing domains including motorcycle racing, although to a much less extent.
Telematics along with relevant **G**lobal **P**ositioning **S**ystems (GPS) and **G**lobal **S**ystem for **M**obile Communications (GSM)/**G**eneral **P**acket **R**adio **S**ervice (GPRS) technologies are used to track and log a vehicles dynamics, in terms of monitoring their location, movements, status and behaviors. This information is generally collected by equipping the vehicle with a GPS receiver used alongside an electronic modem (GSM/GPRS) thus offering accurate object positioning and location information. This

then enables communication between both the vehicle and the software in use.
The collected data is sent to a CBS where it is then translated into relevant information using specialist tools in conjunction with visual display and mapping software and can therefore be used for further analysis.

Although similar systems exist using the same fundamental ideas and technologies which are planned to be utilized and necessary for the development of the PS, including the described (*motor racing*) monitoring system, no system has been found that is designed and aimed primarily as a "learning assistant" that can be used to aid both teaching and learning within motor (*motorcycle*) racing. Similar set ups exist, but are designed and used for different purposes thus creating both a great advantageous potential and marketable domain for the PS.

Relevant technologies (*including and additional to those discussed*) are covered and documented accordingly in a literature review, identifying new areas for investigation that may aid further project progression. Key areas were highlighted and identified during the literature search with some being more relevant than others. These include areas that were previously not thought necessary for consideration, with some additional areas not yet discovered.
After further review into selected areas and supplementary areas highlighted by authors, it was made apparent that for a successful development additional directions of research need to be accomplished and other domains discussed and identified by the different persons (*authors*) and schools within the relevant literatures. The Literature review can be found in the next chapter of this paper.

3.5 - Aims and Objectives

The key goal of this project is to successfully research, analyse, design, implement (*parts of*), test and evaluate a **prototype** system that can be used to aid both teaching and learning in the motorcycle racing domain, enhancing rider skills and fundamentally reducing lap times.

The fundamental aim of this project is to carry out relevant research into the pertinent areas that will affect project development. There are existing areas that are (*at this stage*) as yet not known or understood. They therefore require more in depth research and investigation to obtain a better understanding of. Additional areas exist that could prove problematic and challenging, thus again requiring further research and investigation to either solve or find different possibilities or technologies that can be used as a solution.
Both the knowledge and understandings obtained during this process will be used further to design and implement a prototype application providing an artifact that can be tested and evaluated accordingly.
A relevant methodology will be selected and used throughout the development cycle, guiding and pointing the development in the correct direction. Additionally, existing applications, (*previous and current*) will be reviewed and analysed, and along with the knowledge acquired from the undertaken research, a detailed projects design will be created.
After the system has been designed, implementation will commence using the relevant technologies, programming languages, architectures and environments discovered during the research and investigation process followed by the relevant and necessary testing and evaluation procedures.
At the end of the development cycle a critical evaluation of the project as a whole will be completed discussing the projects overall success rate in regards of meeting these discussed aims and objectives, along with any additional successes and problems encountered during the process.

The following list of objectives highlights the foundational project aims and objectives, and has been developed in relation to the described working plan and deliverables. The project aims and objectives include:

- A detailed project plan outlining the project schedule and development stages/life-cycle.
- In depth research and investigation into all necessary areas concerned.
- The identification of possible problematic and challenging areas.
- Solutions to these identified problematic and challenging areas.
- A detailed design specification.
- A successful implementation stage and guideline creation.
- A relevant and necessary testing and evaluation process.
- A detailed critical evaluation of the project as a whole.
- The completion of the project in both an ethical and professional manner.
- A great and valuable all round learning experience.

3.6 - Project Deliverables

A list of primary deliverables have been created which (*at this stage of development*) are expected to be completed by the end of the development cycle, and if not justified accordingly. This list can be used for project evaluation of the overall project success rate in terms of the final project outcome and meeting these set project output requirements:

- A successful research programme gaining a good and knowledgeable understanding of relevant research areas.
- A detailed project report documenting well each individual part and stage of the project.
- A detailed design specification of the PS.
- A set of detailed guidelines for guidance concerning hardware and technological elements regarding the collection of data from a mobile object (*motorcycle*) in relation to a specific environment (*racetrack*).
- A successful implementation producing a functional prototype (*front-end*) application, using (*test*) data from a relevant database.
- A detailed test plan and evaluation of the test outcome and results.
- a detailed critical evaluation of the overall project as a whole.
- A great learning enhancement within systems development.
- A great learning enhancement within the mobile and wireless computing domain.

It is described that "test" data will be used for the PS. This is due to many reasons, and although not perfect is believed to be more than enough to fulfill the requirements and aims of this project. Moreover, the costs and time schedule required to collect such "real" data is deemed not necessary for this project, however is a great ambition in regards to further project progression and expansion.
Enabling the use of "real" data capture would require the use of a track prepared motorcycle, a relevant environment setting (*a racetrack*) along with all the hardware, software and technologies required for such data capture, all requiring a large development and support team being both too expensive and irrelevant (*not required*) at this stage of a prototype (*alpha*) project cycle.

3.7 - Project Management

Project management generally consists of the appliance of knowledge, skills, tools and techniques that are planned for use when project requirements and expectations are to be strictly and successfully met. The success of project management can be measured, and is reflected in the final outcome of the project, in relation to the project meeting its (*documented)* requirement and functionality standards both within cost and time scales **[86]-(J Marchewka, 2003)**.

Project management, therefore, generally takes the form of using a project goal (*benchmark*) and breaking this goal down into smaller, more manageable tasks and timescales thus separating and aiding the completion of the overall project.
It is described that when considering project management steps, a project can be separated and defined using the following characteristics:

- There is a set start and end point.
- There is a well defined object (*artifact*).
- The challenge is unique and not repetitious.
- The project contains cost and time schedules to produce a specific result.
- The project cuts across many functional boundaries .

In relation to this project, a strict project management is believed unsuitable for the completion of the PS due to both the time scale and situation that the project will be completed within. It is believed that a more laid-back and "relaxed" approach would be more beneficial, however still being strict enough to ensure that the project deadline is successfully met.
At a personal level, project development is preferred (*found more interesting*) when elements and sections can be switched between (*intercrossed*) thus making strict deadlines and milestones set within specific project areas both difficult and problematic in relation to preferred developer practices.

Although relaxed, the project management for the PS will be taken very seriously and therefore the project will be broken down into separate (*more distinct*) areas, and a project plan will be created (*although relaxed*) and referred to throughout the project development cycle to ensure its completion.

The plan will illustrate illustrate the project has been broken down and grouped into relevant tasks and sub tasks (*domains*), and the different time scales that have been allocated to each area, thus allowing the project to be completed within the set timescale and meet the set deadline.

The project plan can be found in **[Appendix 1]-(Summarised Project Plan)**

3.8 - Development Methodology

A definition of the term "methodology" is "... a collection of philosophies, phases, procedures, rules, tools, techniques, documentation, management and training for developers of systems" **[88]-(S Standings, 2000).**

Most system development methodologies follow a **S**ystems **D**evelopment **L**ife-**C**ycle (SDLC), which is a conceptual model that is used to describe the phases involved within system development.
SDLC is the process of developing applications following and investigation, analysis, design, implementation and maintenance cycle.

A variety of SDLC's have been created to guide development processes, with the most common being the Waterfall Model, and other including **R**apid **A**pplication **D**evelopment (RAD), **J**oint **A**pplication **D**evelopment (JAD) and the spiral model.

Different methodologies are better for different projects (*depending on the project and the developer/team working on a project*), and in some cases it is preferred to combine several methods thus creating a "hybrid" methodology, "*... but the most important factor for the success of a project may be how close the particular plan ...* " (*method*) "*... was followed*" **[87]-(SearchVB, 2004).**

Methodologies break systems development into manageable and understandable phases, thus aiding for a successful application development that meets system requirements such as deadlines, costs and system requirements.
Methodologies generally cover the following system development aspects:

- Break a project down into different phases
- Define the tasks to be carried out at each phase
- Define the required outputs of each phase
- Define the actions/events to be carried out at each phase
- Define what constraints are to be applied at each phase
- Define what tools and techniques will used during development

It is understood that following a methodology or set guidelines during the development of the PS is recommended, and will help the development process to cover all important development areas further preventing system errors or even failures that could occur at a later stage, being both costly and time consuming to correct, and for this reason a "basic" system development process will be followed, however it is believed irrelevant (*too restrictive*) to follow a strict set of guidelines for the PS, as due to its (*experimental*) nature and developer preferences a flexible development process is preferred.

Each system has its own individual requirements, and it can be hard to place a distinct development into a definition of any methodology. Different methodologies offer different methods, tools and techniques of dealing with the same problem, therefore selected and known (*favoured*) methodologies have been investigated (*including **R**apid **A**pplication **D**evelopment (RAD), **S**tructured **S**ystems **A**nalysis and **D**esign **M**ethod (SSADM) and **T**ask **R**equirements formulation, **I**nteraction, **M**edia integration and **P**ractical implementation (TRIuMPh)*) in relation to the PS development, and from the three selected (*and even additional methods*) a selection phases deemed relevant and beneficial will be used and documented.

In summary, no "particular" methodology will be used with the development of the PS, as due to the discussed development situation, a more relaxed, free and less strict and structured working process/environment is both preferred and believed to be more productive. Thus a hybrid methodology will be followed taking those most thought necessary development areas from the selected methodologies, glueing them together thus keeping a general structure and ensuring a successful completion although remaining comfortable with a structure that is (*as mentioned*) preferred and thought best suited for a successful project completion.

3.9 - Research Approach

Research is used to investigate the relevant areas that need to be clearly understood aiding in the completion of a successful project, thus playing an important role within the development cycle of any project , "... *Good research can certainly solve large problems but is more likely to shed light on the problematic area as a whole* ..." **[5]-(*Oliver, 1997*)**, therefore research into all relevant areas of the project will be completed gaining a good understanding within each and what is required for a successful development.

There are two main types of research considered relevant for discussion and believed should be used as will aid within the research and investigation of the project:

- **Primary Research** - is designed to generate and collect data for a specific problem area. The data that is collected does not exist prior to the data collection.
- **Secondary research -** is data that is originally collected for a purpose other than current research objectives, therefore as it is revisited and is put to a second use (*surveys and questionnaires*)

[6]-(*McGivern, 2003*)

Regarding the PS, research will be focussed around the subject area of: "tracking the movement of a high speed mobile object". All forms of literature are thought and believed to be relevant and therefore will be included in terms of books, journals, papers and web-sites.

The main (*summarised*) areas that are (*at this stage*) assumed the most relevant areas for further and more in depth research and investigation in regards to the PS project include:

- Development cycle/relevant development methodologies.
- Mobile and wireless computing in general.
- Mobile and wireless networks.
- Mobile and wireless devices.
- Mobile and wireless technologies.
- The tracking of mobile objects.

3.10 - Resources

The resources required to conduct a successful research program enabling project completion include:

- Access to all relevant literatures (*research, design, implementation and analysis*).
- Books and journals (*library access*).
- Online resources, journals and websites (*Internet access*).
- User preferences and usability:
    - User profiling, interviewing and other secondary research methods.
    - Access to relevant development tools (*hardware and software*).
    - Database and programming environments, languages and architectures.
    - Possible access to relevant hardware equipment:
        - Computers, tracker (*if available*), relevant testing materials.

Most of these recourses (*mentioned above*) will and should be freely and widely available and will aid greatly in and throughout the project development. Some areas (*as discussed*) may not be available due to various and existing constraints, however at this level these unavailable resources are believed not to be pertinent to the success of the project. Full advantage will be taken of all those available resources whilst taking account of any discrepancies arising.

3.11 - Ethical Considerations

This project will be carried out and completed in both and ethical and professional manner, with ethical considerations being included within each and every areas and process of the project:

Ethical guidelines will be followed throughout each stage of the project, providing and allowing for professional development. The project will follow the **B**ritish **C**omputer **S**ociety (BCS) code of conduct and other data protection acts that are relevant to the project completion. Development will be completed using a moral and correct method informing all parties and persons involved in the required manner. Parties and persons will be further informed of the reasons of the project before asking for their participation (*if any*) within the project.
Relevant accessibility issues will be taken on board following the **D**isability **D**iscrimination **A**ct (DDA)

taking into account any registered users with disabilities during the development cycle.

3.12 - Intellectual Challenge

The PS proposes many challenges and problematic areas in terms of intellectual, commercial and academic input. The main challenge of the project at this stage is considered to be the identification and understanding of the best, most effective and significant method(s) and technologies required to track the movements of a mobile object effectively and efficiently within the relative context and environments such as the PS, relaying reliable results back to a CBS which is considered the second key and challenging area within the project.

A major and fundamental factor regarding the system and creating possible difficulties plus additional challenges already existing, that the mobile object required to be tracked is a motorcycle racing around a racetrack. This object will be traveling at high speeds making accurate and precise tracking of the vehicle harder than if the object was traveling at a slower pace.

When and if solutions to the discussed areas have been established, the main challenge of the project is considered to be complete and accomplished, although there are additional areas that may still need to be addressed thus creating additional challenging (*although not as tormenting*) areas in terms of the design and development of the GUI (*the front end interface between the user and collected data*) that is used to convert the collected data into useable information. This will in addition (*aiding of the teaching and learning process*) enable the analysis of the collected and stored data, through the use of an appropriate DB further using a relevant programming language and environment to develop a front-end application so that the received and pushed data can be displayed in a meaningful and suitable manner.

4 - Literature Review

4.1 - Review Strategy

To enable the completion of this literature review, a vast literature search has been performed resulting in an overwhelming amount of both relevant and not so relevant materials within the subject area being found, compared, studied, analysed and evaluated to a high level of depth and understanding. This can and will be used to direct and aid the development of this project.

This chapter will review and discuss the discovered literatures regarding existing (*past and current*) methods, projects, systems and technologies (*elements*) in both industry and academia that have either been used, are currently used or could be used with systems when tracking mobile objects, additionally touching on some intended future research works and ideas.
The findings of the review are to be used to enhance the context of the project identifying and opening available options for further progression thus creating additional foundations to be built upon enabling new perspectives and additions within this area.

A large amount of literatures have been covered, however not all are completely relevant (*or not as relevant or useful compared to others*), therefore a careful screening process eliminating some (*both irrelevant and less relevant*) sources has been completed.
Despite the fact some literatures have been eliminated, the time spent reading and investigating these literatures is not deemed as "time-wasted" (*although not completely relevant to this project*) as a lot of new and interesting information has been uncovered and a great amount has been learned during this process.
The material that (*on the other hand*) is deemed relevant is reviewed within this chapter discussing any insights and ideas that can be further developed and thus enhancing the project further.

The review will consist of three "discrete" phases continuing from one another. The first phase will introduce the review discussing in general the literatures found as well as any problematic areas that have been established (*may hinder further project research and progression*). This will be followed by a more in depth discussion of the literatures and the different views and learning curves gained from these literatures. Finally a summarisation of the areas covered will be discussed hilighting the technologies that are potentially best thought relevant for inclusion within further research, and possibly usage at this elementary stage of the development cycle. These areas will be broken down into separated sections relevant to the literature(s) being discussed.

During the literature search the steps and directions taken were documented regarding when and how (*from what source*) the literatures were found and as a result how further literatures were found, resulting in a hierarchical tree styled structure.
The order of this review will differ slightly from the conventional literature review as it is thought best understood (*in this case*) if the review is to follow a similar root to that taken to that of the literature search itself. The arrangement will therefore follow the order of not only relevance but additionally the documented structure, thus creating a flowing chapter moving swiftly onto and between the different literatures and topics.

The literature search was completed with a "wide-mind" keeping all options and possibilities open. The search was not performed in a way that would limit literatures to existing and similar systems (*which turns out to be few and far between*), but also to literatures believed to be (*possibly*) required or could be used to improve/work alongside such a system, as well as literatures discussing relevant tools and technologies. Theses were included in the search and resulted in many literatures being found and identified.

To gain a better understanding and feeling for the reviewed literatures and to obtain better results it is believed best to compare and summarise the literatures together rather than by separating and evaluating them one-at-a-time. The review will additionally select a few literatures that "stand-out" in terms of general usefulness and relevance. This will facilitate the discussion and evaluation of these literatures further in relation to the **P**roposed **S**ystem (PS).

4.2 - Review Introduction

During the literature search many different *elements* have been uncovered and selected for discussion. These elements are extensively analysed, discussed and critiqued during this review exposing the different tools and techniques required/used identifying and justifying any similarities (*common occurrences*) or differences (*inconstancies*) between them. The review will freely discuss the selected literatures comparing and contrasting those relevant to one another, together with personal thoughts and opinions in an attempt to establish the level of scope and depth of any existing technologies and works that have already been undertaken within the subject area, additionally hilighting the level to which it has so far been accomplished.

Elements being discussed within the review that are thought to have a unique relevancy/compatibility with the subject area are separated into groups where the literature(s) appropriate to these elements are further individually discussed. Many Technologies are thought relevant to this project and many literatures have been covered regarding those technologies of which the technology GPS stands out above all as being the most popular and widely used, however GPS is a vast area and therefore is an example of an element that requires individual separation.
In-particular a system called “Race-FX” has been discovered and a contact with one of the head developers and literature authors of this system has been established during the literature search. literatures regarding this system as well as other similar systems that exist will also be be discussed and compared within the review.

The knowledge and ideas gained from the review are considered essential to the progression of the project as is it will predict the foundations for the next chapter (*Research Chapter*) and thus the generation of what is required (*in terms of steps*) for the completion of a prototype system.

As discussed in the Project Proposal the *PS* is constructed of many different domains that require thorough and in depth research and investigation requiring many literatures to be reviewed that are relevant to the required domain(*s*) as well as any additional and new domains that were found during the literature. The thoughts and opinions of additional authors and researchers within the subject area were gained and established.

As to what is currently believed, the fundamental and most problematic area within this project will be the tracking of a high speed mobile object (*motorcycle*) within a set/restricted boundary (*racetrack*). Additional factors and challenges that need to be addressed also include an outdoor environment, a centralised architecture, high-speed object/movement tracking and the need for a lightweight tracking device that will not affect the performance and handling of a motorcycle after installation.

There are many studies and published literatures that focus on tracking mobile objects that are thought relevant, therefore these studies will be searched, investigated, reviewed and evaluated thoroughly and accordingly.
The technologies and resources that are possibly required to enable the development of the *PS* will be covered gaining and good substantial understanding and knowledge of established theories, latest trends and what is fundamental to fulfill the requirements for a successful development.

After reading a vast amount of literature with and around the subject area highlighting solutions to other relevant and problematic domains, it became apparent that there are many directions required to be taken and investigated.
A lot of literatures found contained the same or very similar information which although useful in justifying one another and helping to clarify the reliability of the content, additionally tended to cancel one another out (contradict), thus making possible (*enabling*) the use of the most dominant literature(s).

In a lot of cases the literatures were not really relevant or even helpful, producing (*sometimes*) a very tedious information seeking process, consisting of scanning through different literatures and identifying between both useful and useless articles in order to select the most “thought” relevant for further research progression.
Literatures not entirely applicable to the subject, but sharing similar concepts were used, providing both very useful and helpful understandings of various factors required for development. For example, a mobile object can be anything (*not just a motorcycle or even a motorised vehicle*). Literatures were found regarding tracking moving civilians for military projects or tracking moving animals to aid in Biology research, and although deferring from the project domain are still very relevant sharing a lot of fundamental foundations.

As discussed, the fundamental and key area of research within this project is the tracking and storing of a position/location of a fast moving object. A good starting point therefore would be with the sampling of a mobile object generating real data that can be translated into meaningful information and used for analysis - "Data Acquisition".

Data **A**c**Q**uisition (DAQ) is a very wide subject covering many areas, some of which will not be relevant to this project. Many literatures regarding DAQ have been eliminated in this PS as thought not relevant enough to be included. The focus of the search was therefore concentrated on DAQ where the data is coming distinctly from a mobile object, and it was both immediately and evidently clear that in such cases GPS is the favored technology of use, as almost all literatures concentrated on using GPS with DAQ.

4.3 - Data Acquisition

Data acquisition is the sampling of the "real-world" producing data that can be translated and used by a computer based system for many differing reasons. **[7]-(*C Spencer et al, 1990*)** describes **DA**Q **S**ystems (DAS) as a system that is used to measure and log data parameters for analysis. Additionally the describe how DAS must consist of both hardware (*sensors, cables, and components*) and software (*the data acquisition logic and analysis software*) being used to achieve results.
There are many different examples within differing literatures of DAQ systems, one of which by **[8]-(*S McBeath, 2002*)** includes a DAS being installed on a race car which is used to measure the cars **R**evs-**P**er-**M**inute (RPM) and speed. This can be (*later*) viewed when the car has returned to the pits (*pit lane*) allowing the cars behaviour to be analysed and thus enabling the improvement of the cars setup (*improving performance*) where appropriate.
The example described by **[8]-(*S McBeath, 2002*)** is (*although different*) along the same lines of the *PS* showing that there are current DAQ techniques being used that could be adopted for use with the *PS*.

All literatures summarise and confirm the basic DAQ process in a similar way. DAQ requires the physical property of an object that is to be measured (*a mobile device installed on a motorcycle*). This property is converted into a relevant measurable (*electrical*) signal using a device called a transducer working in correspondence with DAS sensors. "*The ability of a DAS to measure different phenomena depends on the transducers to convert the physical phenomena into measurable signals (either digital or analogue depending on the transducer being used) by the DAQ hardware*" **[8]-(*S McBeath, 2002*)**. A personal worry of the *PS* that has been justified within this literature is signal interference. Within a racetrack environment, interference of signals between the mobile object and a CBS can be caused by many factors including other mobile objects, buildings (*stadiums*) and even weather conditions. Although the DAQ literatures do not discuss a solution to signal interference, they do however discuss the protection and amplification of signals (*known as Signal Conditioning*). If a transducers signal has been interfered with or is for whatever reason not strong enough (*or in some "rare" cases too strong*) for usage, a signal conditioning action may be required. The selected literatures discuss signal conditioning to a high level of detail including:

- Signal amplification.
- Signal de-amplification.
- Signal filtering.
- Signal demodulation.
- Bridge Completion.

At this stage it is enough that the literatures both confirm personal worries and more importantly each other, thus declaring that there will be a need for signal conditioning within the *PS*, and identifying a key area for further research.

DAQ appears (*as expected*) very pertinent for usage in and around the *PS*, however as mentioned within the reviewed literatures, various levels and standards of both hardware and software that have been developed "specifically" towards the use of a "specific" system are required for a successful result. Many literatures divert here and start describing hardware (*such as cables between external hardware and the CBS*) which is irrelevant for a mobile setup and thus is not included within this review. However, it has identified a key area that does require further research and investigation as to what is the best hardware to use for DAQ with a mobile system architecture.

During the literature search an interesting and relevant DAQ paper from **[9]-(National Instruments *Corporation*, 2006)** was found regarding a GPS DAS using a mobile architecture titled "*GPS Synchronization Architecture for Data Acquisition Devices*".
The paper discusses how GPS synchronisation can be used to synchronise measurements over (*extremely*) large areas without the need for cables between systems allowing a mobile architecture to be used. The paper continues to describe in detail the hardware and software that the DAS uses providing a detailed overview of how to set up a GPS synchronised system using DAQ devices. This is therefore believed relevant for further review.

4.3.1 - DAQ Devices and Architectures

National **I**nstruments **C**orporation (*NIC*) have created a DAS and published a paper regarding the system. The literature has been reviewed and has proved very useful in terms of enhancing further research areas and directions **[9]-(*National Instruments Corporation, 2006*).**

NIC explains that using GPS provides the benefit that data is always time stamped (*to a global standard*) allowing the association of data sets from and with other systems that are also GPS synchronised. Some other successful and existing systems found that take advantage of and use such an architecture include:

- Structural monitoring.
- Electrical power grid monitoring.
- Ground vibration monitoring.

Although different to the *PS*, the foundations that these systems are built upon "could" be very similar if such a DAQ architecture is deemed best to be used.
For the synchronisation of DAQ devices it is important that they sample data at the same time, assigning the same external sample clock to the devices that perform the sampling. However, when devices are not (*physically*) connected, a global reference is required from which the local sampling clocks can be derived. Such a global reference can be (*as is used with the NIC system*) GPS.

NIC further describes that "*a GPS disciplined clock is continuously adjusted to stay synchronised to the GPS signal so it does not continuously drift as a free running oscillator would*". GPS synchronisation reliability is praised to a very high level in both this paper and other reviewed literatures. Most literatures reviewed regarding time synchronisation systems "shout" about how the systems offer such reliable time synchronisation through the use of GPS. **[10]-(P Mann et al, 2004)** states that "*GPS is an excellent candidate for transporting a reference clock because each subsystem can have a direct connection to the same reference clock instead of an indirect connection over an unreliable network*". Additionally it describes that the locking onto of a GPS signal provides a very reliable reference clock (*reliable enough that most of the mobile DAQ system literatures reviewed use*), however, it also states that GPS is not always 100% reliable as "perhaps" a subsystem cannot find enough satellites or environmental conditions could make the signal difficult to lock onto. This is justified in a paper by **[18]-(Doble Engineering, 2007)** regarding universal time synchronisers, stating "*GPS-synchronised end-to-end testing has long been regarded as the ultimate method for efficiency and performance...*" however "*... because of physical obstructions or remoteness, GPS satellite reception is not always possible*" this confirms such DAQ problematic areas described previously by **[8]-(*S McBeath, 2002*).** NIC however does contradict itself in another of the company papers stating *(although again praising the technologies reliability) that "... with a cabled synchronised solution, you can achieve the tightest accuracy possible ..."* whereas "*... with a GPS synchronised scheme, the timing accuracy is reduced due to the error in the GPS signals..."* **[11]-(*National Instruments Corporation, 2007*).**

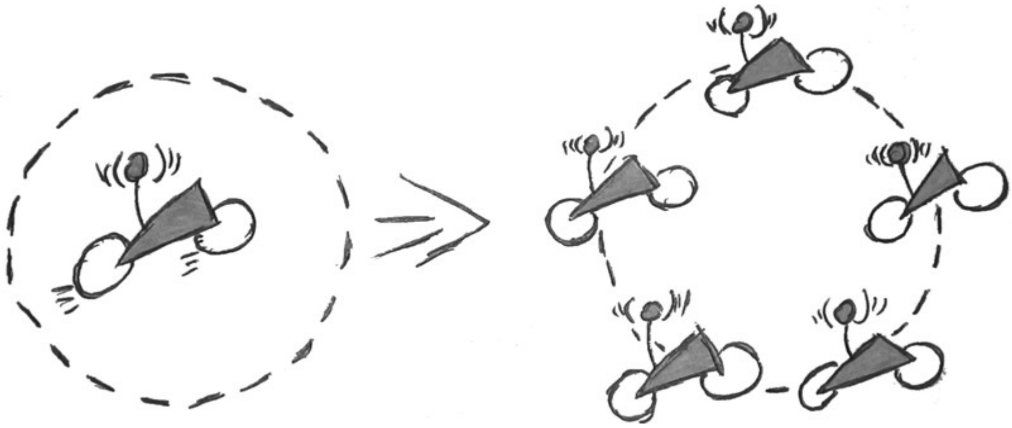
The above statement is not considered a problem or deemed to be of any concern or worry within the *PS* as it is already known and understood that a "wireless" solution would not be as reliable as a "wired" solution. However, a greater concern is of the contradiction between two papers from the same *(although different authors*) source/company.

It is believed that for the *PS* such a high level of reliability is not required. The reviewed papers have again highlighted a problematic area that will require further investigation regarding a decision as to the risk factor of this area, and/or finding a solution e.g. a different (*wireless*) technology that is not GPS based or related.

NIC have developed and use a device (*NI6682*) which they describe in a paper. It explains that it is able "*..to create clock signals which are based on a time reference like GPS*" using additional hardware (PXI) tto enhance the reliably of clock synchronisation, thus ensuring that there is a "minimal" skew on any clock signals allowing DAQ devices to be more tightly synchronised.

The NIC paper describes how this system requires the ability to communicate with each of its sub-systems over "TCP/IP". Cables are not required for the GPS based architecture thus "*if the systems cannot be physically tied into a network communications system, a **W**ireless **L**ocal **A**rea **N**etwork (WLAN) could be used*".
The *PS* will not require such network communication in its "prototype" stage, however this opens another area that can be further researched regarding the projects future expansion possibilities.

[Figure 2] shows how an example GPS architecture could be easily expanded:



[Figure 2]-(Possible E*xpansion of a GPS synchronised measurement system*)

The paper describes how GPS is a very flexible architecture allowing the used nodes to be different. For example "*... in many applications the sensors are not equally distributed over a test area*" thus being relevant to the *PS* in terms of flexibility regarding its use with different racetracks.
Flexibility is a great advantage that GPS offers DAS's. A paper about "Modular GPS Architecture" by **[12]-(N Gerein et al, 2006)** confirms this discussing the advantages of such an architecture including its flexibility and the adaptability it offers as a result. If a location requires many channels, multiple devices "*... can be synchronised together through a cabled solution while the entire node is synchronised to GPS*". Such a hybrid cabled and GPS synchronised solution is although relevant and interesting, a further review in this direction would be a deviation from the goal/aim of the system at this prototype stage.

[9]-(*National Instruments Corporation, 2006*)

4.3.2 - Data Acquisition Software (DAQ SW)

The NIC paper discusses a lot regarding **H**ard**W**are (HW) in terms of possible solutions and architecture variations, but does not touch so much on the **S**oft**W**are (SW) and **G**raphical **U**ser **I**nterface (GUI) that may be required.
Another interesting and informative paper from NIC detailing "Data Acquisition Fundamentals" **[11]-(*National Instruments Corporation, 2007*)** describes DAS SW, as SW "*... that transforms the PC (CBS) and the DAQ HW into a complete data acquisition, analysis and display system*" **[13]-(*E Strassberg et al, 2006*)** similarly states that "DAQ HW without SW is useless..." and that "... DAQ HW with poor SW is almost useless", thus confirming personal beliefs regarding this subject that the front-end GUI is just as important as the back-end (*described architecture*) thus opening a new (*and expected*) research subject for the GUI.

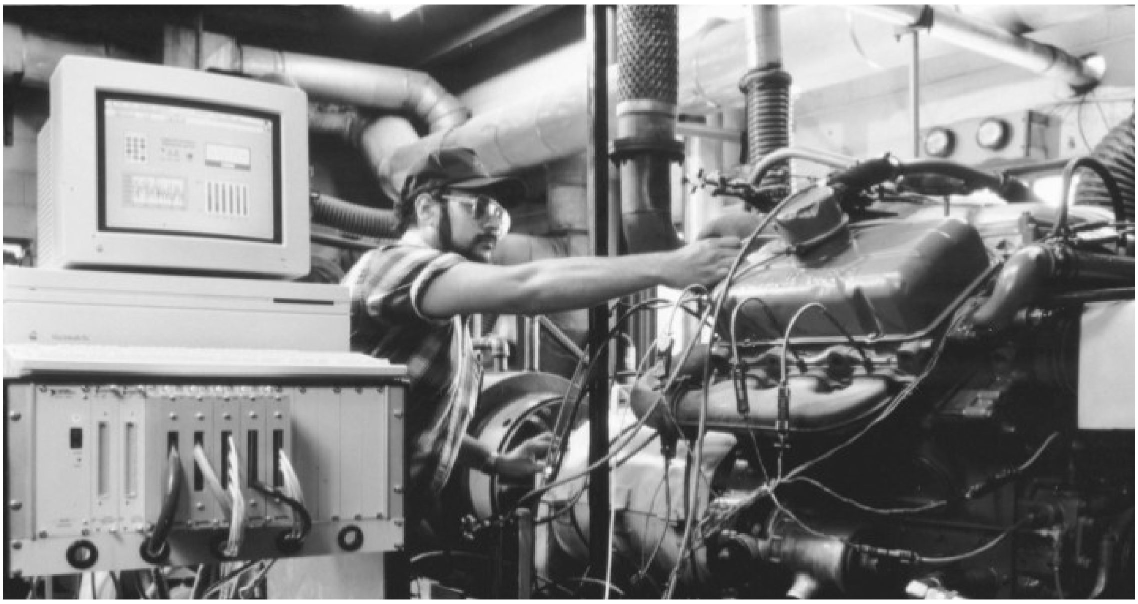
4.3.2.1 - Driver Software (DSW)

The paper states that most DAQ systems use **D**river **S**oft**W**are (DSW) describing DSW as "*... the layer of SW that programs the registers of the DAQ HW, managing its operation and its integration with the computer recourses, such as interrupts, **D**ata **M**emory **A**ccess (DMA) and memory...*" providing the front-end user with an easy to use and understand interface, hiding any complicated low-level hardware programming code/information.

There is an interesting paragraph in the paper describing how DSW can dramatically change the results and quality of the performance of front-end **A**pplication **S**oft**W**are (ASW):
"*The increasing sophistication of DAQ HW, computers, and SW continues to emphasize the importance and value of good DSW. Properly selected DSW can deliver an optimal combination of flexibility and performance, while significantly reducing the time required to develop the DAQ application*" (ASW).

[11]-(*National Instruments Corporation, 2007*)

The ASW is going to be the GUI that the user will see and use to interact with data (r*ead, interpret and analyse)*, therefore it is vitally important that this data is reliable and understandable.



[Figure 3]-(*DAQ automobile lubricant test application*)
[11]-(*National Instruments Corporation, 2007*)

[Figure 3] shows an old yet complete DAS being used on an automobile engine. Although not relevant to the *PS* or even a mobile architecture, during the time of reading, the picture provides a "nice example" as it represents a physical object (*the engine*), visually and simply displaying how the object is connected and communicates with a GUI (DSW) on the CBS using DAQ.

A paper by **[14]-(J Curtis et al, 2002)** designing DSW for DAS's highlights problematic areas and frustrations that can be avoided and therefore saves time. "*A custom computer-based DAS can be easy to design but what do you need to know to avoid design frustrations?*". **[14]-(J Curtis et al, 2002)** continues in great detail discussing these issues. This paper is thought to be the most advanced and detailed literature regarding this area of development. It is justified by other similar literatures, and is an area that will require further investigation within the next chapter.
[14]-(J Curtis et al, 2002) continues describing that DSW links DAS HW to two different types of SW, "***A**pplication **D**evelopment **E**nvironments (ADE's) and Turnkey SW*". An ADE is a programming environment that allows the programmer to create and customise applications to product requirements, whereas Turnkey SW is a pre-written application, thus requiring no user programming skills. At this stage it is assumed that for the *PS* some sort of ADE will be used to program a front-end GUI that can can be used by the user which again opens another area for further research.

The main points that **[14]-(J Curtis et al, 2002)** discusses in his paper relate to and are similar to the points in the **[11]-(*National Instruments Corporation, 2007*)** paper. Both literatures advise the same when considering what DSW to use, and describe how driver functions when controlling DAQ SW can be grouped into analog I/O, digital I/O and timing I/O pursuing them in great detail.
On the other hand this detail is believed not relevant within this review as the driver used will already have such functionality built in. However, the NIC paper additionally states that although most drivers will have this basic I/O functionality, "*... you will want to make sure that the driver can do more than simply get data to and from the device ...*" and additionally:

- Acquire data in the background while processing in the foreground.
- Use programmed I/O, interrupts, and DMA to transfer data.
- Stream data to and from disk.

- Perform several functions simultaneously.
- Integrate multiple DAQ devices.
- Integrate seamlessly with signal conditioning equipment.

This information and level of depth of the literatures is interesting, however in the interest of what is required (*the level of knowledge*) of the drivers for the development (*although good to understand*) of the *PS* it is thought irrelevant and that time could be more valuably spent in other areas. It is additionally advised that ideally, DSW is purchased form a company with a great deal of "*... expertise in the development of the DAQ SW as they do in the development of DAQ HW*" thus taking the pressure and worry for DSW away **[11]-(*National Instruments Corporation, 2007*)**.

Once data has been acquired, the ASW comes into play. Before ASW can be used or even developed the process of what is done with that data and how it will be used has to be known and understood. NIC describes that "*Once data is acquired, the most common processes are to*":

- Save the data to the local **H**ard **D**rive (HD) for post processing.
- Stream the data to a central server / computer (CBS) over TCP/IP for live monitoring.
- Perform analysis on the data as it is being acquired and return results to a central server.

[9]-(*National Instruments Corporation, 2006*)

The paper additionally states that "*Typically, not just one of these is performed in an application, rather a combination of them*". This statement is agreed upon to a certain extent however some concerns include that in relation to both the (*current plan for*) the *PS* and other reviewed systems, only one of the above points will be required and used, thus raising another argument against the NIC literature.
In regards to the *PS* data will be saved on a HD, although data will be (*as believed at this early stage in the cycle*) saved and stored in a DB which would theoretically not have to be stored locally at all. The data would not have to be streamed for "live" monitoring (*as the data is not required live*) and analysis would not be required as the data is being acquired (*analysis could take place at a later time*).
Comparing the initial thoughts and ideas of the *PS* against the NIC paper assumes that the *PS's* process method, taking place once data is acquired would need reviewing. This is somewhat contradicted in other papers detailing existing "successful" systems (*such as the "High Speed Target Tracking System"* **[15]-(*S Bhattacharyya et al, 2008*)**, *the "G-Dyno Plus GPS Performance DAQ System"* **[16]-(*Nology Enginieering Inc, 2007*)** *and the "Traqmate GPS DAS"* **[17]-(*Traqmate, 2008*)** *just to mention a few*) that work in a similar way to what is planned for the *PS* justifying personal thoughts and planned practices.
More detailed investigation and research will be required within this area in the next chapter determining which direction is best taken.

4.3.2.2 - Application Software (ASW)

Literatures describe that ASW is an additional way that DAQ software can be programmed. It uses DSW as a communicator/translator between and to control DAQ HW. The ASW SW is the SW that the user will interact with in the front-end GUI offering features such as graphical data representation and instrument control. It is therefore of equal importance as the back-end system as a whole.
Not many of the reviewed literatures discuss ASW as the focus rather concentrating on either back-end DAQ or promoting what a front-end of a system offers in terms of features other than the technologies used to create the ASW itself.
The reviewed literatures describe and confirm different ASW frameworks, environments and languages available that can be used to create such ASW using the data collected from a back-end DAS, including "C", "C++", "Java", and the evidently popular for DAS ASW (*with a lot of literatures discussing and systems using*) "LabView", all offering a "*... graphical methodology for developing complete instrumental, acquisition and control applications*" allowing the creation of powerful data logging applications.

This area, although very interesting, is distracting from the review aim(s) - (*also why other reviewed literatures may have overlooked this area*), and therefore the review will at this stage be pulled back towards the the topic of tracking mobile objects.

Some literatures discuss, that for a successful DAQ system development it is important that each and every component within the system needs to be understood.

"*Of all the DAQ system components, the element that should be examined most closely is the SW ...*". Because DAQ "*... devices do not have displays, the SW is the only interface you have to the system. The SW is the component that relays all the information about the system, and it is the element that controls the system. The SW integrates the transducers signal conditioning, DAQ HW, and analysis HW into a complete functional DAS*" **[11]-(*National Instruments Corportation, 2007*).**

The above paragraph sums up this section well. What it describes justifies and is very similar to what other reviewed DAS literatures have concluded. It additionally justifies personal opinions on and around this area, even contradicting personal thoughts regarding the importance of this area. However, it is thought that enough has been discussed regarding DAQ SW as this area is believed to be understood, knowing how and in what areas to confidently proceed with in the next chapter.

So far the review has discussed literatures mainly focussing on using GPS for tracking mobile objects (*the review will look at GPS in more detail later in the chapter*) but what other technologies are available to be used instead of or with GPS?

4.4 - GPS Alternate / Replacement Tracking Systems

An example is one of many products from "Doble Engineering" called the "F6050". A paper has been released documenting the F6050 claiming that access to the sky is not required, thus eliminating the use of GPS and promising the same advantages. This statement sounded not only an interesting concept, but the product has additionally been mentioned in other reviewed papers (a*lthough not really "thought" to be praised*) and if what Doble claim is correct this could eliminate the discussed concerns regarding GPS (*and maybe even GPS itself*). Doble state that "*... originally designed for use underground ...*" the F6050's "*... universal time synchroniser extends the benefits of GPS ... regardless of location*". This statement already identifies the fact that the product is not replacing, but "extending" the GPS technology, the paper continues, "*GPS satellite reception is not always possible, but he F6050 solves that problem ...*" providing "*... end-to-end synchronisation at remote locations where access to GPS signals is not possible*" **[18]-(*Doble Engineering Company, 2008*).** Basically, after the F6050 receives an initial signal, it can continue generating "simulated" and time accurate GPS for up to eight hours.

The F6050 is thought to be a good idea, and believed to be a reliable product as a lot of papers discuss the system and many existing and current systems have it installed and are using it on a daily basis.
In relation to the *PS* it is "personally"believed not to be necessary for use. Although the *PS* may present problems (*as discussed*) in terms of communicating with satellites, it is currently believed (*confirmed by additional literatures*) this problem is nothing that cannot be solved using the discussed methods such as signal amplifying/conditioning.
Doble's system is thought to be great for use mainly with (*although Doble do claim the system can be used in all environments*) applications in inaccessible areas such as underground environments or even within buildings.
There are many similar and existing systems including relevant literatures reviewed all describing the use of GPS with none or very little problems that cannot be resolved. These systems are more relevant and similar to the *PS* and it is therefore believed (*and justified by other reviewed literatures discussed in this chapter*) that GPS is a better direction to take regarding the project area and environment.

There are many systems existing like the F6050, again trying to act as a replacement for GPS (*even though in many cases relying on GPS*), and for the same reason they are deemed as irrelevant for further review. GPS is considered to be more reliable having a larger customer base, support facilities and options and has been used long enough now to be proven. Many more literatures read have recommended GPS than have not, and many reviewed existing systems use the technology successfully in similar environments.

Many literatures covered have also illustrated that a lot of previous, current and future projects aimed at tracking mobile objects use the **W**ireless **S**ensor **N**etwork (WNS) technology making it another important area for review.

4.5 - Wireless Sensor Networks (WSN)

WSN are described by **[19]-(*P Jueang et al, 2002*)** as networks to be used alongside "*... systems with numerous compute and sensing devices that are distributed within an environment to be studied*".
Many literatures focus on WSN using efficient sensors to track a moving object with obvious notices of appraisals by the relevant researchers and authors all stating how effective it can be (*if used correctly*) for bridging the gap between physical and logical devices, and processing data into useful and meaningful information.
The literatures reviewed all justify that sensor devices have improved dramatically, currently being of a very high standard and are "envisioned" to improve further making for a promising future solution in terms of usage, robustness and reliability.

According to a paper by **[20]-(*T Hi et al, 2004*)** wireless sensor devices can be used together cooperating and tracking positions of moving vehicles with results proving promising and both effective and efficient, although it is additionally stated that these results were proven using "simulations" which are not always the most reliable sources as they can sometimes produce inaccurate/different results to those obtained in real ("*real-world*") practice. This is confirmed by **[21]-(*S Krishnamurthy et al, 2006*)** discussing tests completed within a WSN system "*... show promising results through simulations. However, the simplified assumptions they make about the system in the simulator do not often hold well in practice ...*".
Some of the reviewed literatures have highly rated and appraised the accuracy and reliability of todays simulations whereas other (*most*) studies have contradicted this, sometimes being very critical towards simulations with for example the paper **[20]-(*T Hi et al, 2004*)** stating that "... *simulation approaches tend to make simplified assumptions that often do not hold well in practice and they are all subject to incompleteness*" thus raising the question to the reader of the actual reliability of such simulations and whether they can be trusted or not in regards to further research.

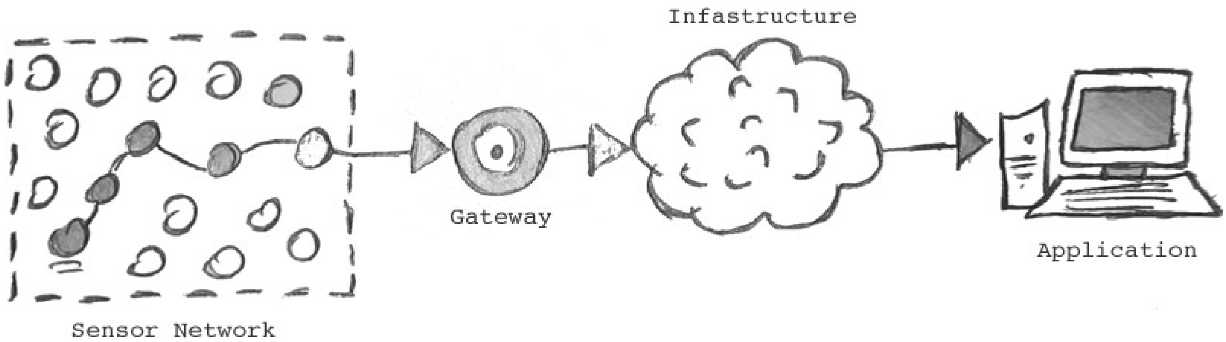
The paper by **[19]-(*P Jueang et al, 2002*)** discusses the field of WSN's, focussing on the "*self-sufficient use of compute, sensing and wireless communication devices*" for varying purposes. The paper additionally discusses how tracking devices are positioned, used and operate a peer-to-peer network delivering logged data back for further analysis. The tracking devices that are being used in this particular case are used to track Zebras using a collar containing the tracking device that is attached to the zebra (*strapped around the zebras neck*) additionally containing further wireless computing devices that allow the system to function including the addition of "*GPS, flash memory, wireless transceivers and a small CPU*".
Although relevant and sharing similarities, this system differs slightly from the *PS* requiring further interesting factors for its success which are not necessary for the *PS.* This includes the fact that the researchers and analysts themselves are also mobile which is unlike the *PS* as there is no fixed CBS towards which data can be sent/retrieved. Additionally within the area in which the research for the described project is being covered, there is no cellular service or broadcast communications, again all creating challenges that can be eliminated for further review relating to the *PS*.

During the literature search and study, WSN's were thought to be very relevant and a technology that could potentially offer a lot to the *PS*. It was therefore decided to look further into relevant literatures regarding this subject.

From reading many differing literatures about WNS's a lot has been learned regarding the subject area. WSN's are becoming more popular each year, and with this increase in interest brings greater investment allowing the technology the opportunity to become very powerful. In a paper **[22]-(*M Dohler, 2008*)** the author discusses that "*... an expected market size of approximately $2b by 2012 at a compound annual growth rate of 41.9%, therefore causing the interest in this technology to augment dramatically*" this growth is confirmed by **[23]-(*H G Goh et al, 2006*)** explaining how in recent years the technology has taken off, although additionally explaining that it still has many problematic areas. Reading the **[22]-(*M Dohler, 2008*)** paper, as well as a paper by **[21]-(*S Krishnamurthy et al, 2006*)** the arguments regarding growing popularity and current problematic areas justify one another nicely. The reviewed papers although speaking highly of the technology contradict themselves because of those problematic areas. Each author has the awareness (*and hope*) that these issues with the technology although known about will be resolved because the growing interest will allow both time and money to be invested into the technology. **[22]-(*M Dohler, 2008*)** further explains (*being the most up-to-date paper*) that "*... WSN's have whitenessed a tremendous upsurge in the last decade, which is mainly attributed to their unprecedented operating conditions and hence unlimited research challenges*". The paper continues justifying **[21]-(*S Krishnamurthy et al, 2006*)** that due to design constraints, none of the major communities (*computing, telecommunications, physics and biology*) can make such systems operate efficiently on there own. This can be confirmed as many of the reviewed existing systems using the WSN technology incorporate additional technologies the most popular and common being GPS.

Other literatures praise WSN's recommending them for use and not mentioning any of the problematic areas. It is believed, due to the fact papers have been written focussing mainly on problems within the technology, that it is evident that these problems exist. It is assumed that the technology does expose those problems to a certain extent when using it, and/or within a certain method or environment. There are many systems that use WSN's successfully and problem free, including usage within systems similar to the *PS*.

The best, most simple and effective summarisation of a WSN is considered to be (*what it is and how/where it can be used*) from all the literatures reviewed to be found in the paper by **[22]-(*M Dohler, 2008*)**. **[22]-(*M Dohler, 2008*)**. It explains that within a typical WSN monitoring application "*... the nodes are generally distributed in a large-scale roll-out (e.g. from an airplane), hence leading to random node placements with fairly homogenous density. Some nodes, however, can be placed along given structures (road, river, etc), leading to a random rollout within a confined space. The sensor nodes typically report their sensing reports on a regular basis, as well as trigger alerts if large deviations in the sensed data are observed. The nodes are generally battery-run and hence energy constrained; they could also be re-charged by, e.g. solar cells. Also, there is a limited set of processing units or gateways which collect the information gathered by the sensor nodes*".



[Figure 4]-(*A simple WSN*)

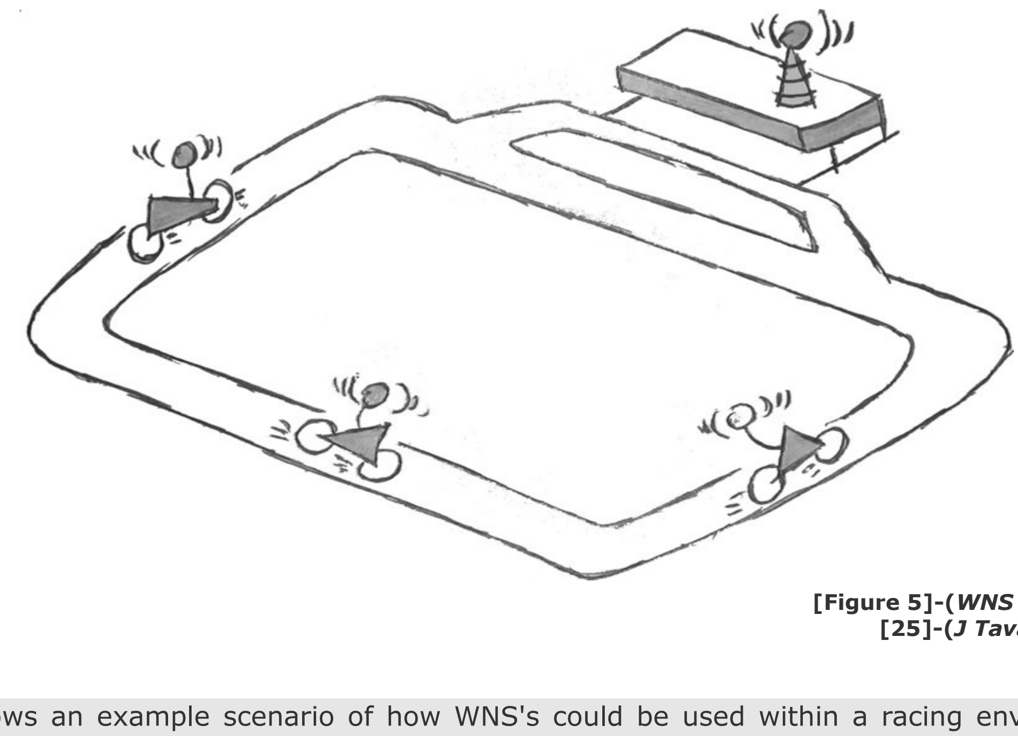
As discussed and seen in **[Figure 4]** a WSN is a wireless network which is constructed of devices that are spread out using the addition of sensors for monitoring and communicating to an application through a Network Infrastructure, in the case of the *PS* movement at different times and locations. Additional (*and also relevant*) conditions that WSN's can be used to monitor (*both physical and environmental*) include sound, temperature, pressure and vibration to name but a few. WSN's are used within many different fields mainly within a "scattered" area where it is to collect data using sensor nodes, with the most popular being: (Surveillance, Military, Civilian, Habitation, Industry, Agricultural, Medicine (*healthcare applications*), Automobile, Traffic Control and Object tracking)

Most reviewed papers and systems use different types of sensors. It is described in the papers that the chosen sensors are (*most of the time*) used as they are best suited for the project, depending on the nature and purpose of the project in terms of physical size, longevity and even cost. **[24]-(*M Friedermann et al, 2004*)** describes that the nodes within a WSN are are "*... typically equipped with a radio transceiver or other wireless communication device ...*" being common in all reviewed WNS's, with "*... a small micro controller, and an energy source ...*", in most cases being a battery (*this will be discussed later in the chapter*). **[24]-(*M Friedermann et al, 2004*)** continues to describe how the size of nodes can vary dramatically from being shoebox sized to devices the size of a "*grain of dust*", and the cost of the nodes is again very differing ranging form a few cents to a few hundred dollars depending on both the size of of the sensor network and the level of complexity required from the nodes.
In regards to the *PS* understanding the fact that nodes can be acquired as a small size, thus not causing any dramatic effect when being used with (*and possibly on*) motorcycles is good, however, raising a few questions that will need further investigation. These Include the fact that the cost of such small nodes could be very expensive, and are they capable of supporting the system requirements. Cost and size constraints can result in corresponding constraints on project resources including speed, bandwidth, power, memory and energy, which has been made apparent in most WSN related literatures.

WSN's were originally developed for use within the military to aid in special military operations. Literatures regarding most of the above areas have been found and studied, with most being successful and in use today.

[21]-(*S Krishnamurthy et al, 2006*) describes efforts in the sensor network community to build an integrated WSN for surveillance missions. "*The focus of this effort is to acquire and verify information about enemy capabilities and positions of hostile targets ...*" allowing "*... the ability to deploy unmanned surveillance missions ...*" (*thus not endangering men*) by using WSN's this "*... is of great practical importance for the military*" **[21]-(*S Krishnamurthy et al, 2006*)**. A disadvantage discussed in this paper, that is additionally discussed by **[25]-(*J Tavares et al, 2007*)** is that of power. WNS's are very power hungry and therefore this needs to be taken into account during the design stage of such a system. Each node within a system has limited power as devices are wireless and are therefore in most cases battery powered. The transmission and receiving of data can be costly (*Ex. 12mA to transmits and Ex. 8mA to receive*) **[25]-(*J Tavares et al, 2007*)**, again draining power sources and limiting recourses. From the reviewed literatures, power seems to be the main concern with WSN's. No reviewed literature has found a 100% secure solution to this but some attempts to keep costs down and efficiency to a maximum include the use of rechargeable batteries, implementing a sleep mode into nodes and also in some cases adding solar panel recharge-ability, each having obvious advantage,s and disadvantages. Depending on when, how and where the system will be used will depend on what (*and even if*) precautions can be implemented. **[21]-(*S Krishnamurthy et al, 2006*)** additionally describes that "*... because of the energy constraints of sensor devices, such systems necessitate an energy-aware design to ensure the longevity ...*" *of the systems*.
A paper by **[26]-(*J Stankovic et al, 2007*)** justifies the above papers as it discusses a WSN system that is used to allow "*... a group of cooperating sensor devices to detect and track the positions of moving vehicles in an energy and stealthy manner*". **[26]-(*J Stankovic et al, 2007*)** continues to explain the same discussed power (*longevity*) limitations describing how the project design can cater for this by its ability to "*... tradeoff energy awareness ... and performance by adaptively adjusting the sensitivity of the system*" continuously monitoring and evaluating the performance with the addition of results showing that this strategy achieves a significant extension of network lifetime.

[25]-(*J Tavares et al, 2007*) explains WNS's at a level that is more relevant to the *PS.* **[25]-(*J Tavares et al, 2007*)** describes how it is possible to fit nodes to vehicles themselves communicating results to a CBS as the vehicle is moving. In the case of a racing scenario with multiple vehicles, there would be multiple nodes (*one attached to each vehicle*). It has been argued by many authors that WSN's are not good for use with High-speed objects, in regard to actual position coordinates and in comparison to a GPS system. Example uses of WNS's in such a system could instead inform drivers of accidents or traffic ahead, and even with latest technologies, slow a vehicle down (*brake*) in such cases. However, more relevant to the *PS,* WSN's are good and have proved positive in current systems for tracking and monitoring vehicle statistics such as tyre pressures, throttle positions, RPM, fuel consumption and even acceleration and braking statistics.



[Figure 5]-(*WNS racing scenario*)
[25]-(*J Tavares et al, 2007*)

[Figure 5] shows an example scenario of how WNS's could be used within a racing environment with each of the three vehicles acting as a node within the network communicating with the pits (CBS), providing the drivers of the vehicles information such as tyre pressure(s), engine management information, failures and acceleration information sending any help requests. Even heart rates and blood pressure levels of the drivers, along with all the driver information could be sent back to the pits, thus opening another level of study, expansions and and possibilities to the *PS.*

Over all WSN's appear to be quite popular. Although many authors describe and focus on the problems that WSN's can cause, with other authors praising the technology. When used at a low level within the system, it is believed that the use of a WSN can be very convenient offering a lot of advantages to the system. the usage of a WSN (*if used at all*) within the *PS* should be investigated further in regards to where and how it will and should be used (*as when designing any system*) enabling getting the best from the technology in such a scenario.

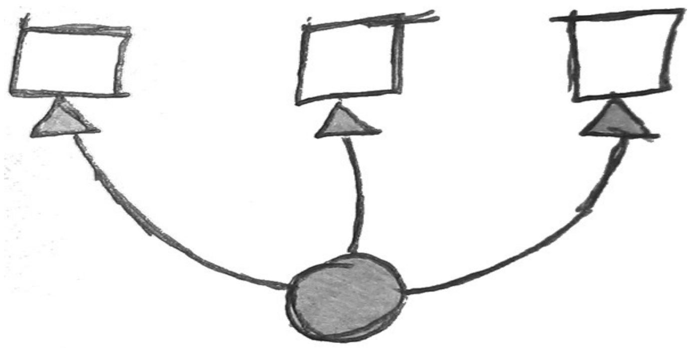
An additional and new proposed area building rapidly onto existing architectures found during the literature search that would benefit further research was found in a literature by **[27]-(*M Ruggeiri et al, 2008*)** regarding **N**ext **G**eneration **W**ireless **N**etworks (NGWN) potentially being a solution to the problematic areas within WSN's and even giving a possibility to tracking the location of the mobile device itself. **[27]-(*M Ruggeiri et al, 2008*)** describes NGWN's as a wireless network that "*... will allow a user to roam over different access networks, such as UMTS, wi-fi, and satellite based networks ...*" which are currently "*... integrated/assisted by more and more navigation systems, which can make available the information on the location of the mobile terminal. This information is typically used to provide location based services*", thus offering hopeful perspectives and options in regards for the PS. As the architecture currently has no solid foundations, it is considered irrelevant to review further literatures regarding NGWN's, however, such technology offers great power and possibilities to future applications.

4.6 - Wireless Architectures (Passive and Active)

During the literature search not many literatures (*regarding existing systems*) mentioned or discussed in detail the architecture that systems were or could be based upon. With a lot of systems it is thought that a "Hybrid" styled architecture would have been implemented judging on the concepts and usages that they were performing and producing.

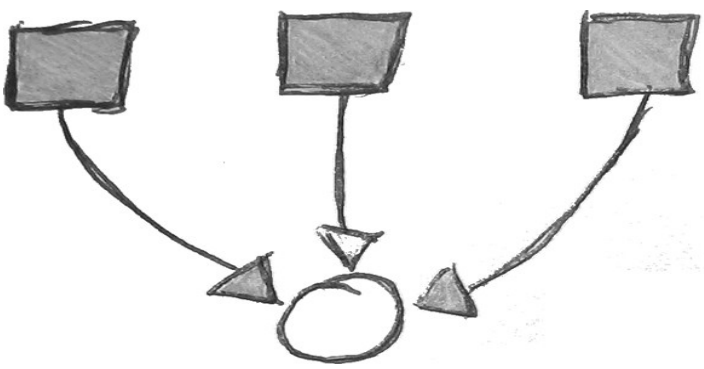
Focussing back to the actual tracking of a mobile object, **[28]-(*A Smith et al, 2004*)** interestingly describes the process by following one of two possible architectures:

1. An **active** mobile architecture.
2. A **passive** mobile architecture.



[Figure 6]-(*Passive mobile architecture*)

[Figure 6] shows a passive mobile architecture with "*... fixed nodes at known positions periodically transmit their location (or identity) on a wireless channel, and passive receivers on mobile devices listen to each beacon*" **[29]-(F *Dugast et al, 2007*)**.



[Figure 7]-(*Active mobile architecture*)

[Figure 7] shows an active mobile architecture with "*... fan active transmitter on each mobile device periodically broadcasting a message on a wireless channel*" **[29]-(F *Dugast et al, 2007*)**.

Literatures have been found with very similar although different opinions and descriptions of such architectures. As a summarisation, a passive mobile architecture has "beacons" that periodically transmit signals to a "passively" listening mobile device, thus in turn estimating the distances from the device to the beacons themselves, whereas an active mobile architecture uses receivers that are accurately positioned at known locations (*for example around a racetrack*) estimating the distances to a mobile device (*a motorcycle*) based upon active transmissions from the device itself.

Both discussed architectures are relevant and could possibly be used (*are both candidates for use*) with the *PS*, although it is thought (*at this stage*) that the active architecture would be better suited to the system as a literature by **[28]-(*A Smith et al, 2004*)** describes that it "receives *simultaneous distance estimates from multiple receivers from the mobile device*", and is described "*... to perform better tracking than the passive mobile system in which the device obtains only one distance estimate at a time and may have moved between successive estimates*" .

[27]-(*M Ruggeiri et al, 2008*) describes in a paper about satellite-based networks, that in such networks the nodes are of a mobile nature, justifying reviewed literatures regarding these networks. Within mobile networks, "*... the term location can also be referred to as the geographical coordinates of the network nodes*" and can therefore be used more efficiently to manage the dynamics of a "*... network topology, the coverage and also the resource allocation*". **[27]-(*M Ruggeiri et al, 2008*)** further explains how such location systems can be separated into either "Physical" or "Symbolic" information. **[30]-(J. Hightower, 2006)** describes (justifying **[27]-(*M Ruggeiri et al, 2008*)**) both physical and symbolic information as:

- Physical information provides the position of a location on physical coordinate system (X,Y,Z) for example "*... for example the Electronics Engineering Department is at (x1,y1) coordinates*".
- Symbolic location information provides descriptions of the locations, "*... for example the Radar laboratory at the Electronics Engineering Department*", Symbolic location information is additionally is related to abstract ideas, and physical location information "*... can be derived by symbolic position with additional information*". "*Using only symbolic location information can yield very coarse grained physical positions*".

[27]-(*M Ruggeiri et al, 2008*) Continues describing that all objects located within an absolute location system use a shared reference grid with each object having its own reference frame, and that an absolute location can be converted into a relative location.
A paper by **[31]-(*H. Balakrishnan et al, 2007*)** explains that active mobile architectures have an active transmitter on each mobile device (*node*) that periodically broadcasts a message on a wireless channel, whereas on the other hand, **[31]-(*H. Balakrishnan et al, 2007*)** explains that, within a passive mobile architecture there are fixed nodes at known positions that periodically transmit their location (*identity*) over wireless channels, using passive receivers on each mobile device (*node*). These listen to each beacon. All literature reviews regarding mobile architectures justify one another, however, **[31]-(*H. Balakrishnan, 2007*)** states that both passive and active mobile architectures are best suited within an indoor environment.
Many papers have discussed outdoor mobile systems using such architectures thus arguing the statement made by **[27]-(*M Ruggeiri et al, 2008*)**. **[31]-(*H Balakrishnan et al, 2007*)** describes regarding "*in indoor location systems, that there are in general two different types of mobile devices: active and passive*", although **[31]-(*H Balakrishnan et al, 2007*)** does not state that these devices cannot be used in an outdoor environment, and furthermore does not state that the architectures themselves would or can be affected by either and indoor or outdoor environment. Both authors, however, do explain that for usages in outdoor environments, a GPS based system is best suited and most beneficial, therefore an obvious contradiction has been stumbled across regarding the architecture types. Literatures and authors have not only different opinions regarding the architectures, but also different factual information. Further and more in depth research is obviously required within this area and also into additional and maybe more advance/suited (*GPS based*) architectures to get clearer and better understanding of if and how each architecture "could" be used with the *PS.*

4.7 - Existing tracking Methods

During the literature search it was surprising to find how many different tracking methods and technologies exist and are covered by differing literatures all for different (*yet similar/relevant*) needs and requirements.
Literatures covered were not only relevant to technologies following the same foundations as the *PS*, but also those that were different, Relevant work including different environmental settings (*indoor, outdoor and even underwater*) and differing architectures (*both centralised and de-centralised*) were also read.
The literature by **[32]-(*A Smith et al, 2004*)** proved both very useful and interesting in regard to the understandings and learning relative to the *PS.* Most of its trends and theories were confirmed in additional literatures. The paper discusses different tracking methods and technologies currently in use that are clearly relevant for reviewed in terms of systems like the "**Active BAT Tracking System**" which like the *PS* uses a centralised architecture.

The "**Active BAT Tracking System**" uses a collection of fixed nodes that are arranged on a grid all receiving ultrasonic chirps from a mobile device. The computing distance is estimated to the device using the time of flight of the ultra sonic signal. These distances are forwarded to the central computer (*CBS*) where the relevant computations take place computing the devices exact position(s).
The paper clearly states, and is additionally supported by other reviewed literatures that although usually expensive to initially set up and implement a "*... centralised structure allows for easy computation and implementation, since all distance estimates can be quickly shipped to a place where computational power is cheap. Moreover, the* **active** *mobile architecture facilitates the collection of multiple simultaneous distance samples at the fixed nodes, which can produce more accurate position estimates*" **[32] – (*A Smith et al, 2004*)**.

The paper additionally discusses other relevant systems in terms of requiring high precision tracking within the presence of "large and erratic" accelerations thus the need and requirement in providing very precise estimates. Systems like the "**HiBall Head Tracking System**" and the "**Whisper System**".

4.7.1 - HiBall Head Tracking System

The "*HiBall Head Tracking System*" uses a panel of infrared LED's that take turns flashing with the use of cameras that measure the positions of the flashing LED's communicating with a CBS that uses knowledge regarding the geometry of the heads device cameras to compute the location of the mobile device.
Although providing very efficient and accurate/precise results, this system is not ideal for the proposed usage as it would prove very difficult, very expensive and inefficient to deploy large numbers of LED lights around a race circuit, as well as being set in an outdoor environment where ambient light may (and is more than likely to) cause interference.

4.7.2 - Whisper system

The "*Whisper system*" uses a spread spectrum audio approach to obtain precise distance measurements encoding information on an audio stream using time-of-arrival methods to get the exact distance measurements.
Although achieving good results, due to the large bandwidth that could be required from such a system and the continuous nature of the spread spectrum signal further leading to effective tracking, like the "*HiBall Head Tracking System*" it is not ideal for outdoor usage as studio background noise can interfere with the accuracy of the results.

This justifies the comments about such architectures previously discussed by **[31]-(*H. Balakrishnan et al, 2007*)** and **[27]-(*M Ruggeiri et al, 2008*)** regarding the use of such architectures within indoor environments. This emphesises and confirms the confusion and highlights the need for further investigation into this area.

This paper by **[28]-(*A Smith et al, 2004*)** along with many others is considered to be a very good find, and although not giving exact solutions to faced problems, the paper clearly hilights and indicates some very important issues, points and areas that will help to peruse and direct further literature searches.
The paper continues (like many others) to discuss the GPS technology, although better (more in-depth) sources are available and were found, reviewed and preferred covering this technology.

4.8 - Global Positioning System (GPS)

GPS has already been discussed within this chapter, explaining advantageous and disadvantageous areas of the technology, with many literatures and existing systems successfully using it.
Due to the popularity and high usage of GPS within such systems and areas, further literatures have been searched and studied regarding the technology to see what additional information can be obtained helping to push the progression of the project further.

GPS is one of the current and most commonly known and used tracking technologies and is therefore a very vast subject area covering large scales providing great opportunities.
There are a lot of literatures available covering GPS in great detail, all of which have proved very interesting and helpful, however in relation to the proposed system the following three papers have proved the most useful **[33]-(H Bienser et al, 1998)**, **[34]-(*H Wu et al, 2005*)** and **[35]-(*V Zeimpekis et al, 2002*)** detailing and discussing general usages of the technology, the advantages and disadvantages of its uses, fundamental and technological areas concluding and evaluating each with the addition that the papers suitably compliment and justify one another well.
All authors generally state and confirm that GPS is currently one of the most sophisticated and commercially available tracking systems, and as technologies have improved and demands for better accuracy and reliability have become stronger, GPS has primarily taken control over the mobile tracking market domain, becoming more available and more widely used, and as discussed by **[19]-(*P Jueang et al, 2002*)** is replacing less sophisticated tracking methods such as aided visual observations (VHF).
The literatures reviewed clearly explain how the technology tracks positions and uses satellite uploads to transfer data to a base station. **[35]-(*V Zeimpekis et al, 2002*)** describes how "*... a constellation of 24 high-altitude satellites with very accurate atomic clocks, along with global network of satellite tracking stations and sophisticated ground processing stations*" are used to communicate with receivers.
As already previously discussed in this chapter regarding the GPS technology from different sources (authors and literatures) and now further sources, all describe and justify that GPS is not perfect and can suffer form significant considerations, additionally including:

- Slow data uploads
- Slow and expensive downloads (*users/researchers charged by the bit*)
- Devices are expensive and bulky (*more modern devices are now available and are a lot better*)
- Devices are power hungry
- Only support of infrequent uploads
- Battery operation (*in most cases without solar re-charge*)

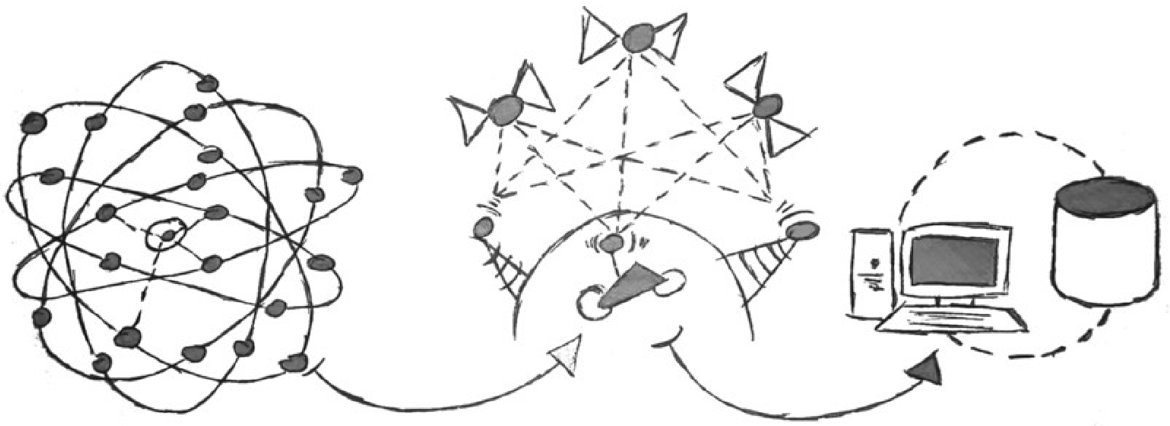
Thus raising the question if GPS is the best suited technology for use with the *PS*, however, it is stated by **[32]-(*A Smith et al, 2004*)** that GPS has its advantages proving suitable for use in terms of being scaleable, operating well outdoors and enabling simultaneous distance estimates, all of which no other (*within reason*) technology provides to such a high level for such usage.

As mentioned, GPS is a very large and vast subject covering a great scale and area(s). Due to the amount of literatures found during the literature search, again an elimination process was required narrowing down and selecting a few literatures that are believed to have helped with the understanding of GPS and additionally literatures that are more relevant in regards to the *PS.* Due to the vast scale of this subject area, to avoid getting distracted and veering off course into irrelevant and unnecessary areas, a set guideline has been created that will be referred to during this section of the chapter. The guideline will focus on the principles of GPS providing a background for more advanced material and further research areas.
This section of the chapter should focus on resources that are relevant to the *PS* however, additionally focussing on not so relevant areas/literatures so long as they are assumed to help the progress of the project. This will help to grasp any related theories and terminology's, and additionally any mathematical foundations (*at a basic level*) that may be required.

GPS is said to be changing how we operate and opening up many new possibilities that without the technology would not even be dreamed of. The "*... GPS technology is changing the way we work and play*" **[36]-(*Garmin International, 2008*)**. The **[36]-(*Garmin International, 2008*)** paper describes some of the different and potential uses of GPS based on a personal (*one-to-one*) usage, detailing that GPS can be used during driving, flying, fishing, sailing hiking, running, biking, working and exploring. Although not completely relevant to the *PS*, the paper is considered very beneficial in terms of its "high-level" overview descriptive nature of the technology which is all that is required at this stage of planning.

A very detailed and technically advanced (yet reasonably old) paper that has proven very interesting and helpful describing the basic idea of GPS summarising nicely what other papers have discussed in great depth. "*... GPS positioning is based on trilateration, which is the method of determining position by measuring distances to points at known coordinates ... At a minimum, trilateration requires three ranges to three known points. GPS point positioning ... requires four pseudoranges to four satellites*" **[37]-(*G Blewitt, 1997*).** The paper although reasonably old explains and justifies what more modern and up-to-date literatures regarding the same subject explain. The writing style of the author of this paper is preferred over other reviewed papers, as the content is explained at a high level summarising a subject nicely with a more detailed low-level description being provided if the reader is interested and wishes to read further, thus making for a vary fast, easy and effective read and method to find exactly what is wanted from the paper.

In a paper **[38]-(*J Zogg, 2002*)**, ***[38]-(J Zogg, 2002)*** justifies the descriptions from both **[36]-(*Garmin International, 2008*)** and **[37]-(*G Blewitt, 1997*)** of what GPS is, its foundations, its principles and for what, why and how it is used describing that GPS is "*... a process used to establish a position at any point on the globe*". ***[38]-(J Zogg, 2002)*** also explains that both an objects "*.. exact location (longitude, latitude and height coordinates) accurate to within 20m to approximately 1mm*" and the "*... precise time (**U**niversal **T**ime **C**oordinate (UTC)) accurate to within 60ns to approximately 5ns*" can be determined anywhere on earth with the use of GPS. This confirms other papers thus adding a better perspective and more information regarding the four coordinates from what is described in the paper by **[37]-(*G Blewitt, 1997*)**.
As described in other papers, these coordinate values are determined by 28 satellites that are continuously orbiting the earth, and the use of four coordinates can calculate the speed and direction of travel.

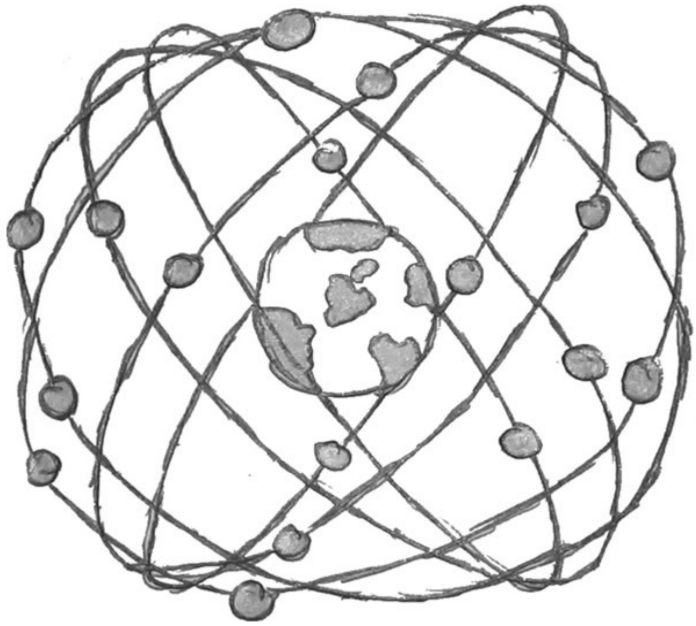


[Figure 8]-(*The basic function of GPS*)

GPS receivers are small devices of different shapes and sizes that contain and transmit location information (*the device that the location of is being tracked*) and are used in and for many different circumstances and purposes. All general GPS related papers reviewed explain very similar systems and purposes of GPS in general and GPS receivers themselves with the most common being related to positioning, location, navigation, surveying and determining time. **[38]-(*J Zogg, 2002*)** further explains that GPS devices are used by both private individuals (*e.g. trekking, cross-country skiing and many other situations*) as described previously by **[36]-(*Garmin International, 2008*)** and also companies (*e.g. surveying, navigation, time determination and vehicle monitoring*) alike.

[39]-(*W Kunysz, 2004*) describes how encoded signals are transmitted from the satellites (*shown in **[Figure 8]***) towards earth. This "*... signal is encoded with the navigational message, which can be read by the users GPS receivers*" **[37]-(*G Blewitt, 1997*).** Navigation messages include orbital values which are used by the receivers to calculate the satellites coordinates (*X,Y,Z*).
The calculated coordinates are cartesian coordinates in a "geocentric" system, which (as ***[37]-(G Blewitt, 1997)*** describes) "*... has its origin at the earth center mass, **Z** axis pointing towards the North pole, **X** pointing towards the prime Meridian...*" (*crossing Greenwich*) "*... and **Y** at right angles to **X** and **Z** to form a right handed orthogonal coordinate system*". ***[37]-(G Blewitt, 1997)*** continues to describe the algorithm needed for calculation of this process (*Ephemeris Algorithm*) however, this is deemed to low level for this review.

The reviewed literatures describe that like WSN, GPS was originally developed by the U.S. **D**epartment **o**f Defense (DoD) for use by the military on a private signal called **P**recise **P**ositioning **S**ervice (PPS) and was initially named **NAV**igation **S**ystem with **T**iming **A**nd **R**anging **G**lobal **P**ositioning **S**ystem (NAVSTAR-GPS) which was later known more commonly as GPS, and additionally the standards being made available to civilians. A separate signal called **S**tandard **P**ositioning **Si**gnal (SPS). SPS can be freely used by the public, whereas PPS is private and can only be used by the military and other relevant governments agencies. **[38]-(*J Zogg, 2002*)** continues confirming already obtained knowledge that there are "*... currently 28 operational satellites orbiting the earth at a height of 20,180km on 6 different orbital planes"* with there orbits having a 55 degree incline to the equator making sure that a minimum of at least 4 satellites have radio communication with any point of the planet. *"Each satellite orbits the earth in approximately 12 hours and have 4 atomic clocks on board*".



[Figure 9]-(*GPS satellites orbit the earth on 6 orbital planes*)

Within reviewed literatures explaining the concepts of GPS, all have discussed "pseudoranges" although none apart from the paper by **[37]-(*G Blewitt, 1997*)** describe what "pseudoranges" actually are**.** ***Blewitt*** explains pseudoranges as "*... time that the signal is transmitted from the satellite is encoded on the signal, using the time according to an atomic clock onboard the satellite. Time of signal reception is recorded by receiver using an atomic clock*". A receiver measures difference in these times:

pseudorange = (time difference) * (speed of light)

Now that the meaning of "pseudoranges" has been established a lot of areas within other papers regarding "pseudoranges" are understood. It is noticed that in other papers including the paper by **[38]-(*J Zogg, 2002*)** when discussing "distance" they are actually referring to "distance", which going on the above description is not completely correct. **[38]-(*J Zogg, 2002*)** explains that "*the distance **S** to the satellite can be determined by using the known transit time **t**" -* (***S=t*c***).

[37]-(*G Blewitt, 1997*) additionally states that the "*... pseudorange is almost like range, except that it includes clock errors because the receiver clocks are far from perfect*", which is an interesting comment as in some papers the clocks are described as reliable and in others (*like this paper*) the clock times are considered to be problematic.

A paper by **[40]-(*G, Lachapelle et al, 1995*)** confirms other authors describing that each satellite has four atomic clocks on board, it also states that they are currently the most precise instrument known, loosing (*a maximum*) up to 1 second every 30,000 to 1,000,000 years, **[40]-(*G, Lachapelle et al, 1995*)** additionally explaining that to make them even more accurate they are synchronised on a regular bases with various CBS's around the globe. "*... Each satellite transmits its exact position and its precise on board clock time to Earth at a frequency of 1575.42MHz*". The signals are "*... transmitted at the speed of light (300,000 km/s)"* and therefore requiring approximately 63.3ms to reach a position on Earth. To establish a position on land or sea, all that is required is an accurate clock. "*... By comparing the arrival time of the satellite signal with the on board clock time the moment the signal is emitted, it is possible to determine the transit of time of that signal*" **[38]-(*J Zogg, 2002*)**.

[37]-(*G Blewitt, 1997*) however, does explain further in his paper regarding the clock time, how to correct any clock errors. "*Satellite clock error is given in the navigational message, in the form of polynomial*". The receiver that has an unknown receiver clock error con be estimated by the user along unknown station coordinates. "*There are four unknowns, hence we need a minimum of four* *pseudorange measurements*". justifying that in fact clock times can actually be reliable when corrected or estimated, thus putting this authors view back in line with authors of other literatures.

The papers continue to describe the workings of GPS in great and advance detail that is deemed unnecessacry for discussion within this chapter. This is however relevant in the next chapter, and the review of this subject has so far opened up a lot of areas need further investigation.

Moving back to the GPS foundations, the ***[36]-(Garmin International, 2008)*** paper overviews generally what all the related literatures are describing at a less detailed level. The paper confirms that GPS is a satellite navigation system sending and receiving radio signals. GPS receivers acquire the signals and providing the user(s) of the devices with information. "*... using GPS technology, you can determine location, velocity and time 24 hours a day, in any weather conditions anywhere in the world for free*", continuing to justify its history about the military and NAVSTAR and that due to its possibilities and advantages was made public by the government.
The paper separates the technology into three segments that are deemed additional areas for further investigation in the next chapter:

- Space Segment.
- Control Segment.
- User Segment.

An interesting comment that the ***[36]-(Garmin International, 2008)*** paper makes is regarding the weather conditions "*... using GPS technology, you can determine location ... in any weather conditions...*" which contradicts what was learned earlier in this chapter that poor weather conditions "can" greatly affect the performance and accuracy of GPS and even prevent communication between device and satellite, again hilighting an area for further investigation. What is strange is that further in the same paper it is stated by the same author that "*... GPS technology depends on the accuracy of signals that travel form GPS satellites to a GPS receiver*". Although putting a different perspective to the argument, the author is still contradicting previous comments further stating that GPS accuracy can be increased when it is used "*... in an area with few with few or no obstacles between the mobile device and the sky*". This latter statement relates to how other literatures describe GPS limitations, but as mentioned contradicts itself, thus raising the question of how reliable this comment is.

The ***[36]-(Garmin International, 2008)*** paper does however give a good summary about what has been discussed, confirming the points made in all literatures reviewed so far regarding GPS at work:

- The control segment consistently monitors the GPS constellation and uploads information to satellites to provide maximum user accuracy.
- The GPS receiver collects information form the GPS satellites that are in view.
- The GPS receiver accounts for errors.
- The GPS receiver determines your current location, velocity and time.
- The GPS receiver can calculate other information, such as bearing, track, trip distance, distance to destination, sunrise and sunset times etc.
- The GPS receiver displays the applicable information on the screen.

Many literatures have been reviewed up to now regarding "general" GPS basics, and a good knowledge and understanding has been accomplished with guidance from different author views, opinions and explanations regarding the technology of GPS devices. It is believed that further literature review of the workings of GPS at such a level is going off track from the set guidelines for this chapter, so the review will move on looking into the tracking of high speed moving objects.

4.9 - Telematics

During the literature search a domain that was mentioned and highlighted by many authors and deemed relevant to the PS is "telematics" and "telemetric systems".

Telematics can be used in a number of ways, but more relevant to the *PS* and described by **[41]-(J *Mikulski, 2006*)** it can be applied to the discussed GPS technology "*... integrated with computers and mobile communications technology*" although "*... the term is used to refer to the use of such systems within road vehicles*".
The **[41]-(J *Mikulski, 2006*)** discussion continues hilighting general uses of the technology with "some" being relevant to this review, with its comments and arguments being reliably justified by many other sources including a discussion by **[42]-(*N Nixon, 2004*)**.

The main uses of a telematics system are described as tracking fleet vehicles, recovering stolen vehicles, location driven driver information services and also collision notification progressing into another relevant area of **D**edicated **S**hort **R**ange **C**ommunication (DSRC) - (*discussed below*) with **[42]-(*N Nixon, 2004*)** discussing the additional uses of remote diagnostics, alert notifications, vehicle configurations, vehicle mileage capture, vehicle wear and tear and route management, very similar to the previously reviewed WSN technology usages.
[43]-(*E Belgeonne, 2006*) CEO of the telematics provider "Thales Telematics" states that "*... by using telematics products, like our own Orchid system, customers can receive street address location, vehicle speed, digital mapping and direction of there vehicles via the Internet*" and although still in early stages of implementation, with fewer than two million users, forecasts predict that the user base will soon have swelled to a massive nine million.

Authors describe later enhancements of the telematics technology providing on-board vehicle diagnostics, reporting both electronically and mechanical faults to both driver and CBS (*as currently and commonly used within Formula One*).
Telematics can be very useful for this project due to its vehicle tracking abilities as described by **[42]-(*N Nixon, 2004*)**. The technology is primarily used by Formula One teams to monitor car locations, movements, statuses and car behaviors during a race. The cars are equipped with a GPS receiver and an electronic device (*usually compromising of **G**lobal **S**ystem for **M**obile Communications (GSM) / **G**eneral **P**acket **R**adio **S**ervice (GPRS) technologies offering accurate positioning coordinates*) **[41]-(J *Mikulski, 2006*)** enabling the communication with the driver and the relevant PC software being used. The data collected is turned into relevant and meaningful information by using specialised tools in conjunction with a visual display and using specialised computer mapping software.

Telematics has proved to be a very relevant and important area/factor that has been identified affecting the progression of the *PS* and is therefore an additional area that will be taken into the next chapter for further more in depth research and investigation.

4.10 - Dedicated Short Range Communications (DSRC)

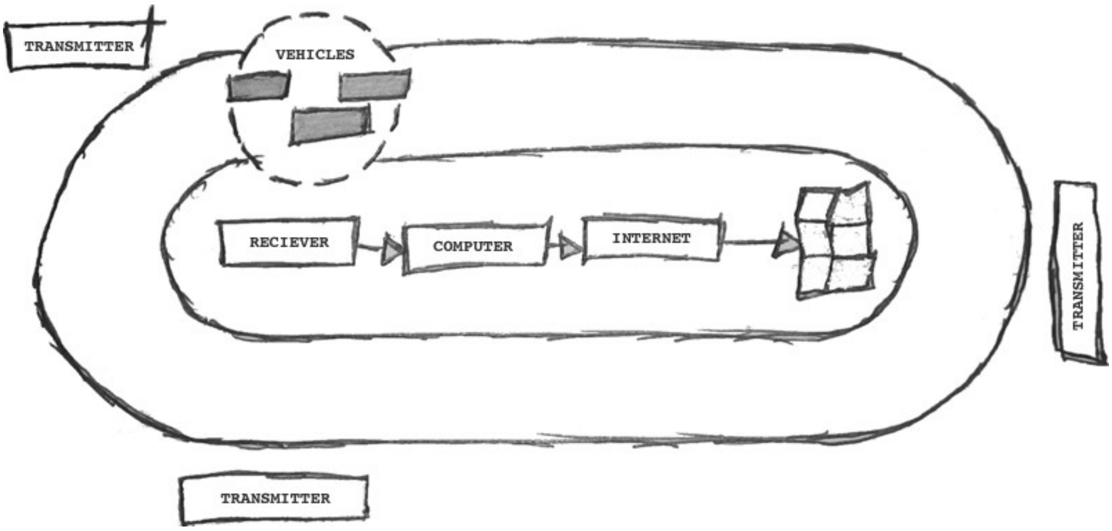
As the *PS* is aimed at tracking an object moving at high speeds, this raises the question of "effectiveness" in terms of how reliable will the chosen technology prove to be regarding both accuracy and latency as both are important metrics when tracking performance. During the literature search a technology called "**D**edicated **S**hort **R**ange **C**ommunications" (DSRC) was discovered that "could" be relevant to the performance issues previously mentioned.
DSRC is described by **[44]-(*L Armstrong, 2000*)** as "*... a short medium range communications service that supports both public safety and private operations in "roadside-to-vehicle" and "vehicle-to-vehicle" communication environments. DSRC is meant to be a compliment to cellular communications by providing very high data transfer rates in circumstances where minimizing latency in the communication link and isolating relatively small communication zones are important.*"
There were few papers concerning DSRC, however, a couple were found containing some interesting factors that can be taken into consideration. DSRC is fundamentally concerned with the support of vehicle safety applications but theories of the technologies have highlighted areas that if not actually used, could be translated across to the proposed system in terms of thought provoking areas.
As discussed by **[44]-(*L Armstrong, 2000*)**, and justified accordingly in additional papers by **[45]-(*J Yin et al, 2004*)**, **[x]-(*RW Hill et al, 2002*)** DSRC uses technologies generally of three parts, road sensors, communication system and an on-board device where the vehicles position is logged in relation to a set area / space that has been defined by the sensors. This gives warning of any impending situations regarding vehicle safety (*congestion, roadworks, stopped vehicles / traffic, low bridges and collisions...*) thus this setup could be translated and the fundamental concepts researched further aiding with the development of the *PS*.

4.11 - Similar and Existing System literatures

Literatures regarding similar and existing systems been reviewed, and it is apparent that GPS is a strong and popular a technology as it is used within each. Selected and favored discovered literatures regarding relevant systems have been selected for further review to both see and understand how they operate, and additionally how they use and incorporate the GPS technology.

4.11.1 - Auto Race Monitoring System

The paper titled "Auto Race Monitoring System" by **[47]-(*J Busack, 1997*)** shows a very similar concept to the one of the *PS.* Very few papers were found having such similarities, therefore this paper is thought to be very relevant to this project*.*
The paper investigates and outlines the best implementation methods of a real-time auto race monitoring system. The paper describes that the system will consist of a ground positioning system transmitting signals to receivers that are installed on the cars instantaneously determining the exact position and altitude of the car in relation to the race circuit as they travel around the race circuit along with further parameters including vehicle (car) speed, engine temperature and oil pressure data that is sent by the "*... means of a transmitter to a receiver ...*" that is "*... interconnected with a mainframe computer ...*" using the collected data to "*... replicate each of the vehicles in the race in real-time*".
The collected data is to be published to the internet where it can be accessed and viewed. Users can select any of the cars participating to view and monitor at any time, additionally offering the ability to listen to the communications between both the driver and the pit crew during the race.



[Figure 10]-(*Auto Race Monitoring System Design*)

The author describes the technical field of this project: "*The invention herein resides in the art of the monitoring devices and system. More particularly, the invention relates to a system for real time monitoring of all the race cars in an auto race. Specifically, the invention allows the user to select a desired vehicle to monitor during the race ...*" allowing the ability "*... to switch from one vehicle to another ... obtaining the perspective from each such vehicle*" **[47]-(*J Busack, 1997*).**

A first impression at this stage is that the paper provides good ideas which are backed up and justified well, however, the author does not write with much enthusiasm thus not giving the reader the feeling or confidence that the ideas and goals can actually be achieved.
There are not many papers found with the main focus and subject being related to actual systems primarily used to track racing vehicles in racing environments, however this paper does so for that reason alone is considered relevant and interesting for further investigation.
The paper is additionally reasonably old and will therefore not be as advance or up-to-date as other related papers and systems. This fact does additionally not eliminate the paper from review, although is taken into consideration when reviewing the paper.

The Author explains well the need for such a system describing the popularity of the sport, and that currently (*when the paper was published*) viewers can only watch their favourite drivers through use of a television which compared to the proposed system of this paper is very limited for many reasons. These include the discussed features of the system and the fact that the viewers can only watch the cars that the camera is focussing on (which is in most cases *the cars leading the race*). This is a good point, however irrelevant for the *PS* and a consideration for further research. This is because the *PS* in its prototype stage will only require to focus of one vehicle (*motorcycle*), although this opens areas for further investigation and future project expansion.

A further disclosure of the project explains the abstract in more detail highlighting again that the system will have the ability to broadcast the collected real-time information to the web, and will not only replicate the exact vehicle location in regards to the race circuit, but additionally the vehicle altitude, which is an interesting metric that could also be very useful and be included within the *PS*. The author becomes more enthusiastic further into the paper showing great confidence in the system continuing to describe how reliable, accurate and precise the system will be and how it will use "*... state of the art apparatus and capabilities*" (at *the time the paper was published*).

A more detailed description of how the system functions is provided in the paper, as follows:
The overall idea of the system fundamentals can be seen in **[figure 10]**. The system is based in a typical racing environment consisting of a race circuit and race vehicles (*in this case cars*). Fixed in position to the circuit is a Global Positioning System comprising of transmitters positioned at selected areas within and around the circuit. The author clarifies what is meant by a Global Positioning System further in the paper explaining that "*... the Global Positioning System is ... a rudimentary GPS ...*" that is both "*... commonly known and widely used*" justifying what other authors have stated in other reviewed literatures within this chapter.
The transmitters send output signals from the fixed locations that are in turn received by the devices installed on the cars additionally generating further signals "*... corresponding to a instantaneous location based upon mathematical triangulation of the signals received form the transmitters*".

The author explains that to obtain the additional data as described previously in the paper, a special DAQ chip (*device*) is to be installed in each of the cars. "*... The DAQ chip which is interconnected with the various monitors and diagnostic systems of the race car*" continuing "*By way of example only, the DAQ chip may acquire information respecting the instantaneous speed of the vehicle, engine temperature, remaining fuel, oil pressure, and ignition efficiencies...*". Additionally this could possibly provide the means of verbal communication between both driver pit crew.
The GPS receivers and DAQ device are all interconnected through a "collecting" transmitter allowing the transmission of all the data to the stationary/fixed receiver (base station). From there the data is sent to a mainframe computer along with the data collected from the DAQ device where the real-time location and altitude of the individual cars are calculated. "*Accordingly, the computer can replicate the entire field of the race, or can isolate on selected areas of that field*". The mainframe computer has an internet connection which allows the collected data to be published and viewed by users on the web where they can see all of the data collected using the described system with the ability to view all of the cars or additionally select and focus and a particular car and/or switching between different cars during a race.

Overall the system described in the paper sounds promising offering a lot of functionalities and abilities. As described the author does not sound too enthusiastic, however appears very confident highlighting and exaggerating items in other areas. Some chapters of the paper contain flakey areas which do not fill the reader with much confidence or interest in the system. For example the author offers the possibility to listen in to driver and pit crew communications but overlooks this area stating that DAQ device will "possibly" provide a means of communication between driver and pit crew and does not explain a real example of how or if this is actually possible. The author also states that the vehicle speed will be obtained by the DAQ device, which is fine, however the vehicle speed could also be obtained by the GPS system that is in place (*which is believed to be more reliable*). The DAQ devices could be more reliable, however this is another example of how the author does not explain or justify why some technologies are chosen over others and how general decisions are made. The vehicle speed is available from two sources so a question (*along with others*) is put forward asking why both sources are not used thus allowing the ability to compare and contrast the speeds obtained further allowing a more reliable result? Perhaps (*and is assumed that*) it is considered not to be necessary (as one source is reliable enough) however this is not discussed or even justified.

In general the system uses reasonable and agreed solutions and technologies as to what would and is expected of such system, including already reviewed technologies and areas form this chapter (*DAQ, GPS and a WSN*) and is therefore considered as a good find backing-up thoughts and ideas in relation to the *PS*.

4.11.2 - Precise GPS Time transfer to a moving Vehicle

This paper identifies, discusses and evaluates in detail advanced experiments on two GPS based time transfer techniques for tracking a mobile object. The use of precise orbit and clock values calculate the accurate time and position post-mission and **D**ifferential **G**lobal **P**ositioning **S**ystem (DGPS) time transfer with an accurate clock and a DGPS reference station.

Like the "Auto Race Monitoring System" paper, this paper is reasonably old , therefore the limitations concerning the review of an older paper will be taken into consideration, affecting further decisions, opinions and directions regarding the progression for the *PS.*

This paper although reasonably old is considered a valuable resource with regards to the GPS technology since the authors **[48]-(*TE Melgrad et al, 1999*)** have many years experience in and around the subject and are currently involved in both GPS related research and current project development.
The paper focusses on the different GPS time transfer techniques between stationary receivers positioned at precise locations with a GPS receiver installed on a moving object, with set accuracy requirements that should work anywhere on earth, thus being (especially during the time of writing) a very demanding system.
It is described that "*... single point precise orbit and clock positioning emerges as the best way of achieving extremely high timing accuracy when computation is performed post-mission*" with a viable alternative being "*... DGPS time transfer using a calibrated time standard at the references station*". Regarding the first alternate, the author describes that "*... Its key advantage is its global coverage without reference stations or communications links*" which differs slightly from the current belief regarding the requirements of the *PS*.
This area will be looked into further to see if such a system could be used maintaining the expected performance If so the requirement of reference stations would be lost thus a different design thinking may be required.

The author explains that due to the context of the application, the study "could" concern the measurement of a moving object on the arrival times of (**LO**ng **R**ange **A**id to **N**avigation (Loran), with the current version being Loran-C) pulses that "*... are required in order to calibrate the **A**dditional **S**econdary **F**actors (ASFs) caused by the propagation of the signal*", with the addition that post-mission processing "may" be required, hence the need for such a (*previously described*) system. However, with regards to the *PS*, post-mission processing would not be a requirement, thus such a system is not necessary, although the performance (*as described in the paper*) is appealing and is something that the PS "could" benefit from.
At this point of the review it is thought that the technologies (*although high performing*) are not needed in terms of the requirements of the *PS.* The question is asked, why complicate areas with additional technology capabilities which are not required and will (*more than likely*) not be used? The author continues to explain that "*Our experiments show that both timing methods CA meet the accuracy requirement.*" and that "*... the techniques presented are, of course also applicable to GPS time transfers between stationary locations*" thus deeming the paper as "again" relevant.

The paper explains the basics of GPS confirming what is explained in other reviewed and found literatures highlighting areas that must be taken into account, "known in advance", and generally catered for (e.g. a signals propagation in the atmosphere is slower than in vacuum and its arrival is delayed in consequence) before performing experiments on the proposed solutions.
The Author explains that (at the time of writing) Loran-C is "*... the worlds most widely-used terrestrial aid to navigation*" and is a mode of operation that is very similar to that of GPS as transmitters positioned at known locations transmit (*radiate*) signals that a receiver "receives" and calculates the distances by timing their arrivals "*... from knowledge of their speed of propagation ...*", however a more recent paper states that the use of this technology is declining rapidly and that GPS is its primary replacement, therefore indicating and again highlighting the fact that the age of the paper needs to be acknowledged during further review.
Due to the age of the paper the fundamental ideas are deemed relevant for further review, however the in depth detail will be disregarded as due to recent advances within the subject area, the paper is assumed of date, and thus to a certain extent irrelevant.
Loran-C is "another" identified area for further investigation. The paper states that GPS is a candidate that can provide both the position and time references required by the experiment to provide precise time regarding a mobile object.

Two additional (more low -level) standard GPS techniques were additionally evaluated, being "Enhanced GPS" and "Geodetic Positioning" both taking advantage of and running on the GPS foundation. The paper discusses these two techniques highlighting their advantages and disadvantages. Both techniques perform well, although both prove unsuitable due to inadequacies found and costs in regards to the application. The following two less conventional techniques were therefore evaluated including **P**recise **S**ingle **P**ositioning (PSP) and DGPS, explaining that with time transfer PSP, "*... employs an orbit and clock values post-mission*" and that "*DGPS time transfer is essentially the same as conventional GPS, however, the reference station is equipped with an accurate clock and generates range corrections instead of the usual pseudorange corrections*". the paper continues to discuss techniques in great detail, which is deemed irrelevant for further review but has opened up additional and further research areas.

The author describes the (*hardware*) requirements of the receiver for the application. The paper describes the two clocks within a receiver, the "hardware clock" (*the externally accessible clock providing high frequency outputs*) and the "software clock" (*which the receiver uses to measure the arrivals of the satellite signals thus calculating the relevant values*), additionally confirming other literatures, stating that the two clocks have to be accurately synchronised or "discrepancy" will be recorded between them.
Referring to previous literatures studied, it is already known that different receivers offer different possibilities and functionalities, thus it is very important that the correct hardware suitable for purpose is selected.
The chosen GPS receiver used for the project (*deemed best for precision in relation to the application*) was the "NovAtel GPS Card 3951R narrow Corrector Receiver".
Novotel have produced many papers regarding tracking mobile objects and have also proved a useful and rich source during the literature search.

After further investigation of the two techniques, the author describes that both are suitable for use with the application. Further, more in depth investigation and tests were performed (*with the "NovAtel" receiver*) which the paper continues to discuss in great detail, including the environment, temperatures and weather conditions, time zones and used times (*UTC Brest*), reference stations, technologies and devices, setup and layout. Any limitations found in which the tests were performed for each technique (*PPS and DGPS*) are well described and discussed to an advance level. It compared both with the different situations (*differing factors*) and environments conditions that affected any results.

After the further tests are completed, the paper discusses the results presenting and comparing them with the initial accuracy requirements in mind. The PPS results were poorer than expected, being not being as accurate as hoped, although it is described that these results were (*not completely*) surprising due to the "*... the relatively high environmental and the low **S**ignal to **N**oise **R**atio (SNR)*" used. The PPS results are discussed explaining unexpected outcomes comparing Graphs and tabular data in terms of positions and times during the experiment.
The paper further describes that the DGPS mean offset was a little less than that of the PPS solution in both longitude and latitude, and the height components are similar, additionally the "*... RMS variations of the DGPS solution are greater than those from PPS*".
The DGPS results are evaluated and overall these results are explained as very satisfactory in terms of performance, thus it was decided to evaluate and experiment further increasing the complexity and test loads including the use of longer base lines. the results from this process again proved very successful with additional improvements being found in terms of accelerating the technology and further improving performance.

The author concludes: "*We have demonstrated that both PSP and DGPS time transfer methods satisfy the requirements set. ... we consider both methods to be excellent candidates for our application*" additionally stating that the NovAtel receiver performed very satisfactorily, additionally justifying the findings form the experiments stating that "*... we have demonstrated that we can achieve timing accuracies commensurate with the positioning accuracies of a GPS reciever ... The performance of the receiver when in motion is known to be similar to its static performance.*" with the addition that the "*... performance of PSP and DGPS and well known and satisfactory which indicates that the results are reproducible*".
The paper further concludes that both techniques are valid and that the results demonstrate that both are suitable for use with there proposed application. "*Peaches orbit and clock solutions give global coverage, and may be used for precise time transfer between GPS receivers at opposite sides of the world, stationary or mobile, in known and unknown locations*".

A lot of valuable information has been learned from this paper, and further areas of investigation have been identified. The application discussed within the paper is not the same as the *PS* although has similarities in that it is used to accurately track a mobile object, although over a much larger scale. Limitations and problematic areas have additionally been identified, and solutions and improvements have been found relating to these areas, thus improving results which can be taken onboard during further project progression.
Overall a greater knowledge and level of understanding regarding the area(s) of using GPS to track the movements of a mobile object have been gained, thus making this a valuable literature source in regards to the foundation for further investigation and research for the project.

4.11.3 - GPS Skiing - With Racing Heart

The paper "With Racing Heart" **[49]-(*J Skaloud et al, 2001*)** is about a GPS based system from Switzerland

used to track high-speed down hill skiers. The system does not only incorporate GPS but also takes advantage of video, biosensors (*for precise positioning*), velocity and heart-rate measurements being used to help both "*... coaches and athletes plot the best line and improve training procedures – while enhancing television broadcasts*".
Referring back to the project proposal, the system described in this literature is very similar sharing related foundations to the PS, although to do with tracking skiers traveling down a slope rather than a motorcycle traveling around a race circuit.

The system has been tested, reliably tracking skiers traveling at speeds of up to 100kph (*60mph*) proving that high speeds can be accurately monitored using such a system (*although the speeds required to be tracked in the PS would potentially be more than double this speed*). The results collected are used for analysis by both skier and coach acting both as a learning/tool-for-improvement (*like the PS's proposal*) and allowing to visual monitoring "*... course profile and technical difficulties encountered*" during a run.

The system requires the skier to wear a device around the waist with a helmet combo containing a GPS receiver integrating "*... GPS positing, video alignment of sequences in time and space, and heart monitoring with internal data recording*" making it possible to precisely measure and compare a skiers performance(*s*) over an extended downhill coarse.
With regards to the *PS*, the heart monitoring equipment is not required, It is however, another area for possible project expansion that has been noted and taken into account.

The paper describes that coaches need to know how and for what reasons racers can improve and perform better at any given time, justifying the need for both this system and the *PS.* It additionally describes that the system offers a **P**osition/**V**elocity/**A**cceleration (PVA) analysis giving the "*... competitive edge of a point-to-point course performance examination and overall profiles between racers own performances and those of other skiers*".

The author justifies the system further stating that "*... coaches and skiers devote long hours to slope recognition analysis, attempting to quantify many factors...*" including "*... the skiers experience on a particular slope, current snow conditions, the skiers ability, weight, and intuition and others*" which again when translated from one (*this*) project to another (*the PS*) the fundamental factors remain very similar. The author continues stating that "*... A skiers race course time only summarises many decisions taken, and until now no mean existed to separate good discussions from bad ones*". The use of GPS and video offer the possibility of the study of each individual in turn, and their integration enabling analysis not only of skier performance, but additionally equipment performance. In relation to the *PS* the suspension and chassis setups and tyre choices are comparable.

The paper goes in to detail describing the advantages of finding the fastest lines that can be taken on a slope and how the system can evaluate and improve these lines, describing that "*... the shortest line may lead to excessive edging (in effect braking), resulting in a slower exit speed from the turn ...*" and that "*... a wider turn may better preserver momentum and speed*" yielding a faster time, which again are all relevant factors when translated to factors of the *PS*.

The paper continues to justify why the system is relevant and how important it can be in the correct hands. A personal view of the paper is that it is correct with all arguments being understood, related to and agreed upon. The paper further discusses how it uses cameras (*SimulCam Functionality*) to blend images taken from a skiers perspective recording a terrain model and blending these images for use with television broadcasting. Although very interesting, again this is an area that is not required (*at this stage*) of development, and further review around this area is deemed irrelevant.

The advantage of GPS with such a system is that is offers an upgrade from qualitative information to quantitate data. "*Finding the precise values of an athletes speed and acceleration provides valuable information on the athletes and equipments performance*". Overall the use of GPS boosts system comparison possibilities "*... by a factor of 10 to 20*".

The paper discusses problematic areas that were encountered when developing the system and during prototype stages. The author states that it was not really a problem collecting the data after a run had commenced, however, the requirements of the system (*especially with television broadcasting*) meant that the data was required real-time as the skier was on the slope. With regards to the *PS,* real-time data is not required (*again at this stage*) so these areas are not discussed further.

The author explains that for useful analysis, the position of the mobile object should ideally be determined with sub-decimeter accuracy, requiring **D**ifferential **C**arrier-Phase GPS (DCGPS). It is described that a rover receiver is mounted onto the mobile device with the addition of a base receiver placed nearby, both sampling and storing the pseudorange data and the carrier phase of the GPS signals (*at 5Hz frequency or higher*), being determined post-mission by constantly evolving algorithms that are designed for high-dynamic applications.

Technology advances have allowed the system to use very small, lightweight receivers not affecting performance when mounted, which is ideal for use with the *PS*. The only limitation (*the author describes)* is that for most receivers (i*ncluding the chosen receiver*) the exact same receiver must be positioned on the base station (CBS) to enable the system to function properly with reliable results, although this is personally not thought or understood as being a limitation as such.
The author describes that the CBS can be positioned anywhere within the vicinity of the slope, however the positioning accuracy is better if the maximum distance between the CBS and the skier is less than 5km, again all being relevant and possible with the *PS*.

The paper explains further (*relevant*) limitations faced during system development in that performance has to be compared and even lost against size as receiver and antenna sizes can and should not restrict skier performance in anyway, however still having to provide decent and reelable results. Larger device tests provided better results although proved to "get-in-the-way" of the skier during the run and additionally in some cases affecting safety issues (t*he skier feeling/being distracted by the hardware*).
The paper continues to discuss limitations and problematic areas and the found solutions to these problems in great detail, proving very interesting and relevant for both thought further reference during project development.

Having studied the paper further, what the author explains regarding the heart monitor appeals to the *PS* and opened up many new opportunities that could make it worthwhile for consideration, even at this prototype stage. The author describes that the use of the heart monitor "*... enables analysis of heart-rate patterns in conjunction with DGPS data of velocity and trajectory throughout the race. Factors such as topography (slope inclination, bumps), technical difficulties (critical turns, jumps), the racers training condition, duration of the race (fatigue), position (extended, "tuck") and attitude of the skier may all play a role*", all being very relevant and very translatable in regards of the PS.

At the end of a ski run, trajectory analysis is performed merging data from both receivers (*mobile and CBS*) and performing "post-processing" obtaining 3D positions and velocities that are taken 10 times per second along the whole trajectory, thus creating vital information regarding a skiers gate-to-gate performance, additionally offering comparisons to other runs highlighting "weak" areas that can used used and work upon for further improvements.

At this stage, the paper continues discussing the hardware used in detail, and how visual rendering is mapped from the camera and processed for broadcast. This is (*as mentioned*) not relevant for review at this level, although is very interesting and again has opened a lot of ideas for future expansion.

Overall, this paper has been far more relevant, informative and useful than was initially thought at the beginning of the review. The system, although different is also similar to the requireemnts of the *PS. A*ll that required is a small translation between the two areas along with some initiative and imagination, and this literature "could" help fo0rm the foundations for the *PS.* This is due to the fact that it has been created, tested and is successful, pushing its progress rapidly in the right direction. The paper was deemed reliable in regards to authors opinions in relation to other papers and therefore considered along with other described factors, a very good resource.

4.11.4 - Real-Time GPS FX – Race-FX

The paper "Real-Time GPS FX" **[50]-(*K Milnes et al, 2002*)** is a very valuable paper found during the literature search being very useful and relevant to the *PS* in terms of progression, direction, potential, ideas, technologies, hardware and software possibilities.
SportVision own the Race-FX system and specialises in and "*... develops technology based enhancements for the Internet, sports television and new media platforms*".
Race-FX (*at the time of writing*) was SportVision's latest innovation that is currently used in the **N**ational **A**ssociation of **S**tock **C**ar **A**utomotive (NASCAR) and Indy-Car racing today, described as "*... a system that incorporates GPS and other technologies to enable real-time tracking and display of the location of all cars throughout a racing event*". The paper also mentions that the system was developed in conjunction (*again*) with Novatel Inc (*GPS technology Supplier*) a company already discovered and discussed within the review.

The RCAF system is very similar to that discussed "With Racing Heart" skiing system, using GPS technologies to track a mobile object in real-time with the additional video overlaying for television broadcasting.

The paper explains how the system differs form the *PS* in the sense that it has been developed as an entertainment system aimed at viewers of races rather than a learning/improvement aid. The Race-FX system gives broadcasters new tools and techniques of showing races, thus increasing the enjoyment factor for fans.
The system is, as mentioned used by NASCAR during every event to "*... generate graphics, calculate speeds and compute other performance-related parameters of interest to the racing fan*" using high-performance GPS receivers providing real-time measurement of object position(s).

A lot of the content of the paper focusses on the video recording from each car and the overlapping of the images created used for television broadcasting. This feature is highly praised throughout the paper and the advantages that it brings to such a system are agreed upon at a personal level. This feature would greatly benefit the PS and would enhance the system extensively, however, this area is thought to be relevant to future expansion and at this (*prototype*) stage of development, and as with the "With Racing Heart" skiing system, this area is believed not to be necessary and thus deemed irrelevant for further review.

The author describes how auto racing has traditionally used "inductive sensors" with such systems measuring lap times and speeds by detecting when a vehicle has crossed the start/finish line, additionally describing that the use of GPS with such system is more accurate allowing the creation of virtual lines by "*... defining positions along the track, such as turns and straight-ways*" thus allowing to see which lines a driver/rider is fastest along during different sections of a racetrack plus the collection of other rich statistical information.
The author describes the development of the system as very challenging, posing many complexities to developers, including "a*ccurate vehicle positions needing to be obtained, calculated, and transmitted during a high-dynamic operations under racing conditions in which GPS satellite signals are frequently blocked or reflected (multi-path)*" justifying what has been discussed and reviewed in other literatures within this chapter.

The paper describes the system in detail explaining that the main four subsystems contained within the RCAF system are GPS, telemetry, time synchronisation and video overlay. The paper describes that each car has a receiver and transceiver installed with mating transceivers remotely located around the racetrack sending data over DSL modems to the control center (CBS) where the communications controller, time synchronisation and video overlay systems reside. The paper continues to describe that the system "*... employs a sophisticated telemetry system that transfers position and other vehicle information from all race vehicles to a central processor at the rate of 5 times per second. DGPS pseudorange and carrier phase tracking techniques generate vehicle coordinates accurate to 50 centimeters (one stigma). The tenementary conveys differential messages from GPS base stations to the car rover units at 0.5 Hertz and car rover information to the video system at 5 Hertz*".
The system processes the 5 Hertz DGPS positioning data to calculate useful parameters. The data packet being sent must be kept to a minimum keeping performance to a maximum, with the author describing that speed and heading are calculated from the computed velocity vector between GPS positions as well as lateral and along track acceleration.

To obtain data such as pan, tilt, zoom and focus the system uses cameras, interpolating position information to correspond with the camera orientation in each video frame with high-speed computers combining "*... this data to appropriately juxtapose the car and data in the video frame*".
System requirements are described including that the cars travel up to 90 meter/second (*200mph, approx.*), the speed accurate to those that the motorcycles within the *PS* will be traveling. The relative timing between the video and GPS must be accurate to one millisecond keeping time-included errors below 10cm, thus being a very complex and demanding system requiring reliable, robust and accurate technologies to enable successful results. The author describes that "*... the GPS based system maintains the timing to about 10 microseconds, or 100 times better than the minimum requirement*" so the technologies used above are satisfactory, and therefore theoretically also suitable for use with the *PS*.

The paper describes more about the GPS system, explaining that the tracking of satellites and computing accurate vehicle positions at such speeds does stress a conventional GPS system. As other literatures, the author further describes that the greatest challenge when using GPS to maintain accurate position fixes on a race circuit is obtaining visibility of the the minimum number of GPS

satellites required for position determination.
The paper justifies other discussed areas and project worries, explaining that although four satellites are the minimum number required for 3D positioning, "*... in constrained environments such are a racetrack four satellites often do not have the necessary "geometry" to obtain the accuracy required for the Race-FX application*". additionally explaining that satellite geometry translates into GPS **D**ilution **O**f **P**osition (DOP) acting as a "multiplier" of any errors that have been generated by factors such as receiver electronics, ionospheric effects and multi-path.

As discussed previously in this chapter the author additionally justifies that within a racing environment many obstacles exist, including buildings, grandstands, fences and walking bridges blocking GPS rover units view of the satellites.
Usually a GPS reference station is installed for generating and transmitting any differential corrections and is positioned on top of grandstands or buildings thus having a clear view of the sky and therefore the satellites. However, as the paper describes "*... for vehicles operating on the track, the various ...*" discussed "*... obstructions frequently combine to reduce the number of visible satellites to as few as 3*" having a poor DOP, thus disregarding the accuracy of the position data. Furthermore, due to the vehicle speeds, and in many cases the short laps and times, these obstructions are coming in and out of play during the entire race further reducing accuracy and signal availability.

The paper discusses in detail the data, hardware and communications used with the Race-FX system, describing that each car within a race is additionally fitted with a **D**ata **A**cquisition and **P**ositioning **S**ystem (DAPS) containing a 12 channel, dual frequency GPS receivers, a 900MHz spread spectrum, a 486 computer and some interface electronics.
The radio modem transmits GPS position information from the cars to the CBS and from the CBS to the rover unit on the car. "*A single computer in the Race-FX control center and the communications controller, manages all of the data between the telemetry and GPS receiver and sample sensors attached to the cars (RPM, throttle position etc.)*". The paper explains that batteries are used to provide power to the system (*thus bringing up oversold described limitations*), and that the whole package is mounted to the roll-cage of the cars with vibration isolation mounts preventing phase noise in the oscillator signal that may create satellite tracking failures when such vehicles experience between 6-18 G's of vibration.

The paper describes how the system uses a specially designed telemetry methods providing continuous coverage between the cars and the telemetry stations. The data from the cars must be transmitted in a predictable, timely and error free manner.

As discussed racetracks can have obstructions blocking satellite visibility from any one location to points on the racetrack. A solution to these obstructions, the paper describes, is this telemetry system, consisting of 3 to 4 base stations (*one of which acting as the master station, master synchroniser*) located at different positions around the racetrack broadcasting control information and GPS differential correction to all rovers in a "time-slotted" manner, providing continuos communication between the cars and stations.

The paper is very detailed and in depth with a lot of areas being irrelevant for discussion in this chapter. The Race-FX system has similarities to the *PS*, and a lot can be taken and learned from the system. One of the controlling factors and means of the Race-FX system is its visual overlay possibility being one of the systems main features additionally bringing the main complexities to the system. A lot regarding these features and complexities are discussed in the paper although are not relevant, however are believed very interesting and possibilities of future project progression.

The Race-FX paper is considered relevant and useful to the PS due to its similarities and possibilities, and is also considered a very reliable source as the developers have had a lot of experience in this subject area, and the system has been rigorously tested. It is used today on a regular basis in both NASCAR and Indy car Racing. The system "*... has demonstrated itself under commercial operational conditions as a graphic system capable of displaying video annotation of NASCAR races to a television audience in real time. The core of the system is set on modified GPS receivers capable of operating in either pseudorange or RTK differential mode with the additional capability of incorporating a a digital clock into either of the positioning solutions*". The author continues explaining that the system meets all initial requirements, and that although the technology has been "*... tailored to a particular racing application, there is no reason why it could not be applied to other navigation problems*".

A very valuable source for further project development advice and guidance has been established from this paper. A contact with both the paper author and developer of the Race-FX system "Ken Milnes" has been established, with them offering advice on many literatures, areas and pointers for further project progression.

Ken Milnes "*... is a senior scientist and project manager at SportVision, Inc... As a cofounder of Etak Inc, he performed extensive work with dead reckoning and digital mapping algorithms for land navigation. at SRI International, he did research of HF radar systems. He received a B.S. in electrical engineering and Computer Science from the university of California at Berkeley*", so is considered a valuable source relevant to this project.

4.12 - Literature Review Conclusion

At the end of the literature search it is clear and apparent that many literatures exist in and around the subject area all offering similar and differing opinions on many of the relevant technologies available.
For the required design and implementation of the *PS* prototype, it is thought that (*at this stage*) one of two possible directions could be taken. Based on the reviewed system and literatures "mixing-and-matching" advantageous areas related to the *PS thus* forming a hybrid styled foundation that is believed to be a sufficient and relevant direction to pursue for further research and development.

It is considered that (*in summarisation*) either a WSN could be used implementing some sort of sensor based system deploying sensors around a racetrack and a lightweight mobile device positioned on the motorcycle in communication with the deployed sensors. Additionally using "repeaters" boosting/amplifying signals from the sensors that may be far away from the CBS. These form gateways allowing the sensors to relay relevant data back to the CBS. This system will rely upon both time synchronisation and localisation with the synchronisation being responsible for the local clock times of the sensors collecting data from the mobile device with the clock time of the CBS, and the localisation being responsible for ensuring that the positions of the individual deployed sensors are known and accurate to the exact geographical coordinates of the motorcycle at different points around a racetrack.
Additionally satellite based technologies could be adopted with the use of a GPS receiver and the racetrack (*possibly and potentially "could" be*) embedded within graphical mapping software. The GPS receiver would be positioned on the motorcycle along with a relevant modem (GSM/GPRS) enabling the identification of the motorcycle in relation to the racetrack communicating this data back to the CBS where it can be processed into meaningful information and pushed and used for analysis within a front-end application.
An advance DAQ system can also be installed on the motorcycle measuring many additional metrics including RPM, throttle positions, current gear positions, passing this data in data packets to be matched and synchronised with the location statistics thus enhancing and improving further analysis.

Much has been discussed within this chapter allowing plenty of scope for additional research along the development cycle of the PS. The literatures studied have provided a great advantage and "body of knowledge" regarding possible uses of varying technologies, additionally justifying these solutions with other authors and schools of thought regarding the subject area.

Overall, the this chapter has proved very valuable. A lot of literatures, sources and information has been discovered along with many different areas (*technologies, architectures, methods, devices, platforms and techniques*) that will greatly enhance, improve and help further research into the perfect solution(s) in relation to the *PS*.
During the literature search, not only valuable literatures (papers, books, websites...) have been found, but additionally a number of contacts have been established through communications regarding interest and personal questions regarding the reviewed literature within this chapter.

5 - Research and Investigation

5.1 - Introduction

This chapter will discuss and extend key areas which are believed to be required to enable successful development of the project. These will include areas that have been identified in both the **P**roject **P**roposal (PP) and **L**iterature **R**eview (LR).
Within the PP the project goals (*aims and objectives*) have been outlined and will be cross referenced throughout this chapter, acting as stable guidelines (*hinting at what is required for successful results*) that will direct the research process into the correct and relevant areas.
Many in depth areas, come of them new, have been discovered within the LR and will be covered further within this chapter.
The LR explores and discusses sources to a level thought necessary to the purpose of the review, however, some of these sources have been highlighted in (*and even removed from*) the review as further research and investigation. These discussed areas have been put on hold for discussion within this chapter due to both time constraints and the level of detail required. Covering these topics in the review may have sent it "off-track" possibly affecting the reviews balance.
The LR has opened up, and left open investigation leads which will be "picked-up" and followed further, tying any discussed loose ends. This will overall produce a better, more knowledgeable and meaningful understanding of each area thus (*potentially*) making for an expanded, detailed and extensive chapter.

The discussed and selected areas from previous chapters, along with any new areas found during this chapter will be researched and investigated entering into deeper and more advanced levels of detail, allowing the examination of subjects at a more "practical" level aiding for a simpler, clearer and smoother (*solution*) selection process.
The bulk of the chapter will focus on grouping areas of the same topic providing detailed profiles of each, exposing strengths and weaknesses and summarising, evaluating and comparing each, thus helping the realisation of what is required, potentially producing solutions for the final project compound.

Continuing from the LR, selected hardware, methods and technologies (*elements*) that "could" be required to track positioning information from a high speed object (*motorcycle*) will be selected that are considered relevant and best suited for use in relation to the **P**roposed **S**ystem (PS) comparing them against one another, always taking into account the environmental setting(*s*) of the PS.

As discussed in previous chapters, one of the most challenging areas of this project is considered to be the process of obtaining accurate and precise positioning/location information (*coordinates*) of a high speed motorcycle during a lap of a racetrack.
There are many types of hardware and techniques available, therefore a selection process has been exhausted, consisting of a simple "fly-through" scanning process, covering (*selecting and eliminating*) *elements* that are fundamentally believed relevant to the PS, as well as similar and existing *elements* used within systems, and additionally (*and more importantly*), *elements* that were found and believed relevant during the literature search.

The chapter will commence by investigating two papers that were found during the literature search regarding **L**ocation **B**ased **S**ystems (LBS). These were deliberately left out of the LR for particular reasons mainly the relevance level of the inclusion of such papers within such a chapter, and also the level of detail and discussion (*investigation*) that is required for the understanding of such source subjects.
The content of these papers is not believed necessary to enable the development of such a system (*the knowledge gained from such sources will not prevent the development of the PS*), however, it is believed that the knowledge gained from the content will help the understanding of the underlaying method(s) used when tracking a mobile object. With this understanding "on-board" many opinions, views or decisions that are to be made further throughout the investigation and selection process may change.

5.2 - Location Based Systems

Many sensor based systems and technologies have been discovered, and it has become apparent that there is no clear individual solution that can be used for calculating location and position based information. The resulting systems rely on additional sensor based, and other technologies (*mainly GPS related*) working together (*in many cases in an adapted manner*) to meet particular system requirements.

The chapter will commence with a paper discovered during the literature search regarding "*multi-sensor location tracking*" that was excluded from the LR and postponed for discussion and further investigation within this chapter. Within the paper the author proposes an expansion to such a (*relevant*) system architecture regarding the acquisition of locational data from mobile objects, introducing an additional (*acquisition*) layer of indirection between both the application and the implemented system sensors.
The paper details the acquisition layer (*a dominant layer within such a system*) in great depth discussing algorithms and mathematical fundamentals required for positional information calculation. It further proposes (*author specific*) algorithms with modifications and advances to current and existing algorithms used within such systems.
The paper is considered a valuable source in terms of the understanding and knowledge of the processes/calculations required "behind the scenes" of such systems (*devices and technologies*) and is therefore investigated further with the aim of providing a higher level of not only understandings, but also the appreciation of such formations.

5.2.1 - Multi-Sensor Location Tracking

The discussed paper is from the author **[51]-(*J Magee, 2007*)** and contains valuable detail and information regarding a LBS architecture and design considerations that are believed relevant for further investigation to help obtain a good level of understanding before further supplementary investigation commences into LBS.

Due to the previously discussed (*within the project proposal*) demand and technological advancements there are many large scale positioning systems existing today, and it is likely that even more ubiquitous systems will appear in the future.
There is currently no particular platform or infrastructure existing to build upon in relation to such LBS, as most systems, like the PS, are focussed towards fulfilling *particular* application requirements or are based upon *particular* sensor based technologies.
This lack of infrastructure makes it difficult to develop location based applications without knowing the fundamental and underlying sensor technology that will be in place, therefore, both technologies and hardware should be sufficiently researched, known and understood before any development process can commence. Some applications require differing measurements and levels of detail(s) from specific LBS, thus a system may be "more" or "less" relevant depending on an applications requirements.
[51]-(*J Magee, 2007*) describes that "... u*nfortunately there is no single perfect positioning technology, so that often multiple sensor systems have to be combined to meet applications requirements*" justifying personal thoughts and opinions regarding the related technologies.

A layer of indirection is required between location aware applications and location sensors acting as a "support" service "*... especially if sensors and applications reside on different nodes in an open distributed system*" **[51]-(*J Magee, 2007*)** thus tracking the physical position of a mobile object, with an additional acquisition function providing a layer of abstraction. Tihs further allows layers to be implemented to a higher level without having to rely on (*be restricted to*) a particular technology, furthermore, this will provide a means to achieve the maximum "spatio-temporal" resolution supported by the input of, and from specific location sensors.

The main concentration (*current challenging areas and advances*) within LBS is improving real-time data collection and acquisition (*latency*) along with the continuously improving **3**-**D**imensional (3D) positioning information. Both of these are not required at this prototype stage of development, however, it proves that most location based technologies are (*or at least should be*) at a more than satisfactory level to those required to meet the documented PS requirements and should therefore should not be disregarded as they could be necessary for future and further project progression and expansion.

In relation to the PS requirements, the discussed acquisition layer is regarded as an import factor to be understood in terms of what is required from a system in relation to the selection of hardware, technologies, calculations and design considerations required for such an LBS.

5.2.2 - Location Based System Architecture

When designing and "speccing" (*choosing specifications for areas within*) a LBS, various issues need to be addressed and taken on board, including:

- **Positing verses Tracking**: "*An Acquisition function can be used in order to build either a positioning or tracking system. A positioning system measures its own location with the help of the infrastructure. A tracking system measures the location of other located objects. Different architectures are required for each case, although the processing functions may be very similar*".
- **Local verses Remote Measurement**: "*Tracking systems may be built on top of positioning systems and vice versa. In such cases, the acquisition layer has the task of providing a specialised location transparency. This also requires physical and logical distribution of the acquisition function*".
- **Synchronous verses Asynchronous dissemination**: "*Applications are likely to require both events and polling, while location sensors may only support one of the two*".
- ***Discrete update verses Continuous change****: Location sensing is mostly opportunistic, with the exception of integrated multi-sensor positioning systems which can provide continuos information*".
- **Stateful verses Stateless sources**: "*Location sources can also be classified into stateless and stateful sources. Stateless sources need to wait for the underlying hardware to supply data, whereas stateful sources can provide information continuously*".
- **Homogeneous verses Heterogeneous sensors**: "*Sensor types are often complimentary in their spatial and temporal coverage. Also, dissimilarity of their error profiles makes simultaneous failure unlikely. Therefore the architecture of the acquisition function must be designed to cope with a variety of sensor types with different computational and communication characteristics*".

[51]-(*J Magee, 2007*)

Taking the above dimensions into consideration and translating them into suitable architectural foundations in terms of the PS, the system is believed to follow the path of a local positioning system using a stateless infrastructure. The use of a "symbolic location model" providing desecrate updates using an asynchronous event based system which is distributed covering multiple sensors of the same type would be required.
With this in mind, further investigation can be continued in the right direction, focussing on areas relevant to the PS requirements. This would further eliminate irrelevant areas thus speeding up the development and learning process making for a more efficient project completion and end result.

5.2.3 - Data Acquisition Stack

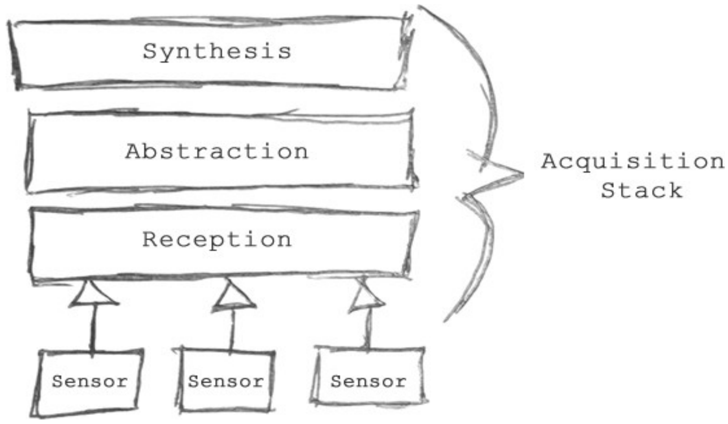
The **D**ata **A**c**Q**uisition (DAQ) structure of an LBS is responsible for the collection of data from the sensors in place and used within a system, passing and presenting this data to higher levels of the application. Such a structure is described (*and known by researchers*) as a "layered" or "stacked" structure.
The DAQ layer is a recursive layer allowing inputs to be received from either the sensors themselves or other DAQ layers (*if available*) allowing and creating a DAQ tree containing multiple stages.
In general the DAQ stack is independent of other layers and thus can be expressed in a number of architectural styles relevant to and depending on a particular systems requirements.

[51]-(*J Magee, 2007*) describes the architecture of a DAQ layer, explaining how it can be split down into three individual layers, including:

- The Reception Layer
- The Abstraction Layer
- The Fusion (*Synthesis*) Layer

The **Reception Layer** provides a "sensor-bus" that is used to support distribution and synchronisation transparency regarding a "communication substrate" for sensors within a system. The acquisition layer removes any limitations regarding the identity of a sensor, as the sensor may be attached to locations that are in-turn "sensor dependent". In the case of positional based sensors (*like that of the PS*) this layer is additionally responsible for hiding these dependencies by (*in most cases*) providing wireless communication solutions/mediums.
The **Abstraction Layer** is responsible for consolidating the sensor data representations, thus requiring sensor independent models. With the requirements of the PS, it is thought that only one type of sensor

is required, however there is still a need for such an abstraction layer in regards to further project expansion. Even if not included, such layer is inexpensive as it contains no additional disadvantages or costs thuds opening future enhancement possibilities.
The **Fusion** (***Synthesis***) **Layer** is responsible for the collection, calculation and correlation of the different sightings of mobile objects that come from the displaced sources/sensors. As discussed the abstraction layer would have constructed a "homogeneous" representation previously to this layer being called, therefore the main responsibilities of this layer is to highlight and detect any overlapping or inconsistent sightings.



[Figure 11]-(*Basic Acquisition Stack*)

[Figure 11] shows a basic acquisition stack compromising of the three individual layers (*Synthesis, Abstraction and Reception*) with an additional and separate layer consisting of individual sensors. This *sensor* layer comprises firmware (*hardware and low-level software*) responsible for the positioning sensors within such LBS and is where any low-level protocols would or could be implemented (*for example the satellite-to-receiver protocol within a GPS system*).

An interesting and valuable mathematical model is presented by **[51]-(*J Magee, 2007*)** detailing the data flow between processing of the individual layers within the DAQ stack. **[51]-(*J Magee, 2007*)** further highlights and expands upon these equations explaining how they are communicated together creating a "high level" DAQ stack structure and algorithm(s) for LBS.
The paper further describes in detail an existing and successful concrete DAQ abstraction algorithm and uses this algorithm as a "building-block" and background upon which to base and propose a new and enhanced DAQ algorithm.

The discussed mathematical model, along with further and more detailed information on (*and including*) relevant algorithms regarding the DAQ stack and containing layers used within a LBS, can be found in the paper from **[51]-(*J Magee, 2007*)-(*Location Based Mathematical DAQ model*)**.

The model prosed by **[51]-(*J Magee, 2007*)** has been prototyped, implemented and tested in a combination of different and relevant sensor LBS scenarios showing successful (*prototype level*) results. The model was encapsulated within a sub-system of the system categories tested, being exposed to a network as a fully functional service using tests pushing the implemented algorithms to a satisfactory limit.
An example of one of the test implementations consisted of an ad-hoc system containing a client application that was connected to all available location services synchronously "fusing" and abstracting the data client-side, additionally storing the collected results (*location information*) in a back-end **O**bject-relational **D**ata**B**ase (ODB) thus in all, providing a specialised DB location service.
Areas for improvements were found within each system, however, in general the model proved successful with a lot of potential additionally highlighting areas for future work and improvements.
Further implementations were performed and tested including an asynchronous setup, and although proving more tedious, the model held out producing accurate and reliable results.

In relation to the PS, this paper has provided understanding with regard to the depth of knowledge, advanced calculations and algorithms required to calculate location based positions and coordinates of mobile objects using sensor based technologies.

The proposed model, like most location based models was that of a **hierarchical** nature (*which will be further investigated within this chapter)*. There are many related and additional existing sensor location tracking systems in use each with slightly different dimensions and algorithms that are constructed towards particular sensors and thus different usages and system requirements.

5.3 - Efficient Location Tracking Using Sensor Networks

Another paper which was discovered during the literature search and postponed for similar reasons to that of the **[51]-(J Magee, 2007)** paper, is a paper from the author **[52]-(H.T. Kung et al, 2004).** It describes in more detail the method of a hierarchical based mobile object tracking model.

The paper discusses in detail two hierarchical based tracking methods, including a "publish-and-subscribe" tracking method called **S**caleable **T**racking **U**sing **N**etworked **S**ensors (STUN) and a method called **D**rain-**A**nd-**B**alance (DAB).

[52]-(H.T. Kung et al, 2004) justifies that progression within the wireless domain, technologies and "*... miniaturisation has allowed researchers to build networked sensors, increasingly compact devices that combine the functionality of sensors, radios, and processors*" further describing that "*... their low cost and wireless communication capability make it feasible to deploy them in large numbers, and without a pre-existing infrastructure*", further justifying personal beliefs and additionally contradicting slightly the belief from **[51]-(J Magee, 2007)** regarding the requirement for an infrastructure within a mobile environment. **[52]-(H.T. Kung et al, 2004)** further explains that with more sensors available "*... in the environment, it is likely that phenomena of interest are near some sensors, thereby leading to the main appeal of wireless sensors compared to the tethered ones*".

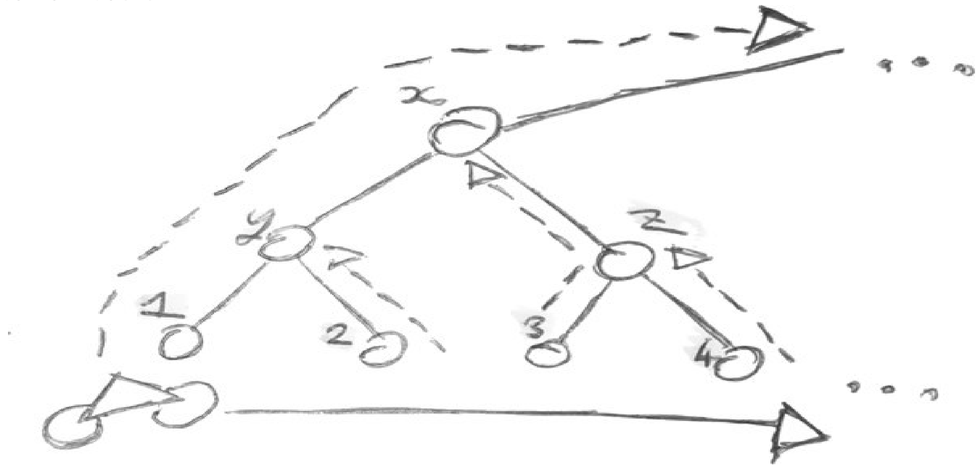
The tracking of mobile objects generally consists of the detection and monitoring of locations of "real-world" objects, fundamentally and possibly using "*... several types of sensing such as acoustic, seismic, electromagnetic ...*" **[52]-(H.T. Kung et al, 2004)**, with many examples of such systems currently and successfully in use (*air traffic control, fleet tracking, habitat monitoring and mobile telephony to name but a few*).
The focus of discussion within the **[52]-(H.T. Kung et al, 2004)** paper concerns an efficient tracking method using network sensor technologies and could therefore potentially be applied to and used within the PS. The paper focusses on the ability of covering large areas of interest with the use of many sensors with a small detection range, with the addition of the ability to track a large number of moving objects at any one time. Although the latter requirement is not necessary regarding the PS requirements, the paper is still relevant for further investigation as the methods describes can be used to track a single object (*as well as multiple objects*) and this additional possibility further enhances future project progression.
[52]-(H.T. Kung et al, 2004) describes that like existing and similar schemes, a hierarchical organisational approach is the best way to proceed addressing and meeting such required scaleability goals.

5.3.1 - Scaleable Tracking Using Networked Sensors (STUN)

"*Consider a set of objects moving through are certain region of interest, where a number of short-range wireless sensors has been deployed for direction purposes. The distributed tracking problems we are interested in solving is about communicating the locations of the detected objects from the sensors to a querying point, where the information is further utilised by the user*". **[52]-(H.T. Kung et al, 2004)**

In a sense there is a known (*average*) speed and limit of a mobile object (*depending on the type of object*) with some being faster than others. Once an idea of this limit is established the "maximum" distance in which an object can travel in a certain amount of time can be bounded and calculated. The STUN method takes such an approach and thus rests on the observation that objects being tracked are moving in a "predictable" manner. This is also the case of the PS, as the speeds of a motorcycle being tracked (*depending on the type of motorcycle*) are known (*or can be obtained*), and due to the environment (*a known racetrack*) that the motorcycle is being tracked along (*allowing for all the directional changes of the racetrack*) a predictable route can be acquired binding and calculating this information with the obtained speed information.
This described approach uses a hierarchical based process, recording an objects "presence" information. Since "*... the movements are limited, keeping this information up to date usually requires updating nodes near the bottom of the hierarchy*" allowing a system to track objects as information travels to the querying point (*thus enabling and expanding the options for tracking multiple objects*).
The STUN system uses a hierarchical tracking method connecting the different sensors within a system using the discussed querying point as the "root" of the structure. **[52]-(H.T. Kung et al, 2004)** describes such a structure nicely as a "Hierarchy Tree".

The sensors within a system make up the leaves of a tree, with the other nodes being communication nodes, known as "intermediate nodes". The intermediate nodes are where information regarding the presence of a detected object is stored. Each intermediate node stores the set of objects that were detected by its descendants, known as the "detected set", thus the detected set of a sensor at the leaf of the tree consist only of the objects that are within the sensors detection range whereas the detection range of the root node of the tree contains all objects that are present within an environment (*single or multiple*).
Such an hierarchal solution is very similar in comparison to other "direct" tracking methods (*like inline methods*). The hierarchal method sends dictation messages from the leaves to the tree root, however, when using a hierarchal solution the messages do not always need to travel directly to and communicate with the tree root, as intermediate nodes process these messages "intermediately". The messages are passed further only if the detected set of the message has been modified, otherwise the message is terminated as it will not modify any further ancestor nodes. This process of elimination of redundant messages (message-pruning) is where such a structure advances in efficiency due to its lower communication costs.



[Figure 12]-(Message-pruning hierarchy)
[52]-(H.T. Kung et al, 2004)

[Figure 12] shows a hierarchical message pruning method. The figure is simplified from a complete method process in that it it only showing the messages of the arrival of a mobile object.
The figure includes a row of sensors covering an area that are all descendant from the intermediate node **X**. The sensors will detect the mobile objects arrival and departure as the object moves through the areas that the sensors cover, thus generating corresponding messages.
For example, messages from sensor **a** will trigger **Y** and its ancestors thus adding the mobile object to its detection sets. "*Subsequent messages form all other sensors do not result in additions to* **X**'s *detction set, and thus* **X** *does not forward them to its ancestors*". This process is consistent throughout the system in the case that sensor **c** will add the mobile object to the detection set of **Z** forwarding this message to **X**, however stopping at **X** and thus leaving the detection set of **X** unchanged. "*Messages from sensors...*" **b** and **d** "*... do not change the detected sets of their parents, and thus do not propagate past them*" **[52]-(H.T. Kung et al, 2004)**.

Queries are routed from the top of such a hierarchical structure via a single path towards the individual sensors that initially reported the detection of the mobile object, thus without the information being stored within the detected sets, the queries would be required to be sent to all leaves within the tree, proving very costly. The maintenance therefore performed within the system by the detected sets allows for very efficient query processing.
"*This method can be viewed as a sort of publish-and-subscribe mechanism. The sensors publish the hierarchy, while the querying points are subscribers who want to track current locations of the desired objects*" **[52]-(H.T. Kung et al, 2004)**.

[52]-(H.T. Kung et al, 2004) describes the STUN method in great detail exposing the fundamental mathematics supporting the process, additionally discussing in depth performance constraints and communication costs that can exist. For further and more detailed (*more in depth*) information regarding the STUN method, refer to the paper **[52]-(H.T. Kung et al, 2004)**.

5.3.2 - Drain-And-Balance Method

Drain-**A**nd-**B**alance (DAB) is a method that is used to create message-pruning hierarchy trees (*as used within the STUN system*). The DAB method designed with efficiency in mind have the requirements of keeping both query delay and communication costs to a minimum.

Before investigation is continued into the DAB method, a more traditional method with similar concepts to that of the DAB method will be investigated since it is believed to be a good foundation for such methods to be both expanded and built upon.

5.3.2.1 - Huffman Tree

A common, popular and similar method to that of the DAB method is the "Huffman Tree" method. The Huffman method varies slightly from conventional hierarchical structures in the sense that it creates structures in a reverse order (*from the leaves to the root*). The basic technique of the Huffman method is to create a binary tree consisting of nodes which are stored within an array. The length (*n*) of the array depends on the number of symbols required. Like the STUN method, nodes within the tree can either be leaf nodes or intermediate nodes. Leaf nodes themselves contain the symbol, the weight of the symbol and a link to the parent node (*a node containing child nodes assigning a reference from itself to each of its children*) thus making it easier to read the tree starting from the leaf nodes (*in reverse order*). Internal nodes contain a symbols weight, links to two child nodes and (*like leaf nodes*) a link to its parent node. The two children of a node generally contain the bit values "0" and "1", with "0" representing the left hand child, and "1" representing the right hand child, therefore a complete tree structure consists of *n* leaf nodes and (*n-1*) internal nodes.

The Huffman process "*... begins with the leaf nodes containing the probabilities of the symbol that they represent, then a new node whose children are the 2 nodes with the smallest probability is created such that the new nodes probability is equal to the sum of the children's probability*" **[53]-(R Schak, 1994)**. The procedure of merging the current two nodes (thus considering them as a single node) can be repeated up the structure until only one node is left, with the remaining node being the root node of the complete tree structure.

Huffman himself summarises the method as "*... an optimum method of coding an ensemble of messages consisting of a finite number of members is developed. A minimum redundancy code is one constructed in such a way that the average number of coding digits per message is minimized*" **[54]-(D.A Huffman, 1952)**.

The algorithm used within the Huffman process in its simplest form uses a "priority queue". Within the queue the node with the lowest probability will be given the highest priority. For example, a leaf node is created for each symbol within the structure and is added to the priority queue. If there are no nodes within the queue then the structure is considered "almost" complete and the remaining node is added to the structure as the tree root node. If nodes exist within the queue then the structure is considered incomplete. The node with the highest priority (*lowest probability*) is removed thus leaving two nodes which are further used to create a new internal node, using these two selected nodes as children. The probability of the created internal node will be set equal to the sum of its two child node probabilities together. The created node is then added to the structure, and the process is repeated until only the (*described*) root node is remains.

The above example is one of many processes that can be used to create a Huffman tree structure. More advance methods can use more queues assessing the priorities of the nodes depending on additional factors such as weights (*both initial and combined*) pushing these factors into a separate array(s) where an array comparison is performed. These identify the the node priorities thus allowing them to be added to the tree stack enabling the creation of the tree structure.

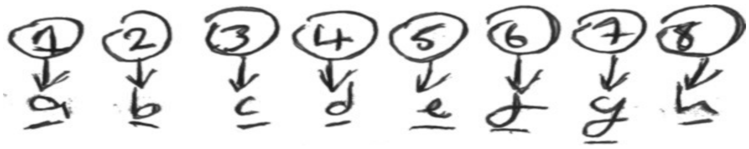
5.3.2.2 – Huffman tree example

Below is a simple example of the creation of a basic Huffman Tree structure. The structure uses the following example table of letter frequencies:

a	b	c	d	e	f	g	h
1	2	3	4	5	6	7	8

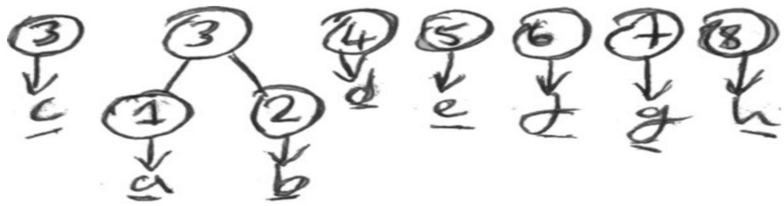
"*Each box indicates the contents of a priority queue of trees, where trees with lower frequency values have higher priority. At each step, the two smallest frequency trees are merged to form a higher frequency tree. This continues until there is only one tree*" **[55]-(L Turbak, 2001)** representing the root bode of the structure.

The following 8 tree diagrams show the 8 steps that are required to create the Huffman Tree structure for the above letter frequency example:



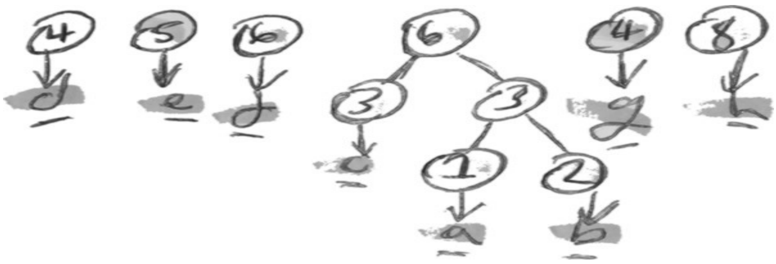
Key: (**1**=>**a**, **2**=>**b**, **3**=>**c**, **4**=>**d**, **5**=>**e**, **6**=>**f**, **7**=>**g**, **8**=>**h**)

[Figure 13]-(Huffman Example Step One)



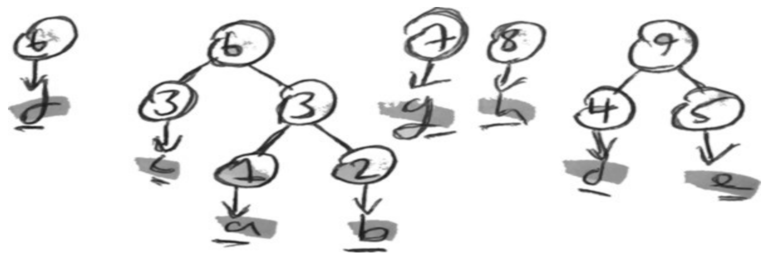
Key: (**3**=>**c**, **3**(**1**=>**a**, **2**=>**b**), **4**=>**d**, **5**=>**e**, **6**=>**f**, **7**=>**g**, **8**=>**h**)

[Figure 14]-(Huffman Example Step Two)



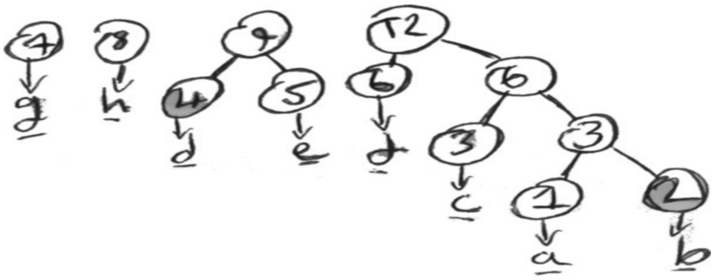
Key: (**4**=>**d**, **5**=>**c**, **6**=>**f**, **6**(**3**=>**c**, **3**(**1**=>**a**, **2**=>**b**)), **4**=>**g**, **8**=>**h**)

[Figure 15]-(Huffman Example Step Three)



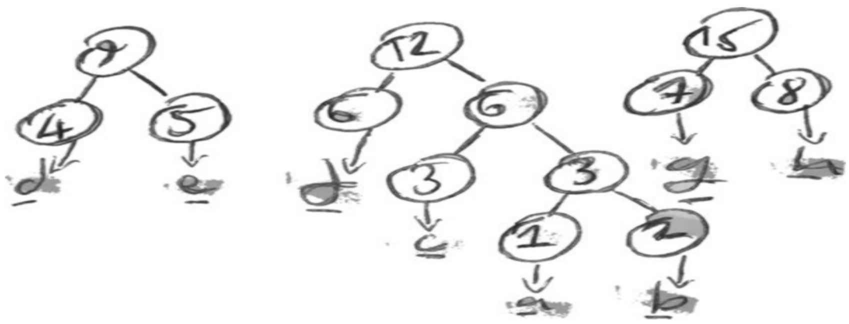
Key: (**6**=>**f**, **6**(**3**=>**c**, **3**(**1**=>**a**, **2**=>**b**)), **7**=>**g**, **8**=>**h**, **9**(**4**=>**g**, 5=>**e**)

[Figure 16]-(Huffman Example Step Four)



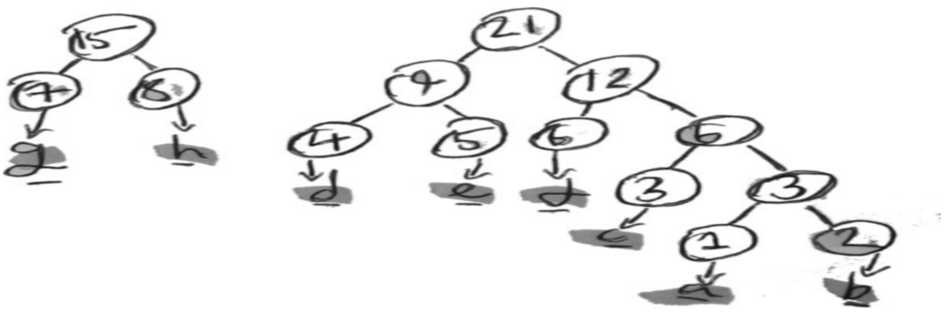
Key: (**7**=>**g**, **8**=>**h**, **9**(**4**=>**d**, **5**=>**e**), **12**(**6**=>**f**, **6**(**3**=>**c**, **3**(**1**=>**a**, **2**=>**b**)))

[Figure 17]-(Huffman Example Step Five)



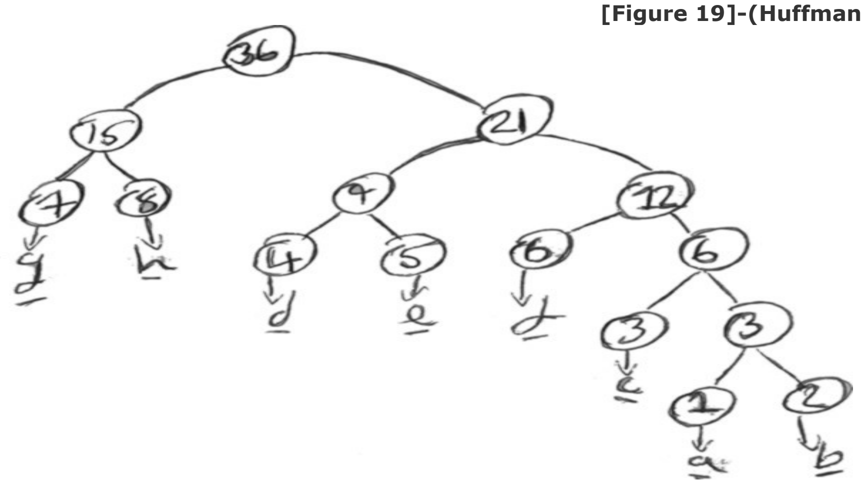
Key: (**9**(**4**=>**d**, **5**=>**e**, **12**(**6**=>**f**, **6**(**3**=>**c**, **3**(**1**=>**a**, **2**=>**b**))), **15**(**7**=>**g**, **8**=>**h**))

[Figure 18]-(Huffman Example Step Six)



Key: (**15**(**7**=>**g**, **8**=>**h**), **21**(**9**(**4**=>**d**, **5**=>**e**), **12**(**6**=>**f**, **6**(**3**=>**c**, **3**(**1**=>**a**, **2**=>**b**))))

[Figure 19]-(Huffman Example Step Seven)



Key: (**36**(**15**(**7**=>**g**, **8**=>**h**)), **21**(**9**(**4**=>**d**, **5**=>**e**), **12**(**6**=>**f**, **9**(**3**=>**c**, **3**(**1**=>**a**, **2**=>**b**))))

[Figure 20]-(Huffman Example Step Eight)
[54]-(LTurbak, Huffman Tree, 2001)

For more information regarding the Huffman Tree method, the original paper written by Huffman in 1952 can be easily found and is freely available online.

5.3.3 - Drain-And-Balance (DAB) Comparison

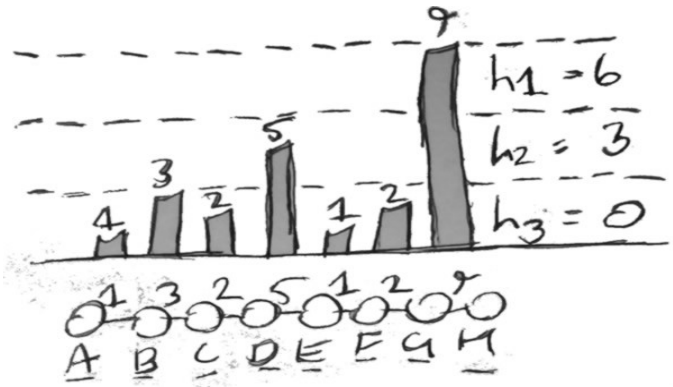
[52]-(H.T. Kung et al, 2004) describes the DAB method to a level of detail that is believed not necessary for investigation in relation to the PS, however this detail will be scanned and summarised within this chapter.

DAB (*like the Huffman method*) creates a tree structure from the bottom up, starting with the leaves and working its way "down" to the tree root.
The method effectiveness and efficiency builds upon the foundations of a "proper" selection process of the nodes used to merge in each of the steps (*and sub-steps)* by utilising event rate information. Sensors that are separated by "high-rate" subsets and one or more "event rate thresholds" known as "draining thresholds" are merged first.
The DAB process is similar to the Huffman process although differing slightly. **[52]-(H.T. Kung et al, 2004)** explains within this paper a basic DAB process:

The input of the method is a sensor graph $G=(Vg,Eg,Lg,w)$*, and its output is a hierarchy tree* $T=(Vt,Et,Lt)$*. The method is parameterised by a sequence of decreasing "draining" thresholds,* $H=\{h1, h2, h3 \dots hj\}$ *where* $hj=0$*. the* ***j-step*** *DAB tree construction for a one dimensional (1D) sensor graph proceeds as follows:*

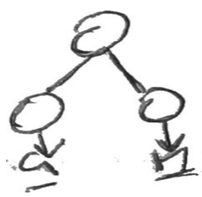
1. *Initialise* T *to be an empty graph*
2. *for each draining threshold (*$hi=H$*), in the increasing order of* **i***, perform a DAB step, consisting of the following two phases:*
   1. **Draining***: Add those nodes* Vg *into* Vt *which have at least one incident edge whose weight is greater than or equal to* hi*. This inserts a number of singleton trees into* T*. We say that two trees in* T *are adjacent if some of their leaves are adjacent in* G*.*
   2. **Balancing***: Repeatedly merge paris of adjacent trees in* T *to form clusters of sensors, in a non-decreasing manner. That is, at each merging step connect the roots of the two adjacent trees with a new intermediate node so that the merged tree will have the smallest number of sensors among all possible merges of adjacent tree pairs. The merging process terminates when there remain only non-adjacent trees in* T

Note that the last DAB step corresponds to the threshold $hk = 0$*, guaranteeing that all nodes from* G *are inserted into* T *and that the output fo the last merging step is a single tree. This follows form the assumption that* G *is connected.*



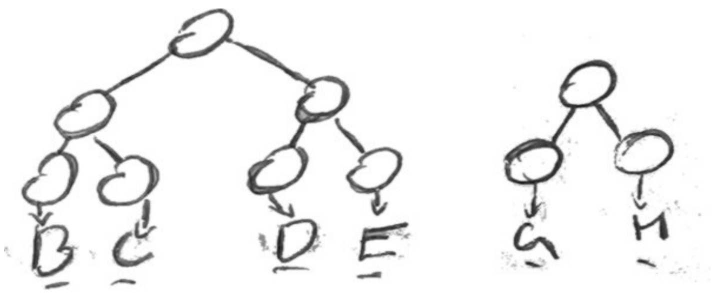
Key: (**A** (**1**) **B** (**3**) **C** (**2**) **D** (**5**) **E** (**1**) **F** (**2**) **G** (**9**) **H**)

[Figure 21]-DAB example Step One)



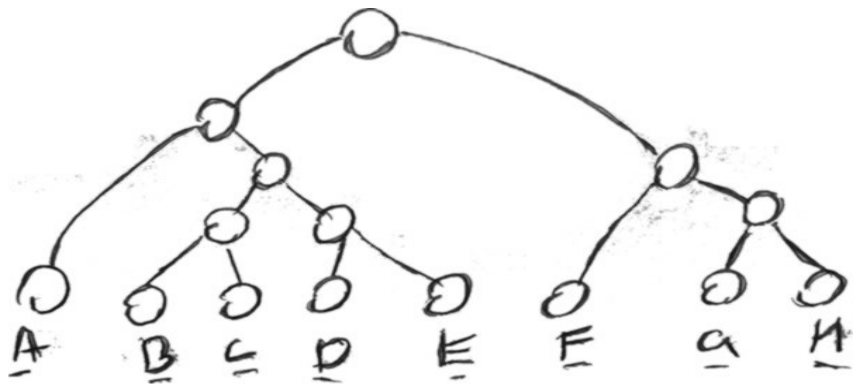
Key: (**G**, **H**)

[Figure 22]-DAB example Step Two)



Key: (**B**, **C**, **D**, **E**, **G**, **H**)

[Figure 23]-DAB example Step Three)



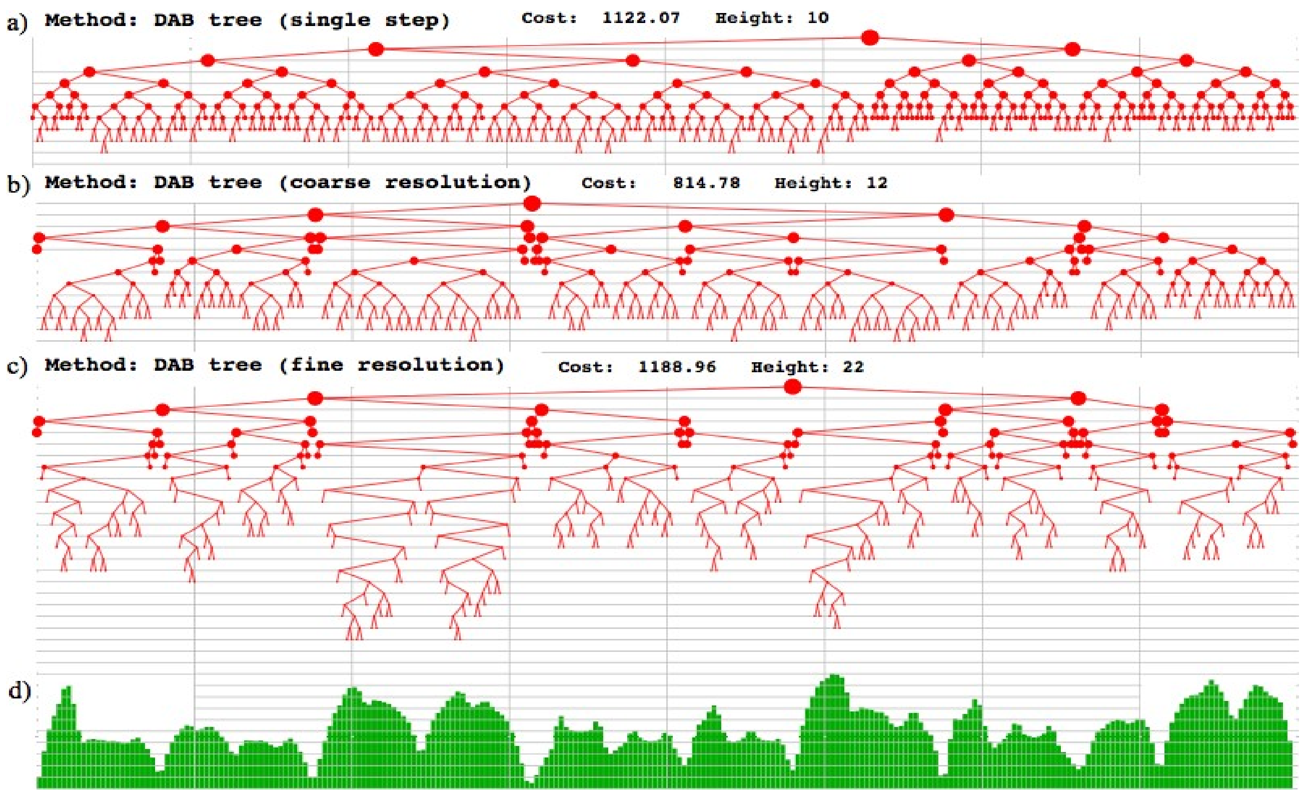
Key: (**A**, **B C**, **D**, **E**, **F G**, **H**)

[Figure 24]-DAB example Step Three)
[52]-(H.T. Kung et al, 2004)

The Diagrams show a typical DAB (three step) tree construction for a 1D sensor graph. The weights are depicted using vertical bars with the draining thresholds **H={6,3,0}** as shown in **Step 1**. **Step 2**, **3** and **4** show **T** through and at the end of the DAB stepping procedures respectively, with the full and complete hierarchical "message-pruning" tree being shown in **Step 4.**
The method processes the "high-cost" sensor nodes early within the flow thus allowing the connecting nodes to be placed near to their leaves in the hierarchy. In a similar way, the "low-cost" sensor nodes are left until the end to be processed, thus scattering the connecting nodes between large areas that have been derived within previous steps.
[52]-(H.T. Kung et al, 2004) describes that "*... each step within the DAB tree construction can be viewed as a process of draining water in a region to reveal high peaks and then grouping these revealed high peaks which are adjacent into clusters in a balanced manner*".
Taking this image on board, within **Step 1**, draining lowers the water level down to the height of **h1=6**, revealing the peak of both **G** and **H**. **Step 2** drains the water further to the height of **h2=3**, this time revealing peaks that correspond to both **D** and **E**. In the final **Step 3**, the water level lowers furthermore to a level of **h3=0**, where in this case, all the remaining peaks are revealed, thus explaining the naming convention "Drain-And-Balance" (DAB).

In comparison to the previously described Huffman Tree structure, the Huffman Process does not consist of intermediate nodes that handle message-pruning between the route from the leaf to the root node of the structure, thus achieving minimal costs for such a structure for a given set of event rates that are associated with the sensors. In contrast the DAB process assumes the message-pruning at the intermediate tree nodes. The Huffman approach merges the "low-cost" nodes first, leaving the "high-cost" nodes until later in the process resulting in them being connected via intermediate nodes close to the root node, whereas the DAB process handles the "high-cost" nodes first, allowing these "heavy" nodes to be processed and pruned at the intermediate nodes near the leaves of the structure. Furthermore, the Huffman process does not take into account or handle any sort of tree balancing functionalities unlike the DAB process, thus leaving (*sometimes)* very unbalanced results/graphs (*either right or left hand side heavy*) consisting of very long links and depth on the heavy side of the relevant graph.

Unbalanced trees are usually caused by a method containing a large number of levels (*steps*).
Using an excessive number of steps to create a tree can limit the choice of sensor nodes that can be merged during each step thus limiting balancing possibilities.
A trees structure is determined by the positions of the sensor nodes that are detected within each step. This sometimes results in the discussed unbalanced tree containing long links and a large height (*one sided heavy trees*), thus the fewer steps required within a method will in theory result in the creation of a more balanced and faster tree. If the complete tree is however created in a single step, there is no known acknowledgment of either "high or low cost" areas. This results in a tree with no linked "rate" information and therefore is more likely to be well balanced having a low height (*no gaps will exist that could cause and unbalanced result*), however, the fewer steps used result in high communication costs.
The nature of the system will depend on both the structure of the method required by a mobile objects movement frequency as a function of locations relative to the environment that it is being tracked within, thus dictating the steps needed to merge the sensor nodes and finally controlling the shape of the overall tree.



[Figure 25]-(An example DAB Tree Structure)
[52]-(H.T. Kung et al, 2004)

[Figure 25] shows an example constructed using the DAB method for a 1D sensor graph consisting of 256 nodes. "*The intermediate nodes are placed between their subtrees, in order to make it easier to inspect the associated weight.* ***d)*** *The input weights are shown at the bottom.* ***a)*** *the top tree, obtained by a single-step run ignores the weights and places some high level nodes into expensive areas of the region.* ***b)*** *the middle tree, obtained using 4 DAB steps, is best of the three.* ***c)*** *The bottom tree, obtained in 17 DAB steps, suffers large delays due to poor selection of merging choices at each step*" **[52]-(H.T. Kung et al, 2004)**.

What is considered to be a good and valuable knowledge and understanding of how mobile objects can be located with the use of hierarchical tree structures has been established and will further aid with any decisions that are to be made throughout the remaining development process. This will additionally providing a better understanding and underlaying foundations of the methods used (*by devices and technologies*) when tracking mobile objects.
Further investigation at such a low level within this area is deemed irrelevant at this stage of the project and will therefore be closed at this point. A higher level investigation process will begin looking into the different technologies and existing methods and options that can be used to obtain such locational based information, starting with Mobile Object Tracking Solutions:

5.4 - Mobile Object Tracking Solutions

There are two "main" solutions that have been identified that are believed to be relevant for potential use with the PS in terms of mobile object tracking, including the traditional **S**ensor **B**ased **S**ystems (SBS) and the more recent GPS.

As discussed and established within the LR, GPS is the favoured technology when tracking mobile objects, with many relevant literatures covering the technology as well as existing and similar systems incorporating the technology for complementary usages.
Those other solutions and technologies used within successful systems include the popular sensor based technologies and even optical tracking technologies consisting of the installation of cameras that are positioned around a racetrack enabling optical vehicle tracking. These areas will also be considered as and where relevant within this chapter.

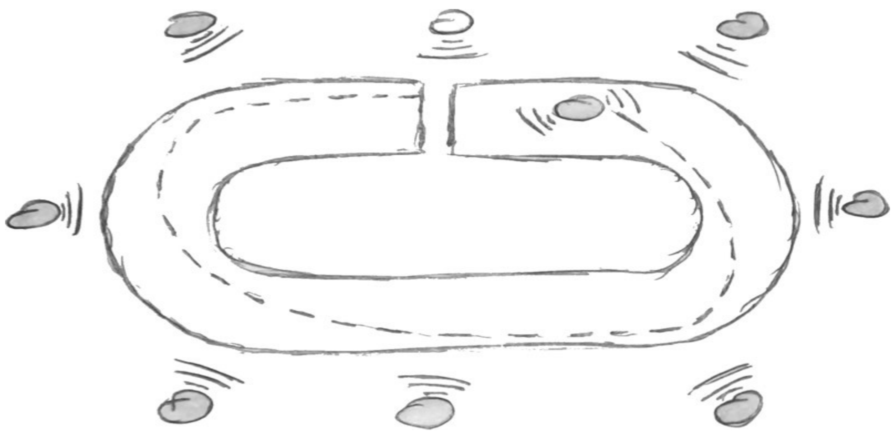
"*Auto racing has traditionally used inductive sensors similar to automated traffic sensors in the track to measure lap times and speeds by detecting when a race-car crosses the start/finish line, using high accuracy data from GPS tracking devices, however, it allows creation of virtual lines by defining positions along the track*" **[65]-(T Ford, Novatel, 2002)**.

The PS requires accurate positioning data to be collected from a mobile device/receiver that is installed on a motorcycle further transmitting this data to some sort of storage bank (*primarily a database*), where the data can accessed enabling data analysis.
The device should be small and lightweight thus not affecting the performance of the bike (*aerodynamically or mechanically*) yet maintaining accurate positioning and coordinated results. It is common (*especially in the case of a GPS based solution*) that an antenna required that is required to be positioned on the outside of the motorcycle (*somewhere on the external body*) where it is not likely to get blocked, thus allowing visibility to the sky at all times, maintaining strong satellite signals throughout a lap and enabling the calculation of object positioning.

These discussed requirements will be used (as well as the requirements documented within the PP) to compare the selected two possible solutions, starting with SBS:

5.4.1 - Sensor Based Systems

There are many existing and relevant systems using "inductive" Sensor Based Systems (SBS) that additionally do not rely on GPS whilst achieving similar results, when used for similar purposes to that of the PS, including Local Position Measurement (LPM). Before the revolution and growth of the GPS technology, inductive SBS was the main and most powerful (*accurate and reliable*) technology for use with such LPM systems.
Due to its popularity and performance benefits, it has produced much interest and demand, bringing further investment, support and and research to the technology. This will aid in producing more powerful enhancements and dimensions which will continue to push the possibilities of the technology. SBS is still popular today and commonly in use with successful, powerful and reliable systems. These offer features and possibilities to systems such as fast, accurate and real-time measurements and even 3-Dimensional (3D) capabilities.



[Figure 26]-(A simple racetrack based SBS)

[Figure 26] shows the foundations of a simple SBS relative to what could be used in respect to the PS. The system consists of an environmental setting (*racetrack*), a mobile object (*motorcycle*) and sensors strategically positioned around the racetrack communicating with a device that is installed on the mobile object throughout a lap(s) of the racetrack collecting and storing relevant statistical information.
The data collected from the mobile object can be later calculated providing detailed analytical information in relation to the mobile object and the environmental settings and sensors, including lap times and location based information, thus providing the possibility of calculating factors such as speeds, lines taken (*track curves*) and distances travelled.
This type of system is not limited to a single object and can be easily expanded to support multiple motorcycles simultaneously within a racing scenario. This will allow comparisons between the data collected from the "multiple" devices and the identification between objects (*riders*) showing such details as fastest lines, fastest machines (*within different areas of a racetrack*) and even during and final race positions.

An advanced and respected company within the electronic solutions industry has developed such a LPM based system. Abatec Electronics AG are continually investigating and researching such technologies developing products and devices "*tackling the new, adopting unconventional approaches and using its lead to maximum advantage*" **[56]-(Abatec Electronics AG, 2008)** thus creating further solutions and enhancements to technologies within relevant fields to that of the PS, opening new doors and dimensions to what can be achieved.

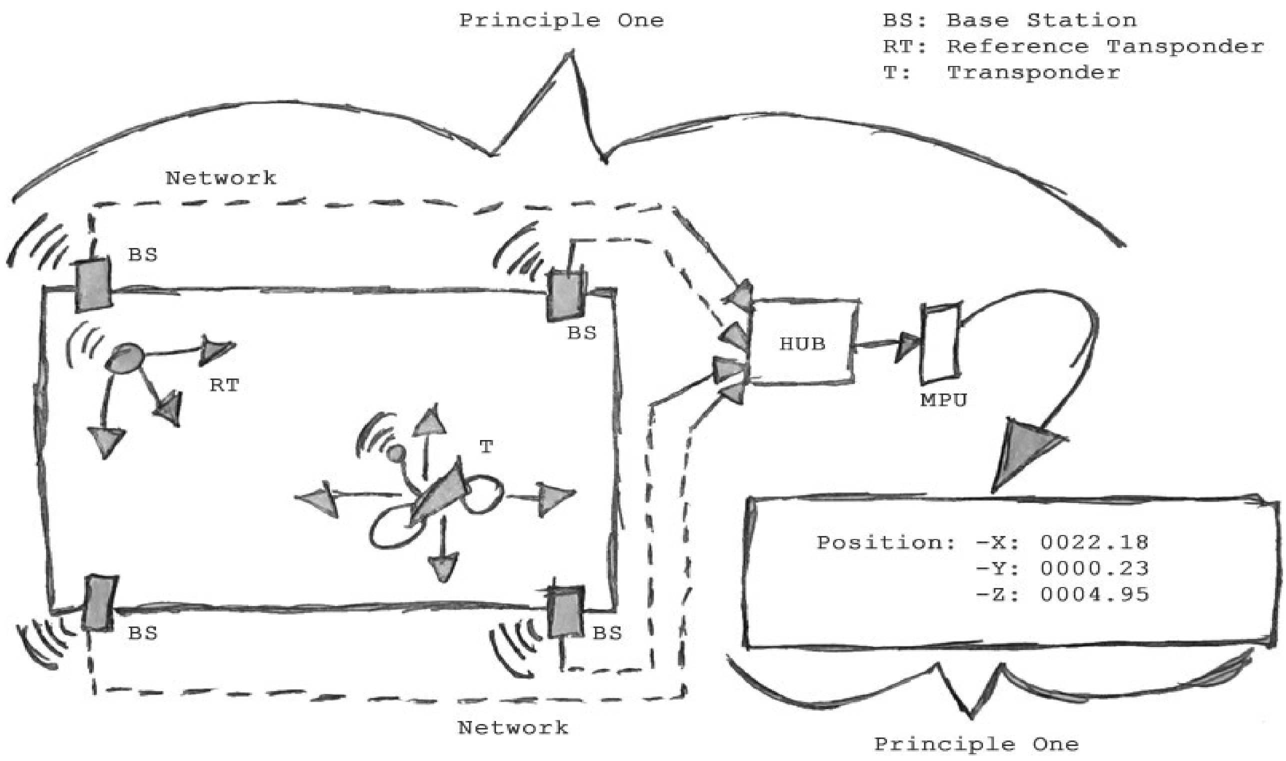
"*In its in-house R&D department, Abatec is constantly developing innovative new products. The scope of our own products ranges from modular high-end sauna control units and control units for infrared cabins to a Local Position Measurement system*" **[56]-(Abatec Electronics AG, 2008)**.

5.4.1.1 - Abatecs Local Positioning Measurement System

The Abatec LPM system is considered to be the most advanced SBS location system found during the literature search and review, and will therefore be used as an example that represents the capabilities of such systems, in terms of achievements and possibilities in relation to the PS. This will act as a foundation upon which to base further investigation.

The system offers and opens analysis solutions and dimensions to many different fields including Sports Industies and Science. The system will be investigated focussing on the needs of the PS, and how it (*or areas of the system*) "could" be used to enhance and push further development.

In comparison to other reviewed and relevant systems, the Abatec system appears very advanced and powerful offering accurate position (including 3-Dimensional) measurements. This is described as the "*only technology to measure the positions of moving persons, animals or other objects up to one thousand times per second second*" **[56]-(Abatec Electronics AG, 2008)**.
Although meeting the requirements of the PS, this system is considered too advance at this prototype stage, however the system highlights and offers many routes for future project expansions.



[Figure 27]-(Abatec System principle)

[Figure 27] shows the Abatec system functional principles. The WSN is connected to "measuring stations" that are strategically positioned around an environmental setting. The stations receive signals that are sent from the "transponders" detecting the signal receipt time which is transmitted to a master computer (*in real-time*) where the current positioning computations are calculated using the transmitted data. "*3D position data, expressed as X-, Y-, Z-Coordinates, as well as the speed vector*" **[56]-(Abatec Electronics AG, 2008)** are transmitted, additionally providing a supplementary telemetry channel allowing the transmission of other data types, which in relation to the PS could be used to send other relevant acquisition data such as RPM and throttle position information. this is a system that both provides and calculates positional measurements and additionally allows further relevant data to be transmitted simultaneously.

To use such a system demands certain hardware demands and items including a hub, base station, transponder and docking station along with various cables and mounting systems.
The hardware is collected and setup building a complete sensor based system for tracking and collecting data from a mobile object. The hardware must be setup correctly and the sensors must be placed in the best position for the transmission results, allowing the whole system to communicate well and reliably. This process can be (*sometimes*) complicated and time consuming, however, when complete can (*and should*) allow the production of good, reliable results. In regards to the PS, the system will further need to be configured/communicated with LPM software allowing the translations and analysis of the data (*this will be discussed later within this chapter*).

Abatec offer different levels of hardware depending on system requirements, offering system adaptability and flexibility. For example, for systems that do not require 3D enabled results, 1D (*less advanced and cheaper alternative*) versions of the hardware are available. they work in the same way, but collect less detailed data, thus improving performance for systems where 3D data in not necessary.

Additional basic software is offered that can be used with the system to help with the translation of the data collected, however its purpose is to act as a more general (*application-independent*) and configurable element that is not tailored to a particular system, thus it will not meet the front-end requirements of the PS.
Altogether the Abatec LPM system has a lot to offer (*a complete model*) towards a solution of a sensor based tracking system, providing most of the foundations required. These can be expanded upon developing a particular system which will meet any set requirements.

The system "*... compromises hardware and software for real-time position measurement (at a rate of* *up to one thousand times per second) on a high-frequency basis, generating both X-, Y-, Z-data and speed vectors. To ensure maximum application flexibility, various LPM models are available: wired or wireless 1D versions, as well as mobile, fixed or semi mobile 3D versions (basic infrastructure plus mobile electronic equipment)"* **[56]-(Abatec Electronics AG, 2008).**

As discussed the system is developed as an adaptable system aimed for use within many different fields (*sports, media, science and industry*) making for a very general and wide range system offering many possibilities and solutions. In a field relevant to the PS (*Motor Sports*), Abatec describe that the system can "*...allow constant analysis of vehicles and drivers. Reaction times in connection with track curve and velocity data help to identify any handling problems and show potential for improvement*". in addition further telemetric data can be transmitted via a separate interface such as the discussed RPM and throttle positions. The system is currently being used in Austria for similar (*although differing*) reasons, within driver training centers of the Austrian Automobile Association (ÖAMTC).

Such systems offer many advantageous areas that can be used with, and aid the PS. It is deemed therefore relevant to investigate different LPM systems and capabilities in more detail:

5.4.2 - Local Position Measurement (LPM)

"LPM comprises hardware and software for realtime position measurement (at a rate of up to one thousand times per second) on a high frequent basis, generating both x-, y-, z- data and speed vectors (including application-independent basic software for data visualisation). To ensure maximum application flexibility, various LPM models are available: wired 1D versions, as well as mobile, fixed or semi mobile 3D versions (basic infrastructure plus mobile electronic equipment)" **[56]-(Abatec Electronics AG, 2008)**.

The discussed system from Abatec is a typical LPM system, and has a foundation that could be adopted and used within the PS. As discussed, Abatec specialises in, and is widely recognised within the areas of LPM systems.
There are two main functional principles to such systems, the first being the acquisition of location information from mobile objects, and the second, the distribution of the acquired data.
The first principle comprises a region in which a mobile object can travel. The real time network in which an LPM system is implemented is connected to several base stations (BS), with each mobile object measured within the system being addressed individually using a transponder that responds with a pre-defined signal. The transponder installed on the mobile object(s) is the actual device from which location data is read and calculated. Multiple BS's (*measuring stations*) that are connected through a WSN or using fiber optic cable where preferred, and are positioned around and within the region receiving the signals that have been issued by the transponder(s), referencing the relevant transponder and detecting a signal receipt time, in turn communicating with a Reference Transponder (RT) that synchronises the collected data from each BS before it is finally pushed (*transmitted*) via the network in real time to a central hub that is connected to a real time calculator (MPU) where the current position of the transponder (*mobile object*) is computed.

The second principle consists of pushing the acquired data to an application where it is converted from data into understandable information, whether it be just raw data (*a low level data structure*), a specialised (*general*) tool or customer specific software. In the case of the PS, the latter is the most relevant as a specific (*tailored*) application is required to be used to analyse the acquired data. 3D positional data, expressed as X-, Y- and Z- coordinates along with the discussed speed vector are pushed to the relevant application software and are automatically displayed (*visualised*) as meaningful information (*for example as charts, graphs or data grids, again depending on the **A**pplication **S**oft**W**are (ASW)*).
An additional telemetry channel (*as explained within the example system*) is also available allowing additional data to be compressed and transmitted.

5.4.2.1 - LPM Components

To enable an LPM system to function various components are required to be in place within the first principle process working both independently and with one another (as shown in **[figure 27]** above), acquiring and processing data from a mobile object.
To get a better idea of the requirements of such a system and the hardware that is needed to achieve such results, an example LPM system setup (*based upon the standard choice of components used within an Abatec LPM system*) will be investigated. This investigation can be found in **[Appendix 2]-(Local**

Position Measurement Component Investigation).
For more detailed information and pictures of the discussed components as well as additional components and tools that are used within LPM systems, refer to **[Appendix 3]-(Local Position Measurement Component Specifications).**

5.4.2.2 - Olympic Slalom Skiing LPM System

An example of another successful LPM system that has been developed and is currently in use is the **O**lympic **S**lalom **S**kiing LPM **S**ystem (OSSS). Although different to the PS, this system shares a lot with the PS regarding not only the "possible" acquisition of locational information, but also relevant trajectory data analysis

The OSSS is again a LPM system designed and used to track slalom skiers during runs of particular ski slopes within competition (*including olympic*) based scenarios, therefore having strong accuracy and reliability requirements, and further proving the stability and potential that such LPM systems offer.

The OSSS system allows the ability to analyse and compare a skiers performance against both other skiers and other slopes. The first principle of the system produces positional data output that is pushed to the second principle for analysis (*track, drift and other analysis possibilities*). Some examples of how the acquired data is used in the case of the OSSS system, include slalom interpretations and drift angle analysis.
For example, the analysis techniques that such a system allows, taken from real slalom runs using the LPM system clearly and visually show a pattern that at the beginning of a run the rear ski neatly and very closely follows the front ski (*the skis are inline*) thus there is no "drift" behaviour. The trajectory information however, shows that by the time the skier reaches the second slalom rod there is a larger drift behaviour in place with the rear ski drifting away from the line taken by the front ski, this increase continues throughout the duration of the run as the speed of the skier increases, asking the question whether the drift is caused by carving or sliding of the ski? This question can be further analysed using the collected data as the drift behaviour can be identified in the case of when the rear ski measuring point deviates from the measuring point of the front ski, whereas carving is when the rear ski measuring point follows cleanly that of the front ski. This information can be used to measure if a skier carves cleanly (*in a professional manner*) through the turns, and looking at the results obtained from the discussed LMP system results, it is clear that the technique is a lot cleaner at lower speeds, with the quality reducing with speed and distance, thus the drifter starting to drift rather than carve.

The tracks (*lines*) taken by different skiers can also be contrasted allowing for the comparison of the different tracks to one another (*and also to what is believed to be the optimum track*) highlighting not only the different lines taken, but additionally the time and speed differences between runs and skiers.
This can be further analysed looking into the behaviour between the different poles highlighting if the skier had reached the rods either too late or too soon, comparing this driven track against an optimal track highlighting and demonstrating some differences (*drifts and different radii to name a couple*) that can be acknowledged and improved upon for the next run.

Although these elements of analysis are different to what is expected from the PS, and also the data is being obtained within a different environment (*on a ski slope opposed to a racetrack*) the fundamental mechanics and ideas can be taken and adapted towards the PS obtaining information that is relevant to the system (the discussed tracks/lines and drifting), however the object being a motorcycle rather than a skier.

The fundamental analysis elements that are used and calculated by the OSSS LPM system include:

- Speed
- Acceleration
- Comparison of skiers tracks
- Distance from ski to the gate
- Length of the trajectory
- Radii
- Time form gate to gate
- Sector times
- Total track

[56]-(Abatec Electronics AG, 2008)

[Figure 28]-(OSSS System in action)

The system and LPM data is used and available in real time with location data acquisition being performed during a skiers run, pushing the acquired data live to the second system principle where it is analysed.
Transponders are attached to the skiers shoes with the antennas being attached to the ski relevant to the shoe that the transponder is attached to. "*Each measuring point (per ski one in the front and one in the rear) will be recorded with a bit rate of 250 Hz. Determined are the X- and Y-coordinates*" **[56]-(Abatec Electronics AG, 2008)** providing real time visualisation analysis of both statistical and competitor values analysis including speeds (*current, fastest, average*), distances covered (*from start to finish gates and between stage poles*), and the radius of the skis, further providing the analysis of the lines taken, comparison of different runs and the optimum angles compared to highest speeds obtained in relation to the poles and gates (*radius vs angles vs speed*), again, all being relevant factors that can be taken, adapted and used in a similar way with the PS.

5.4.2.3 - Further Local Position Measurement System Examples

It is apparent that LPM systems are both very powerful and versatile in the way that they can be used and adapted to fit and work with almost any sort of mobile object tracking application. Such systems are very popular and used regularly within sporting applications, with current implementations being used in cycling, ice hockey and hockey, horse racing, motor (car) racing, speed skating, skiing and ski jumping, handball and football.
Football teams including *AC Milan* and *PSV Einhoven* use LPM systems during training, pushing location data into a coaching application. Each player on a football pitch wears a light weight harness (*with a similar design to a rucksack*) containing a transponder, therefore allowing location information of every player on the pitch, at any time to be collected. This data is then pushed into the coaching application where information such as speed, distances velocity distribution and tactics can be analysed further, and depending on the application, features such as fast and slow motion replays, different perspective views (*including both 2D and 3D*), games scene storage, video overlays as well as player detail overlays (*name and number*) along with various data export services are made possible from the location data acquired from the LPM system.

Less relevant uses of such systems include within theaters or on stages (*at music concerts*) where the performer(s) have a small transponder attached to them communicating and working with the other required components of an LPM system, calculating the performers positional information and sending this information to an electronically controlled spot light system. This provides accurate and reliable

alignment and control (*more accurate and reliable than human control*) of the spotlight, both tracking the performance, and further eliminating the need for a human to operate such a system.
Another example of agricultural research with regard to the location tracking of animals. LPM systems deliver data for such systems to track the movements and distances travelled by animals (*for example cows within a field*) highlighting patterns, behaviours and common occurrences, as well as standing times, resting periods, feeding periods and social behaviours of the animals.

More relevant to the PS is that of systems within the motor-sports/motor-racing environment. Such application uses require (*in most cases*) the real time data for speed, acceleration and distances from which most information needed for analysis can be calculated.
Motor sports: "*... LPM allows constant analysis of vehicles and drivers. Reaction times in connection with track curve and velocity data help to identify any handling problems and show potential for improvement. In addition telemetric data, such as heart rate, can be transmitted via an interface*" **[56]-(Abatec Electronics AG, 2008).**

No matter what the system is being used for, as long as it is set up on the foundations of a recommended LPM system using the minimum component requirements (*for the environment that it will be implemented*) the system "should" offer the following technical and performance numbers and specifications:

- Carrier frequency: 5.8 GHz (between base station and transponder)
- Almost unlimited measurable objects: maximum up to 16,000 (transponder)
- Measuring rate: up to 1,000 times/sec (time-sequential)
- High resolution measuring accuracy: up to +/- 5cm (relatively, outdoor dynamic)
- High Coverage: cascade-ability of 30 sections
- System interface: UDP-Link / Ethernet
- Transponder data channel (telemetry): 32 bit data channel (32 Bits/s / number of transponders)
- PC interface: UDP link (between Linux PC and application)
- Processor: Pentium 4 / 3.0GHz
- Memory: 1GB RAM

A concern at this stage of investigation regarding the use of an LMP system for the tracking of location information of a high speed motorcycle travelling around a racetrack, is that of distance possibility and accuracy.
It is known that if the racetrack does not cover a vast area, the use of an LPM system will be reliable and accurate as all reviewed systems performed well, however all are situated within a 1000m radius (500m x 500m). Racetracks vary in size and distance, and taking an extreme example of the Nurburgring in Nurburg, Germany covering a distance of over 22 kilometers (*14 miles*), rises the question of how such an LPM system would perform over such distances, additionally opening the question of how many BS's would be required to be installed around the track to maintain accuracy. This raises a further and final question of how much it would cost to install such a system in such an environment?

As described perviously, the other solution to acquiring locational information of a mobile object is through the use of GPS. LPM has many advantages over GPS including that it is indoor applicable. It offers higher accuracy and sample rates and is independent of satellite operators and consolidators (sustainable). Abetec also adds that LPM systems have the further advantage over GPS in the ability to operate in real time, however this argument has been contradicted within the LR as many relevant real time GPS systems have been discovered.
During the LR, GPS was favoured to be the technology of choice in relation to use with and for the PS, however further investigation into LPM SBS's has highlighted many advantages and additional features/possibilities of such systems in comparison to GPS:

5.4.3 - GPS Based Systems

Another solution to SBS is that of a Global Positioning System (GPS). A GPS would again be a type of sensor system and share a lot of the same fundamentals as the discussed SBS, although it would use different types of sensors, and it would incorporate an additional layer of technology (that technology being GPS itself).
Before the GPS technology is investigated in relation to how it could be used with regard to the PS, it is

believed important and beneficial to first of all research the basic fundamentals of the technology. GPS has been discussed to a reasonable level so far throughout the paper and its fundamentals will be summarised within this chapter as a reminder of how and for what reasons the technology works.

5.4.3.1 - What is a Global Positioning System?

GPS is (*today*) an accurate satellite based navigation and locating facility that tracks signals transmitted by satellites, performing calculations with these signals to determine a mobile objects precise location (*latitude, longitude and altitude*). A constellation of 24 satellites is consistently and continually orbiting the earth providing a constant (*always available/accessible*) free of charge worldwide coverage.
There are many different GPS devices and receivers available (*that will be further discussed within this chapter*) for use within a GPS based system, all being of a "passive " nature, thus allowing GPS systems to serve an unlimited number of users at any one time. GPS receivers have developed into very advanced and sophisticated pieces of hardware offering accurate location calculations several times per second allowing them to precisely measure to within (*commonly*) meters, centimeters, and even millimeters (*in some cases*) depending on the type of receivers being used and the implementation of the system.

The satellites send both a carrier signal and pseudo-random code signals which are timed by an automatic clock positioned within the satellite itself, with the receiver(s) additionally creating a matching code that is timed by its own (*built-in*) synchronised clock. The distance regarding how far away a satellite is situated is calculated by the time it takes for the transmitted signals (*from the satellite*) to reach the relevant receivers. Normally the pseudo-random code signal is used to perform this calculation, however, the carrier signal can sometimes be more accurate and in some cases is therefore preferred to be used when performing such a calculation. In some additional cases (where accuracy is a strong factor), both signals are used performing the same calculation and thus cross referenced and checked for reliability and accuracy of monitoring.
To enable the calculation of such described location and positional information, the receivers used must have a minimum of four satellites visible at any time (*the more satellites visible to a GPS receiver , the more accurate the results will be*).
Three of the selected and visible satellites are used to "triangulate" an objects position, and the fourth is used to improve the accuracy of the calculated position by further comparing the time offset between the discussed onboard receiver (*synchronised*) clock and the satellites automatic clock.

As discussed, there are many different types of GPS receivers available which "*... vary in the precision of their data and the ability to overcome adverse signal conditions. Some can process their readings instantaneously, while others merely collect data for post-processing. A good quality GPS receiver delivers data that is robust, reliable and repeatable*" **[57]-(S Kennedy et al, 2008)**.

GPS is a popular and fast growing and sophisticated technology that is currently being used in many different industries, under many different circumstances. It is used for many different reasons aiding, producing and opening up great possibilities, options and solutions to many different fields, along with the further developments within **D**ifferential **GPS** (DGPS) further enabling grater reliability and even more widespread use.

[58]-(NovAtel Inc, 2009)

5.4.3.2 - Applying GPS to Tracking Systems

As discovered many systems use conventional inductive SBS to track and measure a mobile object within and around a particular environment/field such as a racetrack. Instead using the high accurate data acquired from advance GPS tracking devices could be a possible alternative to an inductive SBS.
Comparison of lap times using GPS tracking devices "*... and those measured with inductive sensors agrees to within 0.005 seconds ...*" however the use of such devices proposes various complexities and challenges to such systems as accurate object "*... positions need to be obtained, calculated and transmitted under conditions in which GPS satellite signals are frequently blocked or reflected*" **[65]-(T Ford, Novatel Inc, 2002)**.

An overview of the GPS technology has been established together with what has been learned during both the literature search and review. Allowing a high level of understanding of the technology. Furthermore areas that would be required within such a GPS tracking system have been identified and touched upon, along with areas that require further investigation. These areas will be picked up and investigated further (*in relation to the requirements of the PS)* within this chapter.

As discovered within the LR, to enable accurate positioning information, a GPS devices requires at least four satellites visible (*in its line of sight*) at ay one time, and within many outdoor environments especially that of a racetrack, there can often be many problematic areas that can block such signals, including Grand Stands (*stadiums)* or other large objects that may surround a racetrack, as well as in some cases (*although to a much less extent with newer GPS technologies and devices*) extreme weather conditions.
There are no apparent or easy solutions to resolving these areas apart from the extreme (*and not always possible*) process of eliminating those interferences however, a way around the problem is the use of advance telemetry systems.
Sophisticated telemetry systems are used to transfer positional and other GPS tracking information between a mobile object(s) and a central processor/computer. "*DGPS, pseudo-range and carrier phase tracking techniques...*" generate GPS object tracking coordinates that are "*... accurate to 50 centimeters...*" with the telemetry conveying differential messages from GPS base stations to the mobile object(s) at 0.5 hertz **[65]-(T Ford, Novatel Inc, 2002)**.

It has been apparent that for usage and within an environment where signals can be obstructed or reflected GPS alone will not be substantial enough to fulfill the PS requirements. The discussed sophisticated telemetric possibilities would help performance and reliability of such a system when obstructions come in to play however, at a minimum and on top of GPS, DGPS should be implemented into the system design with the addition of a GPS reference station to the architecture. This can then provide a solution for obtaining blocked satellite signals and more accurate and reliable location coordinates by correcting measurement errors taken form a moving object. This will offer a higher level of precision to the taken GPS measurements. This will be looked at again later within this chapter.

5.4.3.3 - RaceFX's GPS FX Tracking System

A respected source and system discovered during the LR is that of the "Real Time GPS FX" system providing on-screen positioning of race-cars from the company "Sports Vision - RaceFX".
"Ken Milnes", one of the authors and developers of the system (*and an established contact within this project*) produced a paper about the system which was discussed in reasonable depth within the LR.

The RaceFX system is considered to be the most advanced GPS based location system found during the literature search and reviewx and will therefore, like the Abatec LPM based system in relation to inductive SBS, be used as an example representing such GPS based systems and what is capable in terms of achievement from such systems and technologies in relation to the PS. This will then act as a foundation upon which to base further investigation and can be further used as an infrastructure from which to build comparisons.

The system is very different to that of the PS but has various and varying similarities in terms of the location calculations required to that of high speed moving objects. Furthermore, the system has demonstrated both its own and the GPS technology to be very capable within such systems since it has been extensively used and tested under commercial operational conditions both successfully and reliably tracking race-cars throughout high league racing events including both the American NASCAR and IndyCar racing series.

"*Special effects or FX in video jargon, have long served as a mainstay of television broadcasting. Now a New York company specialising in enhancements for televised sports has incorporated GPS into its tool. The result? A system that tracks and displays on screen the real-time location of all cars during a racing event*" **[59]-(Ken Milnes et al, 2002)**.

The core of the RaceFX tracking system consists of a set of advanced GPS receivers that can operate in both pseudo-range or differential mode with the addition of the incorporation and use of a digital track model (*that can be used in both modes*) which is applied to a vehicles current location coordinates.
The solution of the RaceFX system is tightly tailored to specific requirements set by the RaceFX system, however throughout the paper the author explains that the system foundations are very adaptable and flexible and can be therefore applied to other and differing navigational and tracking solutions or systems.

The purpose of the RaceFX system is to allow users (t*elevision viewers*) the possibility to watch a race with the addition of on-screen aids, such as car hilighting, driver names and positions (*superimposed above the race-cars*), track location (*a map of a track with a colour coded car key*) and other statistical

information of a cars performance, all being superimposed on a television screen.
Similar systems exist in other sporting fields including, for example, a puck tracking system within Ice Hockey that highlights the puck for television viewers during an Ice Hockey game. There also exists a system within American Football hilighting each ten yard marker on a football pitch during a game. The RaceFX system is similar to these systems, although to a more extreme extent, hilighting fast moving race-cars (*in excess of 200mph*) around a racetrack.
The complete system possibilities (*including that of a video overlaying system*) is, although very interesting, not "completely" relevant to the PS requirements, however, the systems underlying technologies are "very" relevant and could dramatically help and improve the development of the PS.

Each car is fitted with an advanced GPS receiver and sensors, allowing the system to track in real-time each individual car position with accuracy to within millimeters of its actual position (*even at such high speeds*) using advanced GPS and telematics technologies, Furthermore the system uses television cameras to superimpose (*add onscreen*) highlights of each (selected) car, all in real time.
Although the latter step is not relevant to the PS, it both shows and proves what is possible and what can be achieved for future project expansions.

As with most GPS based systems, the satellites are a fundamental source used to locate a race-cars position as the car travels around a racetrack. The satellites work alongside an advanced "earth" based **S**atellite **N**avigation **S**ystem (SNS) allowing the race-cars to be accurately tracked to within 20 millimeters of their actual position(s). This amount of accuracy is very impressive and more than adequate in meeting the PS requirements in which to within a centimeter of accuracy would (it *is believed to*) be satisfactory.
It is specifically described above (*and although is obvious within this situation*) that the system used is an "earth" based navigation system. It is thought relevant to highlight at this point that such systems can be either "earth" or "sky" based, with the "sky" based systems generally used to track flying objects.

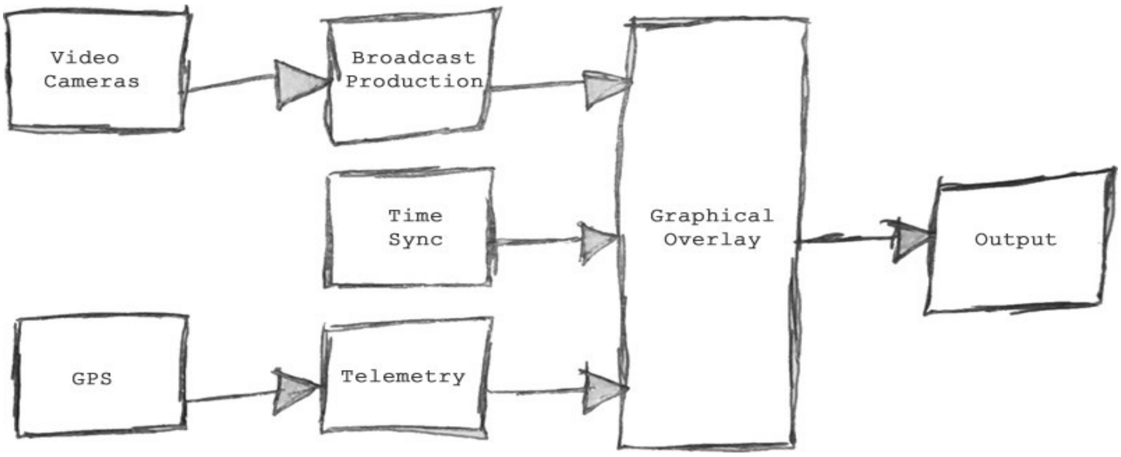
The cars themselves contain advanced sensors and a GPS receiver that communicate with the satellites, allowing the cars exact location to be tracked. As with the PS, these devices are required to be small, compact, lightweight and aerodynamic not interfering with performance in any way.
The sensors transmit to a WSN statistical information regarding a race-car during a race. Such information includes speed, acceleration, braking, RPM and current gear information to name but a few, again, all being relevant to the information required within the PS.
The RaceFX system, however, continues a step further with the addition of a camera installed in each car recording and transmitting video to the system. Digital map models are created of each racetrack within which the system will be used and this model is further merged with the information recorded/collected from the onboard camera, allowing the location of the car relevant to the track to be displayed. All in all this makes for a very impressive and effective final result.

The paper reviewed in the LR regarding this system further describes the challenges and difficulties of such a system within a racing environment and the benefits that the use of GPS based system has over an inductive SBS. These said elements will be investigated further within this chapter.
As already described, both GPS and inductive SBS agree to results within 0.005 seconds, however, the advantage of using GPS within the RaceFX system is that it provides the ability to create virtual lines by defining positions along and around a racetrack, which are used to show "*... instantaneous speeds at arbitrary points along the track, such as turns and straightway*" **[59]-(Ken Milnes et al, 2002)** thus allowing the system to precisely locate a race-car at any time during a race, and in the case of this system enabling the placements of on-screen graphical enhancements. Additionally and more relevant to the PS, it allows statistical information to be calculated such as the (*racing*) lines taken/used and the path/line taken during the fastest lap-time.
As with most GPS based systems, the RaceFX system suffers from the same challenging areas including that of blocked or reflected satellites (*know as multi-path*), and the need for acquisition of accurate and real-time vehicle positions, which, in this case are exaggerated due to the speed and environment in which the race-cars are travelling.

5.4.3.4 - The RaceFX System Concept

The RaceFX system is constructed from four sub-systems (*three of which being relevant to the PS*) including GPS, telemetry, time synchronisation and video overlaying with the latter (*video overlaying*) being considered irrelevant for further investigation in consideration to the aims and objectives of the PS.
Similarly to the previously described GPS based tracking system, each competing race-car has a GPS

receiver installed along with a 900MHz transceiver, with the transceiver being mated with other 900MHz transceivers that are positioned around and within the racetrack, sending data over DSL modems to a CBS containing the communications controller, time synchronisation system and video overlaying system.



[Figure 29]-(RaceFX System Design)

The system uses the DGPS technology along with an advanced telemetry system that is used to transfer information from race-cars (both locational and statistical) to a CBS at a transfer rate of approximately five times per second. The DGPS system along with carrier phase techniques allow location coordinates to be acquired accurate to within 50 centimeters. This together with the telemetry system conveying the GPS differential measurements between the CBS and the units installed within the race-cars (*rover units*) at 0.5 hertz and further from the rover unit to the video overlay system at 5 hertz. The information is then used to calculate additional and useful metrics (*including speed and acceleration metrics*). To improve performance, calculations are performed using the velocity vector between the GPS positions since they are required to be kept as small as possible as they need to be compressed into a data packet for transmission.
The obtained positioning information acquired from the GPS system is interpolated to correspond with measurements acquired by the on board cameras. Each car has an on board camera that measures factors such as the cars pan, tilt and zoom which is further combined "frame-to-frame" with the acquired location information relevant to a particular frame, allowing an accurate video overlay to be defined.
All race-cars transmit data to the overlay system simultaneously, where it is reformatted into individual and relevant video frames and used for broadcast, "*... while the time synchronisation system tags each videos frame with a GPS time stamp*" **[59]-(Ken Milnes et al, 2002).**
For an effective and worthwhile result, the relative timing between both the video and GPS measurements must be accurate to within one millisecond keeping time induced errors to below 10 centimeters. The accuracy of these measurements, especially within such a real-time environment with the mobile objects (*race-cars*) traveling at speeds in excess of 200mph is essential. "*The GPS based system maintains the timing to about 10 microseconds, or 100 times better than the minimum requirement*" **[59]-(Ken Milnes et al, 2002).**
A key and interesting element which the system produces, that could also (*possibly*) be added to the PS, is the digitalised racetrack model. As discussed, for each racetrack that the system is used within, a digital model is created that is used for both support of the real time video display and also for when the satellite signals are processed. This improves the real time positioning accuracy of the race-cars.

It is described that tracking the positions of the race cars to such accuracy and within such conditions will and does stress a conventional GPS system. The greatest challenge is the maintenance of continuous and accurate GPS position fixes from a racetrack to the minimum number of satellites required for accurate positioning determination.
As previously discussed, four satellites are the minimum required for the calculation of accurate 3 dimensional positioning, however, within such a restricted environment (*a racetrack*) it is not always possible to obtain all four signals. In such cases, when four satellites are visible, there is still a chance that they would not maintain the required accuracy within such an environment (*the necessary geometry is not available*), thus measurement accuracy can be reduced and hence it does not meet the systems standards.
The satellite geometry can translate into what is known as GPS **D**ilution **O**f **P**osition (DOP) that "*... acts as a multiplier of the calmative error generated by other factors, such as receiver electronics,*

ionospheric effects and multi-path" **[59]-(Ken Milnes et al, 2002).**
Objects and buildings including grandstands, fences and bridges can block race-cars rover units from the view of the GPS satellites. The RaceFX system usually uses a GPS reference system which is installed somewhere within or around the racetrack where a clear line of sight to all satellites is available (*for example on a building roof clear of obstructions*) and is used to generate and transmit any differential corrections, however, as the race-cars travel around a racetrack the amount of visibly available satellites is reduced in some places to less than three, thus producing a low DOP, thus reducing the accuracy of positional measurements.
Within NASCAR racing (*the main racing source that the RaceFX system is used with*) lap times can be as little as 20 seconds with different satellites continually coming in and out of the view of the race-car rovers during each of these short laps, again lowering both the satellite visibility and hence impairing measurement accuracy.
The RaceFX development team have come up with many ideas and solutions aimed at reducing or preventing these problematic areas, one of which being the discussed track model (*discussed further within this chapter*). Another technique used and developed is the generation of a special message that contains GPS satellite ephemerides that are generated at the GPS CBS and transmitted to the race-car rover units via a differential data link. This generated message is necessary as if the message was being downloaded directly from the satellites themselves. The download time produced will be too long and therefore it cannot be guaranteed that continuous coverage between the satellites and the rover units will be established for the amount of time that is required.
Each participating race-car is additionally fitted with a DAPS containing a "*12 channel, dual frequency receiver, a 900 Mhz spread spectrum radio modem, a computer 486 computer, and some custom interface electronics*" **[50]-(Ken Milnes et al, 2002).**
As discussed, within such a system, the devices installed on the cars are required to be small, compact and lightweight. These devices are also required to be robust as they will be "battered" around during races experiencing various pressures, forces, vibrations, bumps (*from the track*) and even crashes during the course of a race, and again all these being important factors that should be taken into account when choosing such a device to use with the PS. A race-car can experience "*... between 6 and 18 G's of vibration on the chassis. At these vibration levels, the crystal oscillator in a GPS receiver will induce enough phase noise in the oscillator signal to create satellite tracking failures. To avoid this the oscillator must be mechanically isolated from the vibrating car in order to reduce the vibration to below 3 G's*" **[59]-(Ken Milnes et al, 2002).** Installing such devices and additionally positing the GPS antenna without affecting the aerodynamics of a vehicle traveling at such speeds obviously requires in depth thought and testing that are all factors that would be required for the PS.

The track models created consist of a "*... virtual set of planar surfaces that approximate the contiguous surface on which the race takes place*". Typically a race-car can be "*... accurately represented by a series of fewer than 3,000 contiguous triangles defining the planer surfaces on the track. (Each triangle is typically between 5 and 10 meters on a side)*" **[59]-(Ken Milnes et al, 2002).** As soon as the GPS "reference-to-track" model reconciliation has completed, each of the triangulated points it will have an accuracy of 10 centimeters (*1 stigma*) relative to the GPS CBS.
Aerial photos along with advanced programatic techniques are performed at each track that the system will use, obtaining a digital representation of the track surface. The photographs are taken at roughly 300 meters above the track surface using a helicopter and advance photographic equipment resulting in high resolution overlapping aerial photographs, which are processed using analytical software to create a digital terrain model of the track surface. This can be used as an accurate coordinate system, using the GPS survey receivers to obtain both ground and camera control for the photography process.
The track models within the system are therefore very detailed producing very realistic results including details such as track markings (*paint, wear and tyre marks*), fences, walls, buildings and even pavements that may surround and are relative to each racetrack. The actual accuracy of the model compared to the relative racetrack itself would be accurate to within 10 centimeters.

For correct and accurate use of the GPS modeling software, it is required that the GPS reference station is so located that it removes any offsets (*applying corrections*) that may exist between the track model and the GPS reference frame. The GPS CBS coordinates are determined from a timed (*average*) set of single point positions.
With the GPS CBS installed, GPS positional coordinates for any features captured can be represented using aerial photography from the track model and are measured using "*... real-time kinematic, carrier phase differential techniques*" **[59]-(Ken Milnes et al, 2002).** Position offsets (*longitude, latitude and altitude*) between the features within both the GPS CBS frame and track model frame are calculated for a minimum of at least three locations until a consistent offset is maintained, and a further differential reference is generated by the GPS CBS which is moved until the track model and the differential GPS position corrections are in agreement, either equal to, or when not possible within 10 centimeters.

The track model is not necessarily required in relation to the PS, however, it is an interesting method that can be added to the system during any future project progression. This will improve both analysis and effectiveness showing a visual model and overlay of the environment that a motorcycle travels within, as well as the improvement in positional accuracy that such a model adds to a system.
For more information regarding the generation of such a model refer to **[Appendix 4]-(RaceFX System Track Model Generation)**.

An advanced telemetry system is used within the RaceFX system consisting of a "*commercial direct sequence, 900 Mhx radio*" with time decision multiplexing, allowing data from race-cars to be transmitted in an error free, timely, accurate and predictable method.
The transceivers used operate in the **I**ndustrial, **S**cientific and **M**edical (ISM) radio frequency band. Due to the discussed obstructions within such an environment, at least 3 or 4 telemetry base stations are installed at different locations within and around a racetrack providing continuous coverage between both the race-cars and the telemetry stations themselves.
A telemetry base station is selected to be the "master" station and is used to synchronise data as it is acquired, with transmissions also being further sequenced by a time slotting system.
Control and GPS differential correctional information is broadcasted by the telemetry stations to the race-car rovers within a specifically assigned time slot. Each time slot allocated to the rover units has a 10 millisecond slot in which the previous 5 positions are transmitted, along with additional and relevant statistical data back to the telemetry base stations. "*By repeating the previous five positions every 0.5 seconds, a base station sends each data sample twice. With four telemetry stations deployed, a particular data sample may be received up to eight times (Approximately 120 bytes of data are available in each time slot)*" **[59]-(Ken Milnes et al, 2002)**. The data is received by the communications control where the redundant data packets are removed and further forwarded to the video overlaying system.

Due to the speed that the race-cars travel (*200mph, translating to roughly 3 meters per video frame*), the time between both the video time and the GPS measurement time must be closely synchronised to maintain accurate results. The positional data can be further and accurately interpolated to match the correct time stamp that has been assigned to each individual video frame. This time stamp is not in sync with the GPS time and thus an additional step is required relating both times (*video and GPS*) together, resulting in time synchronized and accurate data.
As described, the use of video overlaying is not a requirement with the PS and will therefore not be investigated any further in relation to time synchronisation.

[59]-(Ken Milnes et al, 2002)

Although different to the PS, the RaceFX system shares similar concepts and foundations relevant to that required by the PS. The system demonstrates how such a tracking system can be successfully used within a similar environment, and tracking objects travelling at both similar speeds and orientations to that of the PS. This additionally highlights not only areas that are believed to be problematic, but also areas that have been proven to be problematic explaining possible solutions and methods that can be used to work around or even solve these areas.

Further investigation of the RaceFX system has proved vary valuable in relation to the progression of the development of the PS, as further knowledge and understanding of the possibilities and limitations of using GPS in such a way have been established to a high level.

5.4.4 - Location Based System Comparison and Selection

Both inductive SBS and GPS systems have been investigated and discussed in much detail within this chapter, explaining each systems possibilities, advantageous and disadvantageous areas with the addition of using "real life" examples including those of existing and successful systems that are in use today.
Throughout the project, the acquisition of location based data from a high speed moving object (*motorcycle*) has, and is believed to be one of the most challenging areas/requirements of the project, with both of the selected systems (*SBS and GPS*) potentially being possible solutions.
The systems offer a lot in terms of solutions, possibilities and expansions to the PS, as well as fulfilling to a satisfactory level all of the initially set project requirements.

This section of the chapter will compare the two systems in terms of both relevance and practicality for use and implementation with the PS, with the stronger (*and most thought relevant*) being selected and used for and within the project development.

The fundamental aim, purpose and requirement of such a LBS relative to the PS is to obtain accurate (*to within 10 centimeters*) locational information (*coordinates*) from a mobile object (*motorcycle*) as it travels within and around a set boundary (*racetrack*) at high speeds (*up to and over 180mph*).
During the LR and from investigation within this chapter, it was apparent that both methods could not only meet the set requirements, but also expand upon them. The set requirements are however still believed to be both challenging and demanding, as the acquisition of accurate location data from a mobile object travelling at such high speeds (*although possible*) is not an easy task. It is however believed that prior to the completion of the LR and further research and investigation, the additional potential and possibilities of such systems were not fully understood.
Although, with more (*unknown*) potential available, the initial project proposal is still believed to be that of an advanced system facing challenging and problematic areas that will not be easy to overcome. These additional options merely open more doors for development allowing for further project expansion in and around the differing areas, with the same main challenges still existing and needing to be resolved.

The fundamental aims in relation to that of such a system (*relevant to the PS*) have been taken and relevantly expanded allowing for the comparison of the two discussed systems:

	SBS	GPS
Costs:	*	* * *
Performance:	* * * * *	* * * * *
Accuracy:	* * * * *	* * * * *
Real-Time:	* * * * *	* * * * *
Multiple Objects:	* * * * *	* * * * *
Telemetry Channel:	* * * * *	* * * * *
Measuring Area:	* *	* * * *
Outdoor Usage:	* * *	* * * *
Overall Average:	*** * * ***	*** * * * ***

The comparison table above in relation to the PS clearly shows that a GPS based system is a lot stronger, having a higher amount of comparison stars over the SBS system. This result, however does not completely eliminate the SBS system. The table will be confirmed by further discussion and justification, and where relevant, arguments will be put forward that contradict the table results.

With regard to performance and accuracy, both system are very similar, the GPS system having the edge but this being only when advanced and sophisticated additions have been added to the systems foundation, in terms of DGPS, telemetry and various time synchronisation techniques. The SBS system is however on the other hand also very strong within this area and both systems are more than capable for use with the PS, meeting and overlapping the set system requirements.
Although not a set requirement, both systems support real-time data transfer thus allowing applications to be used in real time, and in the case of the PS, the data being acquired from a motorcycle and transmitted to the front end application for analysis as the motorcycle is travelling around the racetrack. This feature is not intended to be implemented at this prototype stage of development, however, there is a strong argument for further project expansion and advertisement.
The additional telemetry channel is again an additional requirement to the PS. The telemetry channel was discovered during the LR and further researched within this chapter and it is believed to be very beneficial to the PS allowing an additional channel (*within the same transmission setup*) transmitting additionally acquired data such as RMP, throttle position and any other relevant engine management system related data that is (*or could be*) required. Data can be packeted and transmitted along this channel to the same front end application where it can be synchronised and analysed with the acquired location data. Both systems (*GPS and SBS*) support this additional telemetry channel.
Another found area for project expansion is that of the ability to track multiple objects. The

requirements set for the PS is the ability to track one motorcycle, however both systems offer that multiple vehicles can be tracked just as accurately as if one vehicle was to be tracked (*with no additional cost or system limitation*), thus offering great advantages to the PS in that multiple motorcycles can be tracked at the same time, with data being acquired from each simultaneously and transmitted back to the same analytical tool for either single analysis and/or comparisons between each machine.
GPS is designed to operate in an outdoor environment (*in fact most standard GPS systems cannot operate indoors*), whereas SBS systems were originally designed to operate within an indoor environment with later and more advanced SBS being used outdoors, although it has been proved (*and as would be expected*) that moving a SBS to an outdoor environment decreases the accuracy of the system in comparison to and indoor environment. The reviewed SBS applications have been tested outdoors and show reliable and accurate results, however, in regards to both the PS and personal opinions, a GPS based system is preferred for use within an outdoor environment as they are tailored (from the begging) specifically towards outdoor operation, withstanding extreme weather conditions to a higher standard than a SBS could (*although would still be affected to a certain degree*).
Both systems are not "cheap" to install, although the GPS system has much lower setup costs than the SBS system. Each system requires advanced and sophisticated devices which are costly. They must also be lightweight and compact, adding further costs to the device retail prices. In terms of fundamental setup costs, the GPS system is reasonable, being a free service requiring only a direct line of sight to the minimum required GPS satellites which cover a wide area. Limitations however do exist in terms of obstacles that can block or reflect signals, but this can be resolved by using the described telematic and DGPS techniques making GPS in comparison to SBS very appealing. SBS on the other hand can be very costly to setup, depending on the area (*size and technicality*) that a mobile object is to be tracked. The larger and more technical a racetrack is, more sensors are required to maintain the accuracy level of the system. In terms of common motorcycle racetracks, they all cover a reasonably large area, with technical areas (*for example corners and chicanes*) that would require a large number of sensors to be installed within and around the racetrack.

Overall, both systems are very strong contenders and either could be used. Between the two systems, GPS is the strongest from both the comparison table, which will be confirmed by further discussion. For this reason and the additional setup costs, GPS will be the chosen technology used to track a motorcycle around a racetrack. If the system were to be implemented within for example a "mini-moto" or "go-kart" situation, where the racetracks are of a lot smaller scale, then it is believed that the use of an SBS would be a better (*more accurate, reelable and cheaper*) solution, however in the case of the PS, GPS is believed the best technology for use.

As discussed, the GPS technology is not as reliable or accurate enough for use without additional supporting technologies including DGPS and an advanced telemetry system.
GPS systems have been investigated to a sufficient level and much knowledge and understanding has been gained. Although both DGPS and Telematics have been touched upon and are understood, it is believed that as they will be built into the PS and will play an important role within the system. It is relevant to investigate and discuss these areas further allowing for a better understanding regarding their foundations and purpose within such a system:

5.4.5 - Differential GPS (DGPS)

It is apparent that for usage and within an environment where signals can be obstructed or reflected, GPS alone will not be substantial enough to fulfill the PS requirements. The discussed sophisticated telemetric possibilities would help performance and reliability of such a system when and if such obstructions come in to play, however at a minimum, on top of GPS, DGPS should be implemented into the system design with the addition of a GPS reference station to the GPS architecture. This will provide a solution to blocked satellite signals and further provide more accurate and reliable location coordinates, by correcting measurement errors taken from a moving mobile object thus offering a higher level of precision to the GPS measurements taken.

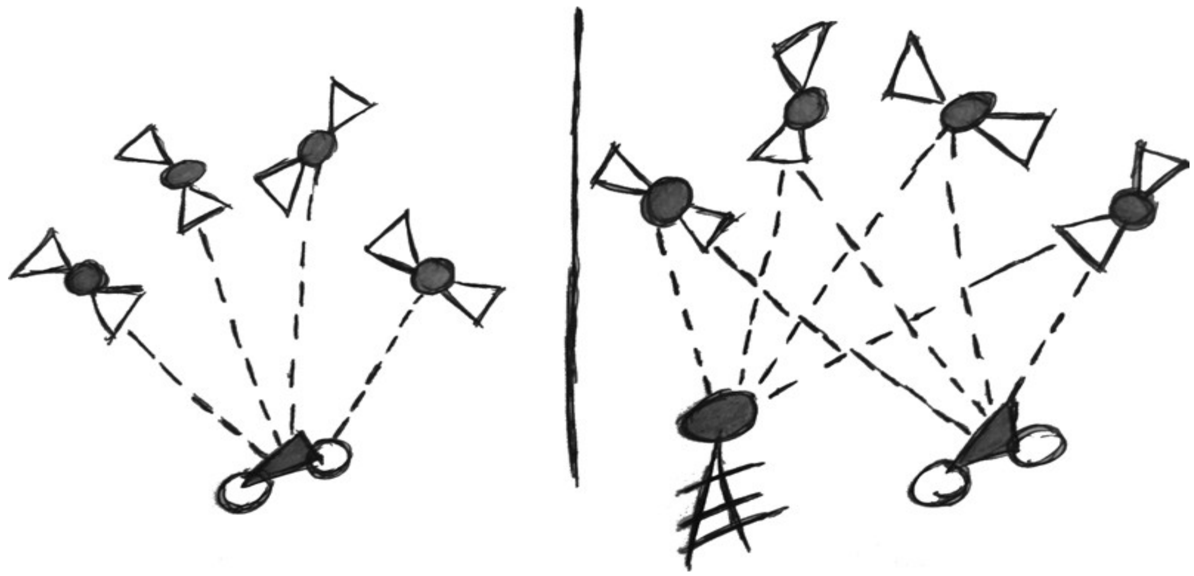
Errors exist and can cause problems within GPS systems, although most errors are in many cases unexpected errors and thus cannot be easily prevented since it is not always known that these errors will occur. Such errors include satellite, multi-path and atmosphere related errors. However, previously existed an intentional error within GPS measurement known as "**S**elective **A**vailability" (SA) which is described by many sources as one of the worst types of errors existing within the technology, and is the main reason for the invention of DGPS.
SA was a deliberate error that was (*and sometimes still is*) inflicted by the US DOD. The error

compromises of noise being injected into GPS signals corrupting accuracy and in some cases introducing errors of up to 100 meters, which is not acceptable within almost all tracking systems. The reason for this noise injection was for war situations to prevent enemy targets locating ally positions during a contact (*battle*). SA still exists today, but is tuned off most of the time, and will only be turned on again within extreme circumstances.
As discussed, DGPS was developed to solve the errors created by SA which it both did and does very well, dramatically correcting and improving the errors. When SA was introduced and initially used it was constantly turned on without the ability to be turned off, thus requiring a solution (DGPS) to be invented.
Even when SA is turned off, signal errors can still exist due to extreme weather conditions and obstacles (*as discussed and identified as problematic areas within the PS*) as well as the massive distances that exist between the earths surface and the satellites that are being referenced (*in space*), therefore DGPS can and is be taken advantage for such cases.

A precise summary of DGPS was found again during the literature search from the author **[38]-(J Zogg, 2002)** in a paper describing GPS:
"*A horizontal accuracy of approx. 20 m is not sufficient for every situation. In order to determine the movement of concrete dams down to the nearest millimeter, for example, a greater degree of accuracy is required. In principle, a reference receiver is always used in addition to the user receiver. This is located at an accurately measured reference point (i.e. the coordinates are knows). By continually comparing the user receiver with the reference receiver, many errors (even SA ones, if switched on) can be eliminated. This is because a difference in measurements arises, which is known as DGPS*".

A standard (*and common*) GPS system is normally accurate to approximately 10m, which is acceptable for some systems, however is not acceptable for others (i*ncluding the PS*) requiring accuracy to with 1cm. Due to this demand for accuracy, such system architectures are expanded to include DGPS to the structure.
The underlying basis of DGPS is that a second reference station (*receiver*) is required that makes up the "difference".
Within a DGPS architecture, the initial receiver (*installed on the mobile object*) is now known as a **R**over **U**nit (RU), and the second receiver is known as the **B**ase **S**tation (BS). The BS is installed at a precisely known and accurately surveyed location within the tracked region. As the exact position of the BS is known, it is possible to calculate the true distance to each GPS satellite and further used to calculate the deviation (*difference*) by a simple subtraction ascertaining the deviation between the pseudo-rage and the true position value based upon the satellite signals. These compare its location against the known location applying the deviation to the GPS data acquired by the RU.
After the correction values have been received, the true distance can be determined using the measured pseudo-range, allowing the exact (*accurate to within a few millimeters*) object position to be calculated from the true distance recorded further eliminating almost all error deviations. Some deviation errors that may still exist could possibly come from either multi-path or receiver noise.

"*Differential correction techniques are used to enhance the quality of location data gathered using GPS receivers. Differential correction can be applied in real-time directly in the field or when post-processing data in the office. Although both methods are based on the same underlaying principles, each addresses different data sources and achieves different levels of accuracy. Combining both methods provides flexibility during collection and improves data integrity*" **[60]-(ESRI, GIS and Mapping Software, 2003)**.



[Figure 30]-(GPS and DGPS operational principle differences)

The two receivers (RU and BS) "receive" the same GPS signals containing (*in some cases*) the same measurement errors. As the BS is positioned in a fixed and known location it can collect the data referencing and comparing the received signals with its own "known" location, thus further being able to calculate the deviation error (*the difference between its own and the received signal measurements*) furthermore allowing the removal of this error resulting in correct positional information thus applying the same algorithm used on itself to the mobile receiver at the same point in time. Depending on a DGPS implementation handler (*system*) being in use, the measurement correction calculation can be performed either in real time (*correcting the data as it is recorded*) or by post processing (*correcting the data after is has been collected*). Both the BS and RU store the collected data on a local file where they can be accessed and compared with one another. In the case of real-time correction the files are compared and measurements are corrected as the data is recorded using transmitters and receivers that are built into the two (RU and BS) receivers, whereas during post processing correction of the files are compared using specialised ASW after the data has been collected. In the case of the PS, a post processing comparison would be enough to meet the system requirements.

There are two different principles within the DGPS process, including DGPS based upon the measurement of signal transit time that is accurate to approx. fl 1 m, and DGPS based upon the measurement of the carrier signal that is accurate to approx. 1 cm, with the process further being separated into three distinct areas, including, wide area DGPS, regional area DGPS and local area DGPS.

When measuring pseudo-range, accuracy of approximately 1 meter is possible, although is again for most systems not acceptable. For accurate results, the satellite signal carrier phase must be measured offering accuracy to within a few millimeters. The carrier phase process observes satellites at different times, continually and regularly comparing the RU with the BS both during and after measurements have been calculated. This results in the determination of a position to within millimeters.

In relation to the PS, the relevant DGPS process required (*taking into account the described principles and areas*) would be a local area DGPS process based upon the measurement of the carrier signal. As discussed, any error deviations that exist can be transmitted directly (*real-time*) or the corrections can be post processed, and based upon this principle the accuracy of such an architecture can can be improved to within just a few millimeters.

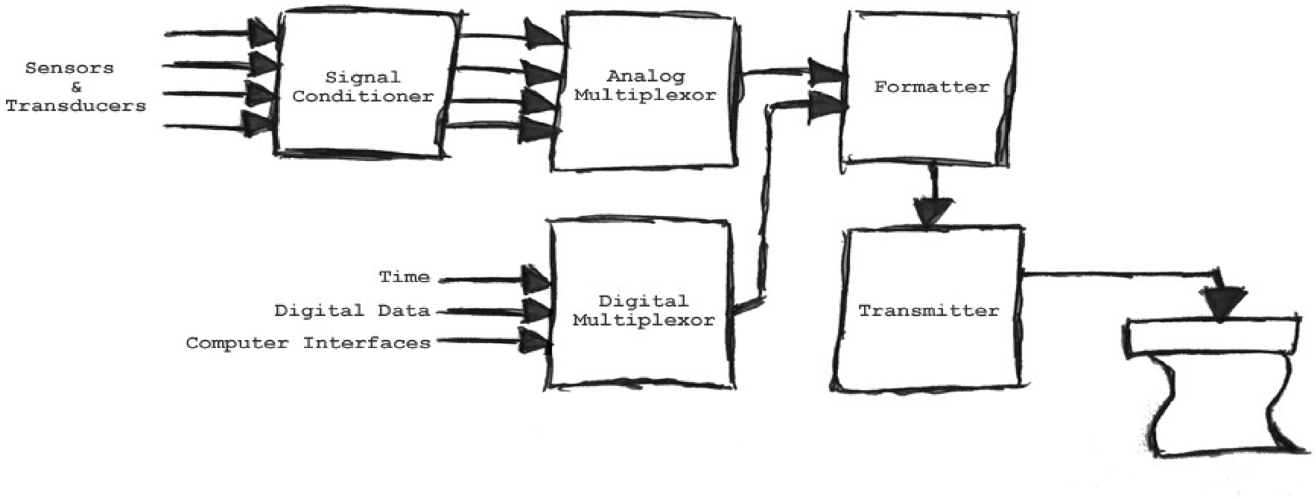
5.4.6 - Telematic Techniques

A telemetry system is a highly tuned communications technique that is used within systems where the measurement of data is acquired from remote locations (*such as sensors or GPS devices*) and are transmitted to a **T**elemetry **B**ase **S**tation (TBS) for further monitoring. In the case of the PS telematics will work very closely with and around the DAQ process.

[61]-(L-3 Communications Telemetry West, 2003) summaries the telemetry process within such systems explaining that telemetry "*... is the process by which an objects characteristics are measured, and the results transmitted to a distant station where they are displayed, recorded and analysed*". This also explains that with most applications it is too inefficient to send separate data packets along separate data channels for each measured value, thus these measurements are grouped by the telemetry process and further compressed into a data packet ready for transmission via a "single" stream. When the packet has arrived at its destination, it is unpacked and uncompressed and the original measurements are used for further data analysis.

There are many different and "complete" telemetry systems available with all with different standards, options and possibilities. Such devices are available to buy "off-the-shelf". They all share common elements, although each are individual and uniquely designed and configured to meet the requirements of a particular purpose or even the requirements of a particular application.

In general, telemetry systems usually consist of an input device (*transducer*), a transmission medium (*for example radio waves*), signal processing devices, and some sort of TBS in the form of a system or application that is used to record, store and render/display acquired data. In the case of the PS the transponder would take location values and compress them into a data packet that is further converted into a corresponding electrical signal ready for transmission where it can be further accessed and analysed accordingly.



[Figure 31]-(A typical telemetry system)

"*Data acquisition begins when sensors (aka transducers) measure the amount of a physical attribute and transform the measurement into an engineering value*". Sensors within such a system require voltage, with some sensors acquiring this voltage directly, and others requiring "excitation". Sensors that are connected to signal conditioners provide power for the sensors allowing them to work with, or modify signals so that they are compatible for further stages of acquisition (*within the previously discussed acquisition stack*). Within most (*efficient*) telemetry systems, a multiplexer is used to eliminate the need of maintaining a separate path for each series. The multiplexor measures all of the analogue voltages and outputs as a series of pulses, with each having a voltage that is relevant to to that of the current channel being measured. This process of merging into a signal stream is know as **T**ime **D**ivision **M**ultiplexing (TDM) **[61]-(L-3 Communications Telemtry West, 2003)**.

More relevant to the PS, are telemetry systems that have been specifically designed for use within motor-racing applications. The use of Telemetry within motor-racing allows race teams to acquire and interpret a great amount of data from a vehicle during both testing and racing scenarios, further allowing them to highlight problematic areas and improve both vehicle and driver/rider performance.
Examples of the sort of data that a telemetry system could handle would be measurements of acceleration, braking, suspension displacements and various temperatures. These measurements all depend on what DAQ hardware is in place and used to acquire such information. There are many different measurements that can be taken and many different DAQ devices available to acquire such data (*DAQ devices will be discussed later in this chapter*).
As discovered within the LR, advanced telematic systems are commonly and successfully used within Formula One (F1) racing, transmitting both recorded vehicle data (*as discussed*) and additionally driver data (*inputs and performance*) that can be further used for analysis, explaining happenings (good and bad), further highlighting areas to either prevent or excel such happenings.

More advance telemetry systems (*like those used within F1*) allow the possibly of **2**-**W**ay **T**elemetry

(2WT). 2WT is not required for the PS, however is an interesting concept. If a 2WT system was installed, engineers within a racetracks pits would be able to not only receive telematic data from the tracked motorcycle, but also send telematic data to the motorcycle as it is traveling around the racetrack. 2WT was (*and sometimes still is, however limited to certain rules that are in place*) used within F1 racing with regards to for example, car setup for particular corners on a racetrack.
Before telematics, race-cars were setup for a particular race, depending on factors such as weather conditions, temperatures, altitudes, cambers, surfaces and other related racetrack considerations, and once the race has started the setup had to stay the same for the remainder of the race (*apart from small changes during a pit stop*). With the introduction of 2WT, both the driver and the team engineers can transmits data packets to a race-cars engine management system (*engine mapping an activation/deactivation of car sensors*) and change the setup of a car as it travels around the racetrack, such as tightening and stabilizing the left hand suspension for a right hand curve (*and vice versa*) and changing a race-cars gearing and torque levels to improve factors such as corner exiting and top speeds on long straights, all in all enabling faster lap times.
Less sophisticated 2WT that could be used with the PS is that of simple communication to an onboard display installed on the motorcycle allowing the engineers/teacher to transmit messages to the rider as he/she travels around the racetrack **[62]-(The European Society of Telemetering, 2009)**.

With regard to the PS, an advanced (*however only 1-way*) telemetry system is planned to be installed and used, allowing data from the tracked motorcycle to be accurately and reliably transmitted (*using TDM*) to a TBS with the number of TBS's being used depending on the amount required to provide continuous coverage.
The TBS's will transmit both control and GPS differential correctional information to the motorcycles RU, within a time slot managed by a time synchronisation system. This will be used along side the telemetry system, slotting transmissions into a time synchronised process, further transmitting telematic data back to the TBS's. When the data is received from the system, it is both decompressed and unpacked with the unwanted packets being disposed of and the remaining data being used for analysis within the front end application.

5.4.7 - Data Acquisition Hardware – (Devices and Sensors)

Many DAQ devices exist today that can be used and installed on vehicles that communicate with the engine management system, acquiring relevant data when the vehicle is mobile. The demand of such DAQ within the motor-racing industry has resulted in specialist companies developing specialised devices and technologies. Selected companies and individual devices that can be used with motorcycles and potentially the PS will be investigated and compared in relation to how they could benefit and improve the system. Also, whether they could be used in terms of the acquisition of relevant, meaningful and helpful data that can be transmitted along additional telemetry channels and used for further analysis will be investigated.
As there are many existing DAQ systems and technologies (*receivers, loggers, sensors...*), a selection process has been completed including the systems that appear to be both the most relevant and appealing in terms of design, costs and additional factors relevant to the PS. These devices will use many of the technologies and architectures that have already been discussed, combining them together offering powerful measurement results and additionally offering (*from an investigation perspective*) an understanding of “real” system examples including how they are and can be used and additionally applied to the PS.

As mentioned, from the discussed literatures and systems, the RaceFX system is the favoured and most aadvanced system reviewed. The RaceFX system uses a very advanced GPS receiver and data logger (*the Novatel RT24 and IONGPPS system*) in which SportsVision have invested a lot of time and work, resulting in very accurate measurements and system possibilities. The devices used within the RaceFX system have been researched even further and as a conclusion are deemed too advanced for usage with the PS.
It may sound a little strange to deem a device or technology as too advanced, however in the case of the PS, what such solutions provide is far more than what is required in terms of possibilities and accuracy, and in this case it is deemed irrelevant to over complicate areas (*especially when they are not required*). Furthermore, the capabilities and solutions offered by such hardware involve increased cost. The devices used within the RaceFX system are very expensive, especially compared to other devices that are suitable and meet the systems requirements to a satisfactory level. Such systems are discussed and compared below.

More information regarding the discussed devices used within the RaceFX system can be found from the following resources: **[63]-(J Neuman et al, 1994)**, **[64]-(A Manz et al, 1996)**, **[65]-(T Ford et al, 2002)** and

[59]-(K Milnes et al, 2002).

Most of the selected systems and devices used have a GPS receiver built in, logging measurements and additional data acquired from further sensors to a storage on (*local to*) the device. Such systems are developed by either companies with the aim to sell and make money from them (*therefore the investigation may, at times, feel more "marketing" related opposed to "technical" related*) or by race teams in an effort to try and improve the performance of a vehicle and/or driver. In consequence such companies are quite secretive with regard to algorithms used to calculate necessary data measurements, however, it is still believed relevant and important to obtain a good understanding regarding what the systems can offer to the PS and how they can be used.
Due to both the investigation and review of existing systems and literatures that have been completed, it is believed that a good understanding and knowledge regarding how such systems work (*the internal maths and algorithms*) has already been established.

As documented within the project proposal, one of the main concerns (i*n relation to the PS*) and point of interest of such devices is the acquisition of locational measurements (*coordinates*) and therefore the GPS receiver. Before relevant systems are investigated further, it is believe necessary to investigate a typical GPS receiver and the basic fundamentals of how such a receiver functions.
Further Investigation into the fundamental workings of a GPS receivers can be found in **[Appendix 5]-(GPS Receiver Fundamentals).**

As well as GPS receivers, additional DAQ devices and sensors that are relevant to a system will also be considered within the investigation to find out if they might benefit the PS.
These additional devices themselves create new subject areas, ideally requiring a separate chapter for discussion, however due to the requirements and purposes of the PS, it is believed enough has been done at the this prototype stage to simply investigate the purpose of such devices and considering what more they could offer (*in terms of DAQ and how data can be used both with the PS's backed and fronted*) rather than a complete investigation into there foundations and lower-level workings.

5.4.7.1. - Nology G-Dyno Plus

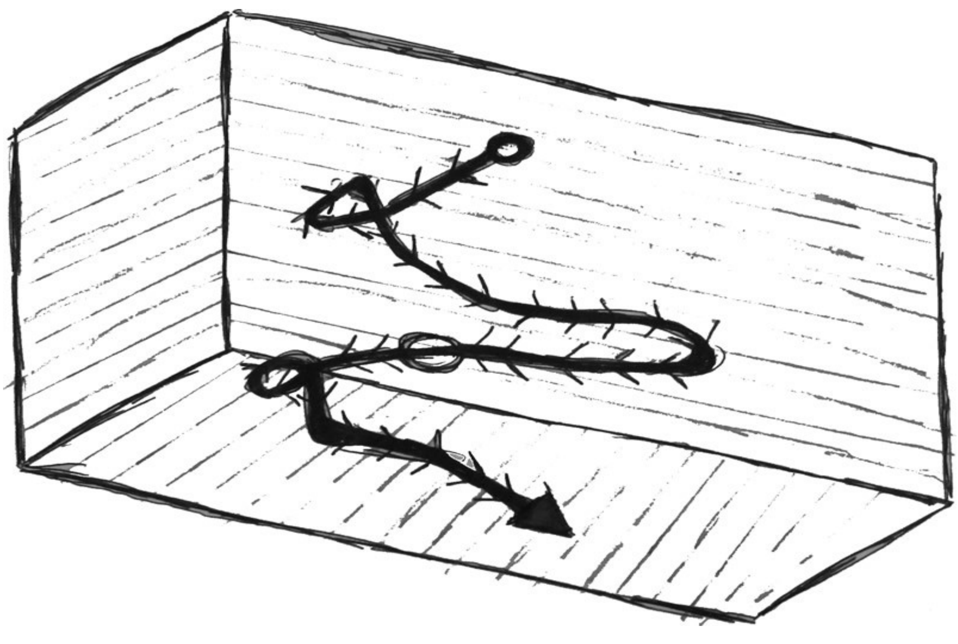
Nology is a company that originally specialised in performance ignition systems for race-cars, however they have expanded and are now offering a wide range racing performance parts and technologies, including the G-Dyno Plus system.
The G-Dyno system is a DAQ system with an on-board accelerometer and GPS receiver, and is described as providing data that can be used for analysis resulting in faster lap-times by finding all areas for improvement in terms of engine performance. The result being the ability to "squeeze" more power from an engine, and (*in relation to a motorcycle*) show what power is being used at both the "crank" and power displaced to the track surface.
The system is a full package including a GPS receiver and offering additional (*11 channel*) DAQ measurements to be acquired. This combination of both GPS and DAQ system creates a very admirable data logging system that could provide very useful analytical data, measuring both locational information and complete vehicle performance data using an Inertial Navigation System (INS) that together can automatically map a racetrack as a vehicle travels around.
The main DAQ measurements that the system provides includes, speed, distance, horsepower, torque, and G-Forces, these can all additionally be position related though the use of the GPS receiver.
The system can measure both the power used in different gears and/or the power and speed/time used throughout the gears, all making for very interesting analysis results. Furthermore, braking times and forces over distance can be recorded with cornering and handling forces all being measured at the same time. Advanced mathematics can then be used to calculate additional factors such as lean angles.
The device records and measures these factors as a vehicle travels around a racetrack, generating and storing the acquired data in a simple (*configurable*) format (*usually either text or CSV*), including an additional excel format, thus the data can be easily pumped into any (*relevant*) application and used for further analysis. The data is stored on a standard MMC 16mb card that allows up to 3 hours of recording, however this can easily be upgraded to a 128 card further allowing up to 12 hours of recording.

The GPS receiver used within the system is a custom "SirfStar 2" receiver that is optional with the system. The DAQ device can be used individually as a power meter or simple data logger, and can also be used with other GPS receivers (if preferred) making for a flexible architecture.
With the (*or any*) GPS receiver included in the system, location data can be measured and further used

to map the racetrack, thus allowing the acquired data to be analysed in relation to an objects position at a particular time on a racetrack. The system can be used to analyse the power usage during a lap (*at each individual section of the racetrack*) thus allowing both vehicle and driver performance to be tuned and compared using before and after data. This data can then all be used to improve rider performance and will overall help to reduce lap-times.

The G-Dyno Plus system measures both power and torque using a precision accelerometer recording both "*... forward and sideways (lateral) measurements every few milliseconds of vehicle movement*" **[66]-(Nology Systems, 2008)**.
To enable accurate results, the vehicles weight and other variables must be entered in to the system, and then using advanced physics and maths (*similar to what has already been discussed*) it is used to calculate the relevant data. The system can obtain not only the actual power output from an engine (*acquired form the engine management system*), but also using this data and advanced algorithms used with other acquired data (*including that from the accelerometer*) the power that is displaced to the racetracks surface can be obtained. This greatly enhances analysis data highlighting where power is not being used enough, and also being wasted.
Cornering measurements are calculated using the acquired lateral (*sideway*) acceleration data, allowing the G-force data to be collected and thus when displayed (*and if displayed correctly*) showing the forces experienced by a vehicle during cornering, further allowing the comparison of "*... front/rear handling setups, ...*" the checking of results and "*... adjustments to toe in/out, camber and caster*" **[66]-(Nology Systems, 2008)**.
The comparison of such data during analysis can not only highlight rider performance related issues, but also issues relating to the bike setup (*suspension and tire choice*) highlighting what works the best for a particular bike, rider and racetrack.
With regards to braking, the G-Dyno system measures braking in a similar way to that of acceleration, by using "deceleration" measurements. This data can help analyse factors such as braking effectiveness at different speeds and racetrack sections, and also factors again regarding bike setup, for example comparing data between different brake pads, fluids and hoses used with a particular setup.

When using the GPS receiver, racetrack mapping is possible. A vehicles position is automatically and consistently logged by the GPS receiver, which then allows all parameters including "*... RPM, speed, acceleration, and logged data such as AFR, throttle position etc, to be viewed against time and track position*" **[66]-(Nology Systems, 2008)**. As a vehicle travels around a racetrack a "track map" is dynamically created thus data can be checked against locational information relating to and further being compared against the racetrack itself. The system further measures a vehicles altitude, thus depending on the chosen ASW used, and its capabilities, this data (*along with the location information*) can be used to create a 3D racetrack model.



[Figure 32]-(Example 3D racetrack model possibilities)

[figure 32] shows an example of such a 3D model that can be created using the G-Dyno (and any similar systems) location and altitude measurements. Depending on the ASW being used and its capabilities, such a component, either built in, extended or even a “custom” component could be created thus representing any racetrack in 3D showing altitude and even surface cambers.

[66]-(Nology Systems, 2008)

Nology summarizes the G-Dyno system features as follows:

Measurements:
- -X and -Y acceleration ay 100 samples per second.
- 0-60mph / 0-100kmph timing, or any other speed combination (30-50, 50-70).
- Standing 1/8 or ¼ mile timings (mainly used for “drag” style racing).
- 60-0mph (or any other speed combination) braking times and distances.
- Horsepower and torque - real road horsepower.
- Before and after test of modifications.
- Peak and average cornering G forces.
- GPS location data.
- Record data from 11 other sensors.

Data Logging:
- 7 configurable channels of 8 bit analog data (*for example, air/fuel ratio, throttle position, brake pedal pressure, exhaust temperature sensors, oil pressure and knock sensor etc*) all recorded at 12.5 samples per second
- 4 channels of digital data (*injector pulse, speed sensors, gear sensors and lap counter etc*) all recorded at 100 samples per second
- Tachometer input (*speedometer, mileage, RPM etc...*)

The device itself is very simple just consisting of the required input and output slots and a simple on and off switch. The device is powered by 2x standard double A (AA) batteries that (*when not providing power ti the GPS receiver*) has a lifespan of up to 24 hours, and when powering the GPS receiver has a life span of 2 hours (*thus the GPS receiver proving very costly to power*). The device is small and compact, and very lightweight thus would fit nicely within a motorcycles sub frame (*either under the riders or passenger seat*).

GPS Receiver:
- 12 channels (*all-in-view*) tracking.
- Cold/warm/hot start time: 45/38/8 seconds.
- reacquisition time: 0.1 seconds.
- Accurate output signal aligned with GPS timing.
- trickle power enabled form power saving.
- Multi-path mitigation hardware.
- On-board DGPS and WASS demodulator.
- SiRFstarII architecture.
- Red/green (*fix available*) indicator.
- Fully waterproof.
- Rubberised magnetic base.

[66]-(Nology Systems, 2008)

The overall G-Dyno Plus system is very powerful offering a lot of measurement and other possibilities that the PS could benefit from. In its standard form, only the data logging device is available with the required items for installation and use. MMC card update and power supply updates are available as optional extras.
As discussed the system can be used with or without the GPS receiver, and can be used with any other GPS receiver (*the discussed receiver is recommended and available from Nology*) transmitting location information in the same way (*along an additional transmission channel*). The provided Sirfstar GPS receiver is of a high specification and could also be used alongside other DAQ devices, thus in general, the system follows a very flexible and expandable architecture allowing for easy customisations tailored towards specific systems, applications and uses, which overall makes for very useable hardware.

To obtain more detailed information about the G-Dyno Plus system, Nology have released the following paper: **[67]-(G-Dyno Plus with GPS, 2008)** explaining the system specifics in an interesting and detailed manner.
More detailed information regarding the discussed Sirfstar GPS receiver (*and other Sirfstar receivers*) can be found on the Sirfstar website: **[68]-(Sirf, 2009)**.

5.4.7.2 - RacePak G2X Data Logger

RacePak is a specialist data tracking company who's main domain is focussed within motor-racing having over 20 years experience in and around data system design and development. RacePak build and supply many devices and types of hardware aimed at improving data tracking from racing vehicles. Some of this hardware includes, different data loggers, vehicles dashes (*dash boards*), sensors and various other components within most motor-racing fields and environments including drag racing, road racing and off road racing.

The selected product of interest in relation to this chapter and the PS is RacePak's G2X Data Logger. The G2X Data Logger is aimed for use within road (*track*) racing, making it relevant to the PS. It is described as a compact (*although not as compact as other data loggers that have been investigated*) and robust data logger, consisting of many features, providing "*... an accurate, dependable and easy to use data logger*" **[69]-(RacePak Data Systems, 2009).** The G2X device has a built in GPS receiver, and can therefore interpret GPS satellite signals and can calculate a vehicles location and movement information. The device can also calculate other performance measurement parameters, including "*... track mapping, lap times/numbers/differences, segments times, miles per hour, lateral G-Force, and acceleration G-Force*" **[69]-(RacePak Data Systems, 2009).**
The device stores acquired data on an SD memory card, with the size of the card determining the amount of data that can be stored. Site an example stating that by using a 128mb SD card up to 20 hours of GPS data can be stored, which is believed to be very reasonable and worthwhile.

An addition that Racepak offer, is a visual display that can be installed on a vehicle (*for example in a motorcycles cock pit / dash layout area*) providing the driver/rider of the vehicle with performance related information, including the "*... display of lap times, lap number, per lap gain/loss, gear indicator, progressive shift light, battery voltage and G-Forces*" **[69]-(RacePak Data Systems, 2009)** as well as RPM information (*for vehicles with an ignition system*) with three additional inputs available accepting data from different/selected sensors, thus theoretically replacing the vehicles dashboard with a single and advance DAQ device.

RacePak summarises the G2X system features as follows:

- 12 v-Net channels and 4 internal channels available.
- V-Net channels have sample rates of up to 100 times per second.
- SD Memory card data storage, with the recording size depending on both the number of channels monitored and sample rates.
- Built in internal sensors measuring:
    - Battery voltage.
    - longitudinal g-meter (acceleration and deceleration).
    - Lateral g-meter (side-to-side motion).
    - GPS location data.

[69]-(RacePak Data Systems, 2009)

RacePak additionally provide a very similar, yet more advanced device called the "G2X Pro" which has been built upon the standard G2C device, this is more powerful plus it offers additional features including more data channels and additional and better sample rates:

- In total 71 data channels.
    - 59 v-Net channels.
    - 8 analogue channels.
    - 4 digital channels.
    - 4 internal channels.
- V-Net channels have sample rates of up to 100 times per second.
- Analigue channels have sample rates of up to 1000 times per second.
- Digital channels have sample rates of up to 100 times per second.
- Digital timing intervals within 1/1000th of a second.

[69]-(RacePak Data Systems, 2009)

"*The G2X-Pro builds on the G2X's already impressive capabilities by allowing the user to monitor up to 71 channels, while providing lap and segment timing along with speed and track mapping functions through GPS information*" **[69]-(RacePak Data Systems, 2009).** The G2X-Pro uses advanced V-Net plug and play technologies allowing data from up to 59 different sensors to be transmitted to the data logger. RacePak has designed and developed its own sensors that can be used (*as well as with any other "third party" sensors that may be prefferd*) to transmit such data. Some of RacePaks road racing related sensors include:

- Pressure sensors (*oil, carburetor, fuel pump, fuel nozzle, transmission, brakes, manifold (boost), manifold (absolute), manifold (vacuum), differential and pan*).
- Temperature Sensors (*engine components, cylinder head, manifold, engine fluids, exhaust gasses*).
- RPM sensors (*engine pickup, driveshaft pickup, clutch pickup, ring gear pickup*).
- Travel Sensors (*suspension/shock pickups, both travel and linear*).
- Air and fuel sensors (cylinders, fuel, weldment and plugs).

[69]-(RacePak Data Systems, 2009)

Data can be acquired form all the listed sensors as well as from additional sources. Those sources being endless, as long as a sensor is available or it can be adapted to get the required data. Such factors and be measured and transmitted along the discussed channel(s) where they can be handled appropriately (*in the case of the PS, pumped into an analytics application*).

More information about both the RacePak G2x and G2X Plus systems can be found at the RacePak website: **[69]-(RacePak Data Systems, 2009)**.

5.4.7.3 - Velocity Racing Data DL1

Velocity racing is a local (*UK based*) company that designs an develops GPS data logging systems, with the latest addition to the family being the DL1 model.
The DL1 is a stand-alone data logger that is small in size, light weight and compact and is described as easy to operate including just the essential input and output options. The device has 6 data channels that are user defined, thus any sensor based data can be chosen and transmitted over these 7 channels.
The device stores all acquired data on a compact Flash memory card like that of an SD card, with the amount of data being limited of the size of the memory card (*16mb, 32mb, 64mb, 128mb etc...*).
Built into the device is both a GPS receiver allowing location measurements to be calculated with the addition of a 2 axis G-force sensor together with additional inputs for measurements including RPM, wheels speed sensors (*for up to four wheels*) and one digital input line.
The DL1 is packaged and can be bought as a kit including all the bits required to run such a system reliably and accurately. The kit compromises of a power cable that is attached to the vehicles battery, an external and magnetic GPS antenna and terminal blocks which will allow additional sensors to be added.
Velocity Racing explains that the system is very simple to set up for use as "*... all you need to do, is add a Compact Flash card, and plug in the unit and you are ready to start logging data*" **[70]-(Veracity Racing Data, 2008)**. This statement is believed to be in relation to a standard car as the power connection has an optional connector that can fit into a standard car cigarette lighter, however in the case of top specification race-cars and motorcycles, cigarette lighters will not be in place, thus the system will have to be "manually" connected to the vehicles power source. Although still not complicated, this is not as easy as what is made out to be - a simple "plug-and-play" unit.

As described, the DL1 system is the latest in the family of data logging systems each building and improving upon one another, with some of its favored features (*already mentioned however being discussed more detail*) including the built in GPS receiver calculating an objects position 5 times per second, combined with the measurements from additional built in accelerometers (*2 axis accelerometers*) that "*... calculate very high accuracy positions and speeds at 100 times a second*" **[70]-(Veracity Racing Data, 2008)**.
The logging of data to memory is described as a robust and economical method for data logging systems, and is personally believed to be ideal for such systems since the memory can be easily and efficiently moved and passed between many different types of devices and hardware (*with almost all recent systems supporting such storage formats*). This method of storage also provides a decent amount of storage capacities additionally supporting fast upload and download times.
The DL1 system has and supports 8 high accuracy analog inputs one of which being a connection for the power supply input, measuring the battery life (*voltage*), with the remaining 7 inputs being available for user defined sensor inputs. "*All the inputs are 12-bit accuracy (4096 different levels), 3 of the external inputs have a maximum of 12v, the remaining 4 have a maximum input of 5v*" **[70]-(Veracity Racing Data, 2008)**.

The system offers (*in comparison to other related DAQ systems*) more reliable RMP measurements as 2 data channels are available, both accepting and calculating RPM inputs, one input being of a high-level nature source acquiring data from a vehicles ignition coil/system and/or the HT leads, and the other being of a more low-level nature source acquiring data from signals from the vehicles engine management system.
Up to four independent wheel/shaft inputs are available (*although only 2 being required within the PS*) that are used to measure the speed of each wheel, allowing the analysis of vehicle traction, further comparing the wheels throughout a lap highlighting drifts, slides, "wheelies" and more interestingly (*mainly concerning a motorcycles rear wheel*) slip ratios across a torque converter.
A configurable serial data input port is available using an RS232 input allowing data from an external source (*for example a special engine management system or any other specialist tools that are installed obtaining data that is required for analysis*) to be accepted and included for transmission. As well as the serial input port, the system offers a serial output port, which on reflection, actually makes the system very powerful in. It can not only output to the discussed onboard compact flash drive (*memory*), but also to an external/separate system or device, that could also be a DAQ device (*or even the same device*), configured to measure different factors and thus saving both its own acquired measurements, and the measurements pushed from an external serial port into its own internal serial port.

The DL1 system is the least powerful system that has been investigated, it does however use the lightest and smallest device that can be installed, and has been tested on motorcycles with accurate results.
Veracity ensure reliability of the system by performing various tests including calibrating, temperature testing and vibration testing on each individual unit. Due to the high demands of motor-racing, the requirements for such devices are very high, and therefore need to be vibration proof and very strong to ensure nothing fails during critical monitoring.
The system is described as very robust as it is "*... housed in a 2mm thick aluminum enclosure and carbon fibber end panels for very high impact resistance*" **[70]-(Veracity Racing Data, 2008).**
Although not relevant to the performance of the system, the DL1 tracker is believed to be the best looking tracker investigated which is (*from a personal*) level also a very important "selection" factor.
The system is easy to use having just a simple on/off switch.

A plus point of the DL1 device is that it uses a very up-to-date (l*atest generation*) processor (RISC) "*... which features both higher speed operation and flash upgradability ...*" **[70]-(Veracity Racing Data, 2008)**
As a new feature-as and when system upgrades are completed the device can be simply upgraded (*free of charge*) rather than having to be replaced, making for a very easy update and maintenance package/process.

More detailed information about the Veracity Racing Data DL1 system can be found on the Veracity Racing website: **[70]-(Veracity Racing Data, 2008).**

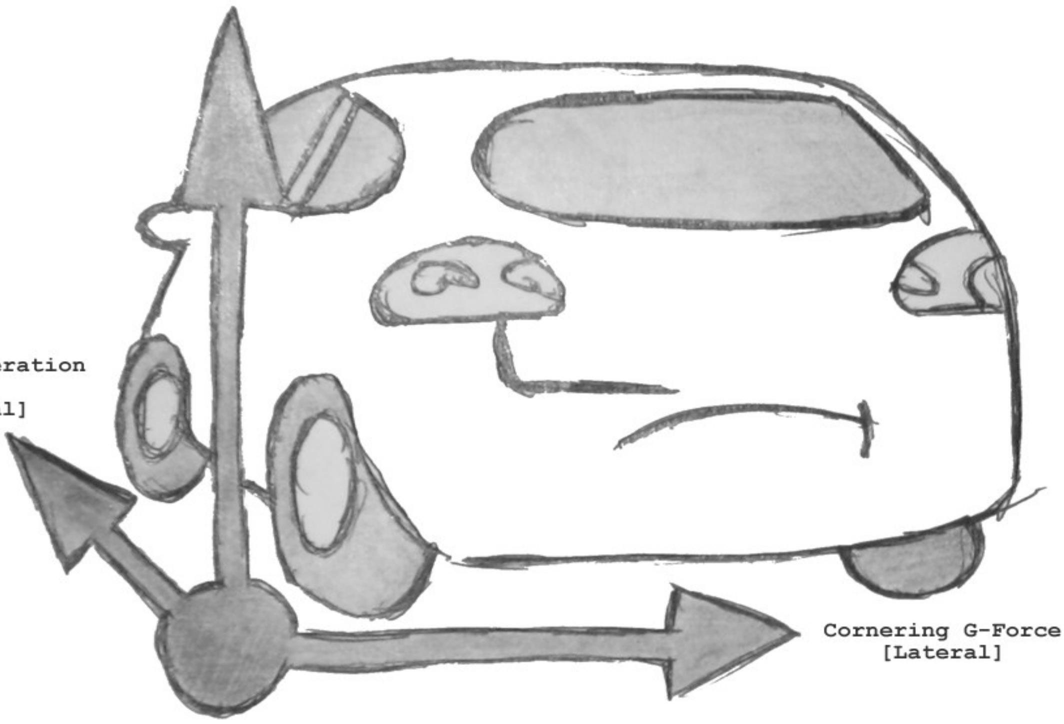
5.4.7.3.1 - Race Technology DAQ System

Race **T**echnology (RT) have created a DAQ system similar to that of the PS, it is however, aimed primarily at automobiles (*cars*). Within RT's system is incorporated the discussed DL1 device from Veracity Racing describing it as "*... a very highly flexible, powerful data logging system. It features an integrated, high accuracy 5Hz GPS receiver, with data logging directly to compact flash. It is ideally suited to automobile applications as well as bikes, boats and industrial/professional applications*" **[71]-(Race Technology, 2008)** thus confirming personal thoughts of implementing the system for motorcycles.

RT have created a very interesting and relevant system which at this stage of the investigation is believed to be not only deserving of further research but also a system from which a great deal could be learned in relation to the further development of the PS. Highlighted areas would include processes that could speed up development time and overall system performance and effectiveness.

The RT system is supplied with comprehensive software that is in communication with the DL1 device. The device uses its input serial port to accept configuration changes and setups that have been defined by a specifically designed "system configuration application" allowing a user to choose and set and save configurations relevant to a specific car and race. Once the configuration has been saved, the car can begin to travel around a racetrack where relevant measurements will be acquired and saved to the devices onboard memory, and further be outputted to either a lightweight monitoring application or specialist analytical software (ASW) similar to that required by the PS).
This ASW will be investigated further within this chapter where relevant.

Race Technology also explains how to enhance the DL1 hardware further in order to aqcuire additional analytical information including 6G acceleration, 20Hz Advanced GPS and Drift measurments.
The DL1 device comprises of a very high quality 3 axis accelerometer that is used to measure factors including both acceleration and braking G-forces (longitude), cornering G-forces (lateral) and up and down (altitude) G-forces (vertical).



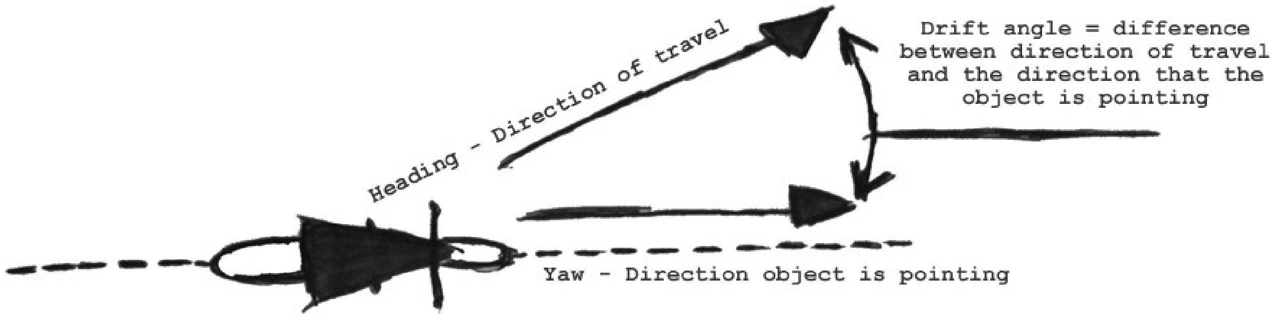
[Figure 33]-(DL1 G-Force measurements)

The DL1 device supports as standard G-forces of up to 2G, although can be upgraded to handle anything up to 6G, however this is not relevant for the PS as 2G is believed to be enough for any motorcycle application. This is confirmed by RT stating that 2G "*... is enough for any racing application where there are no large aerodynamic aids*" **[71]-(Race Technology, 2008)**.
The advanced GPS enhancement consists of a 20Hz option which the system engineers have worked upon allowing the system to calculate location positions at two times a second "*... with no interpolation or other tricks*" **[71]-(Race Technology, 2008)**. This 20Hz system is said to be very reliable and (*as expected*) accurate, being described by the system engineers as a "technological breakthrough" within such a positioning environment. This enhancement basically consists of a 2nd (*in-between*) processor (*computer*) that is used to handle the "calculation" of the additional "calculations", with this 2nd processing power making for the real (*noticeable*) performance difference/improvements.

RT describe that its 20Hz GPS system can be used in one of two ways, either in a standard 20Hz mode, or an advance PPP mode. When used in the standard mode, the system logs the raw GPS data from a GPS module inside the DL1 device, converting this raw GPS data into relevant speeds and positions as and when the GPS data is measured On the other hand when the system is used in its Advanced PPP mode, GPS corrections are downloaded from the internet before the data is processed, this resulting in better locational accuracy, although, and as discussed requiring an additional computer (*processing power*) which is connected to the Internet during the time of data processing.

RT further state that the "*... main benefit of this GPS system doesn't really come from the fact that it is 20Hz, this is simply a measure of the number of measurements that are taken every second ...*" and "*... the key to the performance of the system is the maths that is used to make the calculations - using the processing power of the PC to calculate the GPS position/speed*" **[71]-(Race Technology, 2008)** allowing real improvements to be established. RT also state that typically "*... the 20Hz option wont improve "poor" GPS data, it will make good GPSS data even better ...*" with the situation being the same of the PPP option as it "*... doesnt correct for a poor GPS lock, but instead just improves an* *already good data set*" **[71]-(Race Technology, 2008)** therefore the GPS antenna must be mounted correctly and efficiently to gain good GPS accuracy.

The drift enhancement that RT add to the DL1 system is an interesting concept, and one that could be adapted and used with the PS. A "gyro" is built into the DL1 device and is used to measure the direction in which a car is pointing. It copares this data with that of the acquired GPS data that is used to measure in which direction the car is travelling and thus with a simple subtraction calculation **((Heading – Direction of Travel) – (Yaw – Direction the vehicle is pointing))** the "drift" angle of the vehicle can be measured.



[Figure 34]-(Motorcycle drift calculation example)

This "drift" measurement is believed to be both very interesting and very beneficial to a motorcycle racing analytical system such as the PS, thus the advantages of such a "drift" measurement will be discussed later within this chapter.
The discussed drift angle additionally opens a further and interesting subject, possibly solving another of the PS requirements as documented in the Project Proposal, being that of the motorcycle lean angle.
The drift angle is calculated as a "horizontal" angle (drift) using a special "gyro" built into the device further performing relevant calculations. The motorcycle lean angle is fundamentally the same (requiring *the same calculations*) however at a "vertical" level opposed to a "horizontal" one.
The lean angle calculation is both an important and required subject and will therefore be picked up and discussed in more detail further on in the chapter.

More detailed information about the DL1 DAQ devices and Race Technology DAQ system can be found on both the Veracity Racing website: **[70]-(Veracity Racing Data, 2008)** and the Race Technology website: **[71]-(Race Technology, 2008)**

5.4.7.4 - TraqMate GPS Data Acquisition

The **T**raq**M**ate (TM) system is exactly as its name describes, a GPS based DAQ system. Like the "Velocity DL1" system, the acquired data is stored locally using built in compact flash memory.
The system is described as a vehicle DAQ "*... device that uses a high speed GPS receiver and high resolution accelerometers to track and record the speed of a vehicle, its absolute location and forces acting on the vehicle*" **[72]-(TraqMate, 2008)**.
TM explain and confirm both personal beliefs and what has been discovered during research into such systems, that due to both current and recent demands, such DAQ devices are now very accessible, are good values for money. They are easy to set up and in the case of the TM system, all that is required is that it is attached to a mobile object (*with as little as velcro*) and switched on.
The TM system uses the GPS receiver to (*like the Nology system*) map racetracks from lines taken as a vehicle travels around the racetrack, thus allowing the tracking of lines taken through corners, with additional data being acquired by the system including Speeds, forces (G-forces), braking and acceleration points. All this data acquisition is configurable, allowing users to "configure" and add "*... drivers, vehicles and tracks, quickly upload sessions and get immediate results*" **[72]-(TraqMate, 2008)** thus allowing laps between different driver/riders and vehicles to be acquired and further allowing comparisons of the data when pumped into the proposed analytical front end application.

TM state that "*... professional race teams can't live without data acquisition. They use it to hone the performance of their cars and drivers for that winning edge*" **[72]-(TraqMate, 2008)**.
This statement is agreed upon at a personal level, and was one of the main reasons that this subject area and space was chosen for further research and development.

DAQ is believed to be one of the most important sources and types of information that a racer (*driver/rider*), team and even teacher can have at their disposal when trying to improve performance and lap times of a vehicle travelling around a racetrack. When acquired data from a such system is pumped into an analytical application, its gives the described persons the ability to visually see what and where good and bad points were/occurred during a lap, and in the case of the TM system, allowing comparisons between both drivers and vehicles, highlighting why and how they are faster or slower and this in turn enables the improvement of "*... driving abilities far faster than learning by feel or worse – guessing*" **[72]-(TraqMate, 2008)**.

One advantage of the TM system is that is was designed, developed and tested by real racers and race teams and thus is tailored specifically to acquiring and finding unexpected data (*that other systems may overlook*), thus gaining that important extra second from a lap time that can make all the difference.
As described the system is very easy to install (*the easiest system so far investigated*), as it is as simple as installing the device somewhere in/on a vehicle, positioning the GPS receiver and turning it on (*using one large on/off button*). No additional sensors are required.

The system uses a powerful high-speed processor offering a base that can be easily built upon and upgraded over time as new and improved technologies and solutions become available. This processor is contained within the system **S**ensor **U**nit (SU) along with "non-volatile" memory, high resolution accelerometers, a high sensitivity GPS receiver and a RS232 serial interface, all in all making for a powerful system.
One thing that the system lacks in comparison to other investigated systems, and that is also considered an important requirement for the PS, is that of additional transmission channels allowing the transmission of additional data acquired from external sources/sensor (*that for whatever reasons may be included within the system acquiring further important data for analysis*).

A real world example of the TM system is SCCA Pro Racing (*one of the most competitive road racing series*). SCCA have chosen to use the TM system for use as a performance auditing tool within its "Speed World Challenge" GT and Touring car series.
"*The portable nature of TM and the fact that it can gather highly accurate information with in-car sensors made it an ideal choice for performance auditing*" **[72]-(TraqMate, 2008)** within the series. All the cars have the GPS antenna and TM device installed with the units themselves being randomly assigned to the different racecars at the different race meets (*like that of the discussed RaceFX system*).
The data acquired during the race is reviewed and analysed by the SCCA personnel to ensure that the cars are fairly matched in terms of setup and overall performance (*highlighting cars that are unnaturally over-performing in comparison to all other cars in a race*) thus highlighting suppositions and detecting unfair behaviour of racing teams. The aim of the system in this case being to keep the racing as fair and tight as possible, also making for a more entertaining and challenging series to watch and experience.

More detailed information about the DL1 DAQ device can be found on both the TraqMate website: **[72]-(TraqMate, 2008)**.

5.4.7.5 - Data Acquisition Hardware - Summary and comparisons

During the investigation into existing DAQ devices and systems that could be used and incorporated within the PS, it has become apparent just how useful they actually are. These confirm the stated reasons, needs and considerations documented within this paper regarding such systems.
Before such DAQ technologies were available, object tracking was very limited relying on just a stopwatch and lap-charts they being the only possible progress measurement method. If no progress was being made, it would be unclear as to where and for what reason. **Jeff Lorriman**, a professional racing driver states in an interview regarding DAQ systems, the old saying that "*... the car that feels the best, isn't necessarily the fastest*" which is a very worthwhile statement and one that has proved itself during years of testing within motor-racing. This statement also highlights the need for such advanced DAQ systems. When these (*more advanced systems*) became available, they were very expensive, and due to this expense and lack of knowledge (*at the time*) in terms of the systems accuracy and reliability, it was deemed too expansive for the average racer or race team to justify that expense. As technology has improved and matured, more systems and possibilities have become available (*such as the previously investigated systems*) and at more reasonable cost.
Such DAQ systems are today a very valuable tool within race development as they replace the need for a specialised race engineer to interpret data and highlight problematic areas. The system takes care of this itself, drawing race track maps, highlighting areas where a vehicle is fastest and its average

speeds, its cornering speeds and forces ..., all from (*in many cases*) a small and easy to install device that today is very reasonable in price.
Theoretically, data can be downloaded from such devices and pumped into an analytical tool where in effect, laps can be super-imposed onto a computer screen highlighting lap performance and areas that can be improved upon. For example, asking the questions, Can braking zones be extended? Can the power be put down earlier? These are all important factors that can be (*in many cases*) compared against lap after lapx after lap, segmenting both good and bad areas and overall improving skills and reducing lap times.

A concept highlight during the investigation of individual DAQ devices was that of “drifting” measurements. This is just one example of the many advantages and uses of such an analytical system, such as the PS and why there is a market and great potential existing for such a system.
In relation to the PS and motorcycle racing in general, this is a very interesting analytical concept, especially within recent motorcycle racing as due to personal knowledge and experience within the sport, “drifting” (*known as sliding*) can be both an advantageous and disadvantageous effect with some cases improving lap times and some reducing them.
As a measured metric, the drift angle of a bike can be used to identify bike setup issues and even different rider styles. More aggressive riders drift more, often both into (*on the brakes*) and out of (*on the power*) corners than less aggressive riders. The inclusion of drifting improves lap times, thus bike setups (*suspension settings*) can tailored to meet a riders (*both aggressive and less aggressive*) riding styles.
During a lap, when unexpected drifting occurs this could be highlighted within the analytics application and thus further solutions and explanations as to why the motorcycle drifted can be found. For example additional factors could be used and compared relevant to the time and place the drift occurred, including factors such as current speed, current lean angle, current power or braking displacements and even track surface cambers. Extreme drifts are also the cause of a lot of accidents (*especially when a motorcycle is under power during cornering*) commonly known as “High-Sides”, thus this data can also be used to monitor how drifts were caused, additionally and theoretically preventing such accidents from occuring.

The most interesting use of the drift measurement is believed to be within the testing and improvement of the latest motorcycle racing enhancements, being that of **T**raction **C**ontrol (TC). Most of todays top-end race specification motorcycles are fitted with a TC system with the aim of preventing a motorcycle sliding/drifting. An analytical application can be used to identify any drift measurements that may have existed during a lap(s) with a motorcycle having such a TC system installed, thus highlighting if or where and for what reasons the TC system may have failed/been overpowered.

The following comparison table has been created with its arguments tailored towards that of the PS requirements in regards to DAQ systems. The table will be used to compare the previously discussed and investigated DAQ systems to get a better (*more visual*) understanding as to how each system can benefit the PS:

	Nology G-Dyno Plus	**RacePak G2X**	**Velocity Racing DL1**	**Race Technology**	**TraqMate DAQ**
GPS Receiver:	* * * * *	* * * *	* * * *	* * * * *	* * * *
Data (Log File) Storage:	* * * * *	* * * * *	* * * * *	* * * * *	* * * * *
Ease of Use / Installation:	* * * *	* * *	* * * *	* * *	* * * * *
Data Channels:	* * *	* * * * *	* *	* * *	*
DAQ Measurements:	* * * * *	* * * * *	* *	* * * * *	* *
Track Mappings:	* * * * *	* * *	* * *	* * *	* * * * *
Lean Angle Calculation:	* *	* *	* *	* *	* *
Motorcycle Compatible:	* * * *	* * *	* * *	* * * * *	* * *
Size / Weight:	* * * *	* * *	* * * * *	* * * *	* * * * *
Cost / System Value:	* * *	* * * *	* * *	* * * *	* * * *
Overall Average:	*** * * ***	*** * * ***	*** * ***	*** * * ***	*** * * ***

Photographs of each investigated DAQ devices can be found in **[Appendix 6]-(Investigated Data Acquisition Device photos).**

The comparison table confirms personal thoughts regarding the investigated systems in that they are all (*although different*) very similar in terms of performance and possibilities, with the Veleocity Racing system falling slightly behind in comparison to the other systems.
As the comparison table shows, all systems are very strong. Although the results form the table are quite similar, with no particular system "standing out from the crowd", what the table does show and help to identify is in which particular area (*from the created arguments*) each system excels within. These areas will be further investigated helping to select the system that is most relevant for use with the PS (*if any*), further either justifying the comparison table and/or contradicting its results, overall identifying the strongest "contender" DAQ device.

All investigated devices have a built in GPS receiver (*this was a personal requirement when selecting/reviewing devices for further investigation*) it is preferred to use a combined DAQ, data logger and GPS device rather than having seperate hardware items being used for many reasons effecting installation speed and ease, maintenance, weight, aerodynamics. Another hindrance is finding space to install separate devices (*especially on a motorcycle where space is limited*). All of the standard GPS receivers within the devices are very similar in terms of specification (*power and accuracy*), however the Velocity Racing and TraqMate systems are slightly limited in comparison to the others as they do not have or use an external GPS antenna. As a result the satellite signals could (*and most likely will*) be restricted for these systems (*not as strong or accurate signal in comparison to the other systems*).
The Race Technology system has an upgrade available allowing the use of a claimed 20Hz Advanced GPS. This upgrade is very powerful with regard to both accuracy and performance, however it comes at a price since a second processor is required to maintain this accuracy adding additional complications, maintenance and overall cost. In relation to the PS this is all for something not required (*especially not at this prototype stage*), but is however an area for project expansion (*if required in the future*).

All the systems offer more than enough in terms of data storage (*the storage of the acquired data*) in terms of log files that are in most cases stored in a simple ".csv" or ".txt" format, saving the file to an on board (*and removable*) Compact Flash Memory with storage of up to (*typically*) 128mb translating to on “average” 15-20 hours of object tracking time.

All of the systems are reasonably easy to install and use. With regard to the PS, the device is to be installed on a motorcycle, and therefore it is required to be small, lightweight and compact as installation space on a motorcycle is limited. The only “ideal” installation place for such a device is somewhere within the motorcycles subframe (*under the riders seat or in the tail unit*) as due to aerodynamics, this is the part of the bike that is likely to have a “hollow” space for storage . Additionally the subframe is (*in most cases*) where the battery is positioned, thus allowing easy/short access to a power supply (*if required*).
The investigated systems all run from a built in battery as default, however they can be wired to a different power supply, and if it is easily possible to connect a device to a motorcycles on board power supply, this would be preferred. It would provide a more reliable power source than a built in battery and give a continuous supply of power without the risk of it running out. The size requirement for such a device is therefore even more important than initially thought due to device placement and connect-ability to the motorcycles power source.

All investigate devices are believed to be suitable for installment on and within a standard motorcycles subframe, with even the largest and most bulky of the devices, RacePak's G2X device squeezing in nicely. From all the systems, the TraqMate system excels most in terms of size, being the smallest and lightest of all the devices. It is however, also the least powerful device investigated, showing that even with todays technological advances, device size is still paying a hard penalty. The TraqMate device however, does excel in terms of ease of installation and use, simply requiring just a “male” velcro pad to be in place on a motorcycle, and a “female” velcro pad in place on the device itself, thus mating the two and fixing them together. It now requires just the flick of one switch to turn the system on, and thus to start tracking data.

At this prototype stage, it has been determined that it is not possible (*in terms of both time and cost*) for a track overlay or visual display of the racetrack (*as used within the discussed RaceFX system*) to be available in the PS's front-end application, however, as discovered within this investigation process, using GPS measurements and some simple maths, a “rough” track map can be established acquiring an objects location measurements as it travels, and further mapping them to the racetrack at the same time.
The Nology G-Dyno Plus and TraqMate systems offer this functionality as a feature, thus eliminating any additional (*client-side*) maths, as the devices themselves will perform these calculations locally.

All of the systems investigated are motorcycle compatible, however, all have been designed and developed with automobile (*car*) racing in mind. The Nology G-Dyno Plus and Race Technology systems do state that the systems can be used on other forms of transport (*including a motorcycle*) with the Race Technology system actually being tested on a motorcycle additionally showing successful and reliable results.
In theory, each system can be applied to a motorcycle, as the basic concept is the same (*in terms of location and further DAQ tracking and monitoring*). In some cases slight algorithm modifications are required (*for example the device maybe positioned slightly higher on a motorcycle than on a car*) thus any differences need to be taken into consideration. This should “theoretically” not cause any problems, as each device investigated is configurable in terms of installation settings (*including position and height*).

A factor deemed increasingly important for the PS is that of additional data channels, allowing further data to be acquired by additional sensors and transmitted within the same data packet.
The more data channels a device has available, the more (*external*) data measurements that are possible to be acquired, transmitted and further analysed in the front-end application. Additionally having spare data channels opens up more project progression options, measurements and functionalities.
The investigated devices differ with the amount of extra data channels available. The TraqMate device has no extra data channels available, whilst the enhanced RacePak G2X system has the most (71). The fact that the TraqMate system does not provide any additional data channels eliminates it from any further use and investigation since it prevents the system from being expanded or interactive with additional/external systems/sensors. With regard to the RacePak system, having 71 additional data channels this is an extreme case, although very advantageous, since the other devices have both a reasonable and satisfactory average of 11 additional data channels.

The cost of each system is roughly the same averaging at around $1000. As the system prices are so similar, cost is not an issue concerning or favouring a particular system and must be paid if such a system/device is wanted/required. This is the case in terms of the PS.

A factor that has been identified earlier within the project proposal, and returned to throughout the LR, and now during this research and investigation chapter, is that of the acquisition and calculation of a motorcycles lean angle measurement. None of the investigated systems calculate a lean angle, mainly due to the reason that they are not "specifically" designed towards motorcycles, and/or it is not deemed as a necessary measurement to be tracked within such systems. Within the PS the lean angle is an important requirement and will therefore require further investigation.

Overall the investigated systems are all very strong offering many differing areas in terms of possibilities and solutions to the the PS. Each system has its discussed advantageous areas with each being stronger in certain areas than others and vice versa. Overall they are a well balanced mix confirming that a robust selection process/review was completed before the selected systems were chosen for further review.
The preferred and chosen system for use with the PS, is the "Nology system", mainly due to its "all-round" nature thus being (*believed to be*) a "safe" and reliable choice in comparison to the other systems which excel rapidly within some areas but having noticeable fall-backs (*suffering*) within other areas.
There are many reasons that the Nology G-Dyno Plus system is preferred, including the fact that it has a strong built in GPS receiver with the addition of an extendable (*external*) GPS antenna that is not only very small, powerful and practical, but can also be positioned anywhere on a motorcycle. The position of the device is irrelevant and enables a strong (*best possible*) GPS satellite signal to be sent at all times during the tracking process. The system also offers many more additional DAQ measurements (*as standard*) in comparison to the other devices, along with the additional data channels provides for further analysis making the front-end of the PS application a lot stronger (*and more rich with information*). Another selling point of the system is that of the built-in (3-Dimensional) racetrack mapping system that is believed (at a personal level) to be very beneficial to the PS.
During investigation, the Nology system appeared correct in terms of the advertising and marketing it used in putting across its message to its users/customers. Advertising was ignored to a certain extent as considered irrelevant, since the systems technological capabilities have priority, however, with this in mind (*and in this case the marketing team for the Nology device did a good job as*) the advertising of the system did have an affect on the final selection.

All systems investigated are believed to be satisfactory in meeting the PS requirements, however some will obviously meet them to a higher standard/level than others, and the final selection *(the Nology G-Dyno Plus system*) is believed to meet the PS requirements in the best, most reliable, accurate and satisfactory way.

5.4.8 - Motorcycle Lean Angle Acquisition

One of the requirements set in the project proposal and discussed throughout this report, is the acquisition and calculation of a motorcycles lean angle.
Such a measurement would prove very advantageous at an analytical level as the art of "leaning" is a main factor in relation to a mokitrcycles cornering process. If the lean angle can be measured, it can be compared against G-Forces and speeds to calculate the optimum lean for particular concerns, highlighting where a rider needs to add more or less lean in comparison to power delivery until the optimum cornering performance is met.
It is believed that with the data that the investigated systems can already acquire, it is (*theoretically*) possible to calculate a motorcycles lean angle to a certain level of accuracy and reliability (*enough to get a general idea of a motorcycles lean angle at each segment of a racetrack*). As long as the algorithm used is accurate and reliable, then so should the result.

Using the discussed concept of distance, speed and acceleration (*similar to how the previously described drift measurement was calculated*), however translating these measurements into angles, the calculation of the lean angle of a motorcycle should be possible.
With the acquired location measurements (*-X and -Y coordinates*) the exact position of a motorcycle can be known, and in relation to the lean angle, how fast the angle is changing can be calculated. With the angle change speed and time measurements determined, the motorcycles lean angle can relatively easily be calculated. For example a default "start" angle would need to be provided (*assumed as zero*) and from this start angle, using time and location measurements and the angle change speed, the current lean angle can be calculated.
As mentioned it is the same concept as distance, speed and acceleration calculations, instead translated and applied to a motorcycles lean angle.

With the calculation of lean available, the measurement itself can be analysed within a front-end application as either raw data, or it could be used as a data provider for an advance "custom" component that visually represents the lean angle. For example an actual motorcycle could be modeled and could "lean" using the calculated measurements in relation to the current time and location.

Instead of having to calculate lean angle, by relying on the accuracy and reliability from both existing acquired measurements (*location and speed*) as well as the custom lean angle algorithms using these measurements (*additionally adding an extra server side calculations and costs*). An **I**nertial **M**easurement **U**nit (IMU) could be installed and used.

IMU devices are typically used (*and are a main component*) within **I**nertial **G**uidance **S**ystems (IGS) consisting of a built in accelerometer and gyroscope that sense and track and objects motion (*type, rate and direction*)**.** Data collected can then be used to display and analyse an objects orientation and motion throughout a "run".
In relation to the PS, a specialised IMU device could be focussed on the acquisition and calculation of lean angle measurements. Such a device could be installed to a motorcycle and its main role/objective would be to acquire such lean angle measurements, transmitting these measurements along the same data channel as any other information acquired from a specific DAQ device in use.

[73]-(PI Research, 2008) describe such and IMU device that is developed for used with motorcycles in mind, with the aim of calculating a motorcycles lean angle. The discussed IMU combines "*... a sensor-axel Gyroscopic, Accelerometer and earth-magnetic field data to give in addition to the sensor data, drift free 3D orientation*" **[73]-(PI Research describe, 2008)** thus resulting in the ability to both log and analyse every movement of a motorcycle.

The most high class motorcycle racing series (MotoGP) installs s10uch a device on all the machines participating in races. The device is used in a very extreme and demanding environments to monitor a motorcycles leans angle, and even in more advance scenarios can allow the difference between the motorcycles and the riders lean angle to be monitored.
Special 3-rate gyros are used within such a system to track rapidly changing orientation in 3D additionally measuring the directions of gravity and magnetic north thus providing a "stable reference". The Systems "real-time" algorithm communicates and uses the sensor information to calculate accurate 3D orientation with the additional of a "*... highly dynamic response and stability over time*" **[73]-(PI Research describe, 2008).**

A company called "Xsens" specialises in different and adaptable motion technologies aimed at both machine and human mobile tracking scenarios. Xsens has developed two devices that are interesting and relevant in relation to the PS, the "MTx" and the "MTi".
Both devices have be built up from the companies fundamental **M**achine Mo**T**ion (MT) technology, with the MTx being a "*... small and accurate "3DOF" Orientation tracker*" and the MTi being a "*... gyro enhanced altitude and heading reference system (AHRS)*" **[74]-(Xens Motion Technologies, 2009).**
Both devices include:

- Accurate full 360 degree 3D orientation output
- Highly dynamic response combined with long-term stability
- 3D Acceleration, 3D rate of turn and 3D earth magnetic field data
- Built in solid state miniature MEMS inertial sensors
- High update rate
- Very compact (*match box size*)

The MTi device is slightly more advanced offering more in terms of performance, accuracy and additional measurements. Both systems accept synchronisation pulse however the MTi device can also generate and transmit such pulses plus additionally including different digital output modes. The MTx device has to be specifically calibrated for factors such as 3D measurements/alignment, sensor cross-sensitivity and temperature, whereas the MTi device compensates these factors which make it more useable and even accurate.

The Additional digital output modes on the MTi device make it a far more appealing and relevant device for use with the PS. It proves to be very advantageous as such a device could be treated as a "child" and the acquired measurements could be transmitted over output modes to an external "parent"

device (*such as the described DAQ devices*) accepting measurements through the described inputs of such devices, further synchronising and packing this data together and transmitting it back to a CBS (*again using the previously described additional (telemetry) data transmission channels*).
This is just one example that shows how such a system architecture can be built and expanded upon allowing additional information to be calculated, collected and further transmitted for analysis.

As described, the MTi devices contain a low-power signal-processor providing drift free orientation as well as calibrated 3D acceleration, 3D rate of turn and 3D earth-magnetic field data.
The aim of such a device is to provide an accurate measurement unit that can be used in stabilisation for almost all human-controlled equipment including vehicles (*motorcycles*), and in the case of the PS, it would not have to handle any unexpected measurements, just simply acquire and use them for further analysis.
Xsens interestingly explained that the standard versions of the device contain "*... accelerometers with a full scale of 5G and gyroscopes with a full scale of 300 degrees per second*" **[74]-(Xens Motion Technologies, 2009).** These are more than powerful enough for use with the PS as such specifications are pushed further within methods of use providing higher G's such as aircraft monitoring, and if not the device can be upgraded to a full scale of up to18G and gyroscopes with a full scale of up to 12000 degrees per second, making for a very powerful device that can be tailored to meet specific application requirements and further configured to provide the most accurate results possible.

"*The MTi uses 3 rate gyros to track rapidly changing orientations in 3D and it measures the directions of gravity and magnetic north to provide a stable reference. The systems real time algorithm fuses the sensor information to calculate accurate 3D orientation, stable over time, with highly dynamic response rate*" **[74]-(Xens Motion Technologies, 2009).**

More information regarding the Xens MT systems can be found at **[74]-(Xens Motion Technologies, 2009)** more information regarding such devices, accelerometer and gyroscopes can be found at **[73]-(PI Research describe, 2008).**

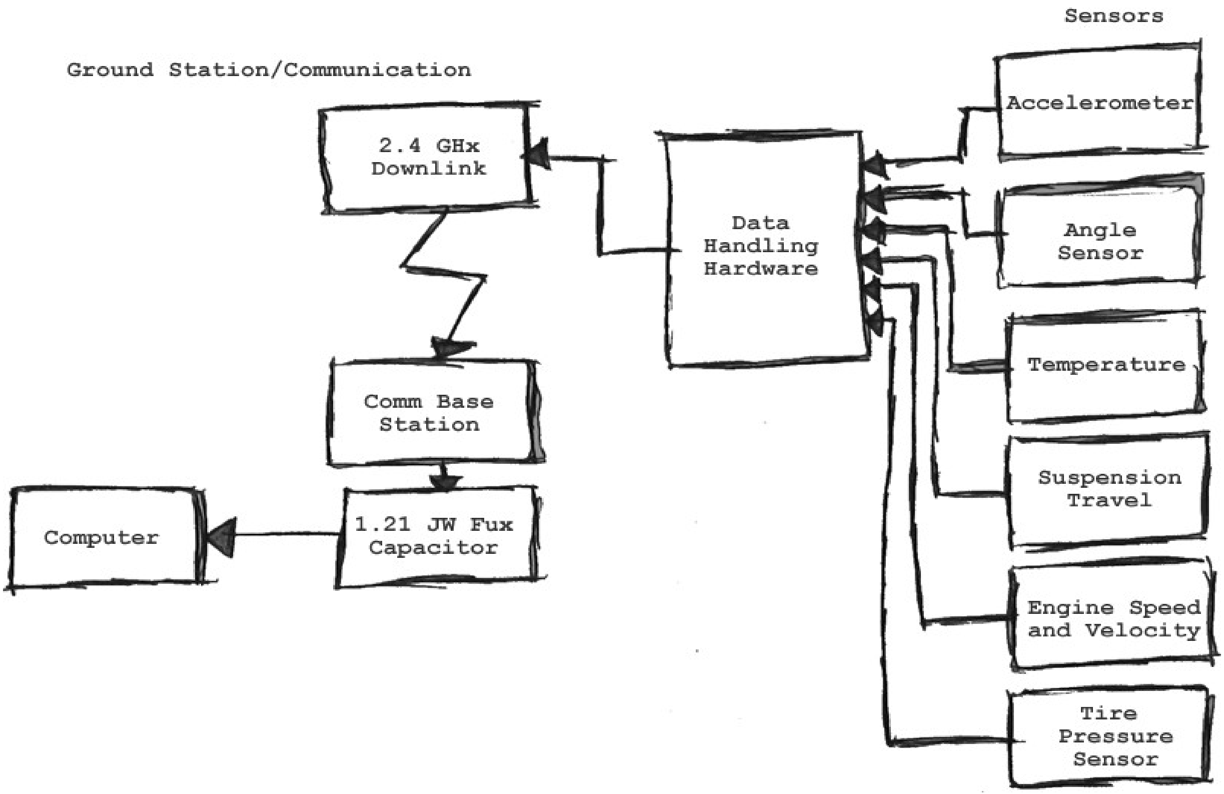
5.4.9 - Team Domination Racing Application

During this chapter, a similar and existing system (*the **T**eam **D**omination **R**acing (TDR) application*) has been discovered and is believed relevant for investigation and comparison, even at this late stage.
The TDR system is very similar to that of the PS and opens up some areas of concern that should ideally be reviewed. The fact that the system has been uncovered so late within research and investigation highlights (*to an extent*) that the undertaken literature and system search and investigation processes may have had some slight flaws. It demonstrates that not all outcomes were covered or exhausted (*including TDR system that is so similar to the PS aims and objectives*).
By finding this system it confirms and justifies both personal and discovered thoughts, opinions and arguments regarding such a systems existence and development.

"*Motorcycles are complex systems, containing many variables beyond what is displayed by standard gauges or rider intuition ... Understanding these variables becomes more important in racing situations, where a slight change can significantly alter outcomes*" **[75]-(Team Domination, 2006).**

The main variables measured within the application are acceleration, deceleration, lateral acceleration, lean angle, suspension travel and tyre temperature. Disregarding the final two (*suspension travel and tyre temperature*) all other measurements have been documented and put to one side for future project expansion.
To acquire these measurements, the system uses an accelerometer and IMU devices similar to those previously been discussed, with the addition of further sub-system specific sensors that are wired into the system. These will acquire additional data such as engine speed, velocity, throttle position and a (*current*) gear indicator, synchronising this data using relevant input sources and further transmitting the data over relevant and wireless data transmission channels, again, all being previously covered within the report.
To communicate and convert the acquired data, the system converts signals from analog to digital (*using a specialised converter*) and this converted digital signal is transmitted over I2C to a data handling unit where it is received and relevantly processed.
The justification of the transmission of a digital signal over that of an analogue signal is the same as was already discovered in relation to the PS. The quality of the digital signal transmission being far superior to that of analogue signal quality, thus providing better and more accurate/reliable (*more*

realistic) measurements for further use.



[Figure 35]-(Team Domination System Block Diagram)

Overall, the system includes A processing board, I2C inputs, RS-232 outputs, sensor selection multiplexing and software processor emulation. All the data from the different sensors (accelerometer*, angle sensor, temperature, suspension travel, engine speed and velocity, tire pressure sensor*) are collected at a local data handling (DAQ) device, where it is synchronised and transmitted to a CBS for storage (*and where any other relevant processing that is required can take place*) via a 2.4Ghz downlink. From the CBS it is then transmitted to a font-end application (computer) via a JW Flux Capacitor where it can be rendered/displayed relevantly.

The risks to the TDS system (and similarly the expected risks to the PS) include and are documented as:

- Extreme temperatures from both the engine and exhaust. The sensors and devices would either have to be out of the way of such components, designed to withstand such temperatures, or specific heat shielding around component should be installed to prevent further damage.
- Engine and road vibration. It has been discovered within the report that some devices contain anti-vibration controllers, or special mountings (*vibration damping mounts*) that can be used to eliminate the occurrence of such extreme vibrations on a device.
- Physical damage to devices and sensors in terms of an accident (*crash*). This is difficult to resolve, however, precautions can be put in place by using strong and robust devices and protecting them in either strong and protective cases or positioning them inside a motorcycles' frame (*where the highest amount of protection exists*). Should this be a big problem where there is a specific budget for a particular project, then to be safe, an additional budget should be created for replacing any damaged parts.
- The intricate dynamics of motion are too complex and depend upon the selected sensors and devices that are used. If the research and investigation into such devices in comparison to the requirements of a specific system have been completed properly, this should not be an issue. Completed reviews and investigation demonstrate that with the technology which exists today, the measurements required and produced by a motorcycle in an environment similar to that of the PS, can be handled at a more than satisfactory level, and can acquire data in both a reliable and accurate manner.

As discussed, although late in finding, this system has proved very useful and even confirms current thoughts and researched areas and problematic areas.
The fact that the system has been found so late also aids with the development of the PS since it has

confirmed that so far a great understanding and knowledge has been gained (*in the correct and relevant areas*) and that all system areas have been covered to a high level. These include many additional areas and concepts that have not been touched upon in the TDR system. In comparison (*and if successful*) the PS proves to be a very advanced and capable system that could offer much to people needing to use it for and within its designed uses and environment.
The TDR system does not include positioning measurements as it is purely a machine statistic DAQ system, thus data cannot be tracked or analysed in relation to a motorcycles positional information on a racetrack. This further limits the system greatly from an analytical point of view in comparison to that of a similar location based system (*such as the PS*).

5.5 - Application Software (ASW) Languages and Environments

One of the PS requirements is to create a prototype front end application demonstrating how the discussed data acquired from a motorcycle during a lap of a racetrack can be represented.
The application is to be analysis based and thus should convert and translate the acquired data into both understandable and meaningful information, maybe even representing the data visually (*as relevant graphs or charts*) allowing easy and detailed data analysis. This will further enable the fundamental aim of the system by "*highlighting problematic areas that can be improved, overall enhancing both rider and motorcycle performance and fundamentally reducing lap times*".

At this stage it is deemed relevant to refer back to and remember a valuable statement made within a paper discovered during the LR: "*When developing a DAQ system, be sure to thoroughly evaluate the SW. The HW components can be selected by determining the requirements of your system and ensuring that the HW specifications are compatible with your system and your needs. Carefully selecting proper SW, whether it be at driver level or application software can save you development time and money*" **[11]-(*National Instruments Corporation, 2007*).**

The statement above was found, documented and postponed for further thought until this section of the report. The statement is considered to be very valuable since what it describes justifies personal opinions on and around the subject area. It is agreed (*from previous personal experiences*) that selecting the wrong, or even not the best suited SW for a particular job, can waste both time and money, both of which are very valuable resources and should be used to the most efficient level within all areas of this (*and all*) project(*s*).

ASW offers x further (*more detailed*) options and possibilities as to how DAQ SW can be programmed using DSW to communicate and translate data that has been acquired by DAQ HW.
The ASW is the SW that the end user will see and interact with in such DAQ systems, and is described as SW that is used to display, analyse and store acquired data on a computer (*system*) by the means of either vendor supplied SW, or by developing a custom GUI meeting specific project requirements.
There are many different ASW frameworks, environments and languages available that can be used to develop ASW using data that has been acquired from a relevant back-end DAS. Such custom SW can be achieved either by using general purpose programming languages (*for example Basic, C, Java, Lisp, Pascal ...*), or by using specialised programming languages aimed at the development of such DAQ systems (*for example LabVIEW and MATLAB*) offering a "*... graphical methodology for developing complete instrumental, acquisition and control applications*" thus allowing the development of powerful data logging numerical based applications.

As discovered previously within this chapter, applications exist eliminating the need for SW to be developed in regards to the PS, since applications exist that can handle the acquired data from the moment it is transmitted from DAQ HW all the way through the life-cycle to the front-end GUI that the user views, analyses and/or interacts with. These applications would meet the basic PS SW requirements allowing the production of a basic analytics tool that renders and displays the acquired data. However, such applications are not developed or aimed at a specific area or requirement, so are therefore limited to a certain extent, further limiting ASW possibilities both as an analysis tool and product. For this reason (*and additional reasons including personal interest (both in possibility and outcome) as well as general interest in the development of such a tool*), the project will be extended at this stage allowing the development of not only a custom front-end GUI, but a complete and controllable DAQ SW supporting the data right through the acquisition life-cycle to the front end "analysis" application.

In relation to the PS, a custom application meeting the project requirements is to be developed using specialised SW. Different (*relevant*) languages that could possibly used within the PS will be selected and investigated further within this chapter.
Due to the application objectives (*as outlined in the project proposal*), the main requirements of such development tool(s) include:

- Access an external data source (*e.g. a database*).
- Support for graph/chart or any other data visualisation (*support analytical based components*).
- Allow component extensibility and custom component creation and compatibility.
- A fast and powerful rendering engine (*for effective graphing*)
- Support for Object Orientated Programming (*OOP*)
- Support either skinning or styling (*or both*)
- Provide good performance (*including data loading and rendering*)
- Support and handle large data sets (*caching or quantising methods*)
- Cost effective – be good value for money (*or even free*)
- Support a MacOS development environment (*as the application will be developed on MacOS*)
- Distribution support for all platforms and browsers (*depending on the required application runtime environment*)

The data that has been acquired from the motorcycle will be required to be stored somewhere, with the best suited source (*and the source used in other similar and general systems*) being a **D**ata**B**ase (DB). Most DB's have no strict limitations on where they can be stored, and can be further accessed, read and queried easily. The main and only requirements for a DB in relation to the PS, is that it should support large data sets and perform well.
There are many DB's currently available that meet the project requirements and could be used successfully with the PS, therefore DB's will be investigated further within this chapter.
The PS application will be used as an “analytical” tool based upon the acquired data, which will have the ability of converting and representing the data as meaningful information that can be used for in depth data analysis. “Analysis” application design is a large area that has been investigated in order to gain a general (*enough*) understanding of such development tools, however, this is yet another area that will require further investigation.
For effective analysis, it is beneficial if data can be represented visually, for example as images, graphs and charts (*line charts, area charts , bar charts, pie charts*) and even custom components that can be tailored towards specific data analysis thus supporting specific application requirements and/or specific data types. Such visual data rendering possibilities should be available within the chosen development tool, and should support interactive and dynamic data analysis. It is believed (*especially in the case of the PS*) that static data analysis, although good, and better than no data analysis, it is not enough, and that if the data is dynamic, it can be further interpolated thus making the analysis process more accurate, in depth and in general a richer and more satisfying user experience.
The ability of dynamic data analysis does however bring further challenges and requirements to the proposed development tool, in that it is required to handle the movement and rendering of (*what can be sometimes*) large amounts of data points. It will therefore require a powerful and reliable data and rendering engine, and further (*possibly*) be advance enough to have its own on-board caching system and or data quantising system. Such requirements are a lot to ask from such a tool (*especially a low cost or even open source frameworks*) and therefore if these requirements are not supported, the support or inclusion of relevant third party data handling tools, or the creation of custom handling tools that can be used within the framework, will be accepted.

5.5.1 - Programming approaches

There are several directions computer programming has taken over the years, two of the newer (*more commonly used*) and popular approaches relevant to this project include:

- Structured programming
- **O**bject **O**riented **P**rogramming (OOP)

Further information regarding both Structured programming and OOP approaches, highlighting both the strengths and weaknesses of each, further comparing the two together justifying the best for use with the PS can be found in **[Appendix 7]-(Programming Approaches).**

For the development of the PS, an OOP language and design pattern is preferred (on a personal level) rather than a structured top down programming approach for many reasons. These include those of inheritance, data encapsulation and the fact that it is believed to be the most practical and efficient way of programming large data intensive applications.

"*The concepts of OOP have been around for over four decades. Initially developed in the field of artificial intelligence, OOP was embraced by Xerox as a means of developing systems that better reflected real life needs and were more user friendly. OOP's popularity and sophistication has increased in the past several years as businesses are abandoning their mainframe systems and incorporating more client-server models to run their businesses and are integrating web technology as a business tool. A change in the overall pace of business has also contributed to the increased popularity of OOP programming. One of the primary features of OOP is its relative flexibility and adaptability to changing business needs*" **[76]-(AWI TechNet, 2001).**

The use of OOP framework will require a lot more thought and programming to initially build the application foundations, however, it will offer many benefits in the long time including faster and easier maintenance and expansion possibilities. In theory, if the application is treated as, and built using objects, and this process is done in a correct and efficient manner using class interfaces and factories, the addition of new features and components should be as simple as creating the component (*new classes*) as an object and wiring up these objects with the existing infrastructure *(like building blocks*).

More information can be found regarding both OOP and structured programming approaches in **[Appendix 7]-(Programming Approaches).**

5.5.2 - Data-Flow Programing

The initial thought when considering a programming language for use with such a DAQ and analytics based system is that of a **D**ata-**F**low **P**rogramming (DFP) language, implementing a DFP architecture and principles.
The use of DFP allows applications to be modeled conceptually as a detailed canvas (*for example a directed graph*) with data flowing between application operations. The DFP language was originally developed to bring "*... functional concepts to a language that is more adapt to numeric processing*" **[77]-(W Sutherland, 1966).**
Most programming languages use a different paradigm (*imperative programming model*) to that of DFP languages, in which the application is modeled as a specific set of operations that are performed (*with the data not being visible*) during the programming process, whereas, during programming using DFP languages, it is the data itself that is being worked with.
Sources explain, and it is agreed that a key concept within any computer related programming is the idea of "states", i.e. knowing the state of data and/or of an application during the development life-cycle**.** They also explain that when using imperative programming models (*opposed to data-flow models*) the state is often hidden, thus slowing the development process, causing a programmer to add additional (*unnecessary*) code to determine what data parts are important. Much of this code will not be required, thus decreasing code efficiency and cleanness proving expensive regarding both performance and debugging factors.
In many cases most data handling applications (*disregarding the language in which it was developed*) is believed to obtain data as an input, perform some "magic" (*process the data*) and output (*render*) the data to a front end application, which in most cases is a realistic scenario, pushing data between tools and processing the data through each step. DFP languages allow data interaction during the processing stage, obtaining the data, again as an input however visually showing the data (*on screen*) illustrating how the data is and can be used and modified within an application, further highlighting the data flow.
The advantage of this over conventional programming is that the data can be viewed (*in real-time*) as the application executes, and can therefore be handled and used as and when the data arrives, thus being a parallel process, opposed to creating handlers and performing stepping sequences through the data having to test and cater for different cases or behaviours (*as a series of instructions*), further organising and ordering each case for further data handling.
Due to advances within DFP, **G**raphical **P**rogramming (GP) is available allowing programmers to create applications by drawing the application rather than coding it. Data-flow diagrams can be created using advanced data-flow SW to link (*wire*) up the different areas of the different diagrams. The SW will effectively create "behind the scenes" the code required (*which should be very efficient code*) from the diagrams, further building and compiling the code allowing the application to run.

5.5.3 - Programming Language Selection

Although DFP offers a lot of benefits to DAQ application programming, a more structured and code based controlled (*imperative based*) language is preferred. However, due to the success of such DFP languages and their popularity and usage quantities within the DAQ application domain, they will not be excluded from investigation or possible usage with the PS, as it is believed (*for the reasons discussed*) that such a language could benefit the PS application greatly. Although having many advantages it is believed to be restrictive in terms of component controllability and extensibility. If components could be created or handled at a code level, with the allowance of additional extensibility and adaptability options, it would incur more flexibility. Therefore, an ideal solution would be a to find a balance between the two (a DFP language with additional code/component control and extensibility).

DFP languages that allow such a "mix" sharing similar features to that of a more functional based language will be further considered for selection, since such language bases are developed in order to bring functionalities and concepts to data based languages, further enhancing the control of data based applications and numerical analysis. This will provide advanced design and analysis techniques and more accurate solutions to and within data based SW development.
Most of the available (*and relevant*) programming languages (*especially data flow languages*) have such numeric analysis solutions, algorithms and libraries built in as standard, with the example of the "NetLib" repository (*commonly used within both Fortran and C programming environments*) consisting of a collection of routines and classes to handle such numerical complexities.
There are many popular and commonly used data flow and numerical computing applications that can be used to program such analytical (*data based*) application including LabVIEW and MATLAB (*being the most popular*) as well as "S-Plus" and even some open source examples including "Freemant", "GNU Octive" and "R". Each tool offers similar outcomes and possibilities, although to different standards and performance figures, using different methods of development.

Due to the amount of SW, frameworks and programming languages (*development tools*) available that could be used, a simple (*yet effective and detailed*) selection process has been completed covering both the discussed SW, as well as many other (*common, interesting and relevant tools*), scanning each and identifying areas both relevant and beneficial to the PS development, eliminating irrelevant and disliked tools, investigating further into preferred and more relevant areas.

Because of the discussed concerns regarding DFP languages, only the main (*leading*) two DFP tools (LabVIEW and MATLAB) have been selected for further investigation in order to (*initially*) get a better understanding of how they work and what their benefits are over more traditional and functional based programming languages.
If, after further DFP investigation, a data flow route is preferred, DFP languages will be investigated and compared further.

5.5.3.1 - LabVIEW

LABoratory **V**irtual **I**nstrumentation **E**ngineering **W**orkbench (LABVIEW) is a DFP language developed by (*a popular and already discussed company within the mobile domain*) **N**ational **I**nstruments (NI), and is both a strong and dominant platform and development environment for a visual programming language (DFP). Of all the development tools investigated, LabVIEW is both the most common (*popular*) and powerful tool available for the development of such a data-based analytical applications, described as being used both regularly and successfully within DAQ based applications as well as instrument control and industrial automation.

[79]-(National Instruments, 2009) state that "*... for more more than 20 years, NI LabVIEW graphical programming has revolutionised the development of test, measurement and control applications. Regardless of experience, engineers and scientists can rapidly and cost effectively interface with measurement and control HW, analyse data, share results and distribute systems*". This sounds very promising in relation to the requirements of such a development tool required by the PS, as today many solutions to such problematic areas, are visual solutions such as graphs and charts.

As the standard processor (*computer*) has become a very powerful tool for creating measurements and automation applications, along with other custom computer based solutions, it requires a certain level of SW development that can prove challenging. Due to this reason NI developed LabVIEW which is described by **[79]-(National Instruments, 2009)** as an intuitive graphical programming language deigned to remove the need to focus on areas like thread management. LabVIEW provides connectivity with thousands of measurement and control devices (*like the discussed DAQ devices*) further including hundreds of analysis functions storing data in multiple formats (*simple files, reports and databases*).

LabVIEW is separated into four main usage areas including, **Graphical Programming**, **High Level**

Development Tools, **Built in Measurement and Analysis Functions, Multi-platform** and **Embedded Devices.** Not being strictly separated they allow areas to communicate and operate (*overlap with one another*), with the most relevant area in relation to the PS being **Graphical Programming.** This can be further broken down into areas including Automated Test and Control, Data Acquisition, **Industrial Measurements**, **Control Embedded Design** and **Academia**, thus showing the size and power that LabVIEW offers and supports, additionally explaining usage quantity and overall success rates.

From the LabVIEW Graphical Programing sub-areas, the area of Data Acquisition (DAQ) is the area relevant to the PS.
Such an area handles and allows generation of SW including all test, control and embedded applications allowing live data to be acquired and displaying this data within a custom related interface based upon and requiring:

- Physical measurements
- Analysis and Signal Processing
- Decision making
- Data Logging
- Report Generation

[78]-(M Neil, 2005)

LabVIEW can be used to create data driven applications using a visual approach in terms of a block diagram configured to retrieve and process data, further pushing and using the data within a front-end **U**ser **I**nterface (UI).
Execution of an application is determined by the structure of a specific block diagram, in which blocks and function nodes (*block diagram children*) within the diagram can be connected by drawing (*linking*) nodes, thus representing and propagating variables with nodes further executing them as soon as data becomes available to a specific node.

A LabVIEW application can be connected to, and acquire data and measurements from almost any sensor, which in relation to the PS could be GPS location measurements or that of the discussed additional measurements such as RPM, Throttle position and lean angles, enable the creation of a fully functional measurement application with built in analysis tools with the addition of customisable UI that is connected to and powered by the block diagram.

LabVIEW is described as taking more traditional programming languages and making them easier to use by offering configuration based tools that instead of code, to program such acquisition applications and components, act as a high-level programming assistant, offering and combining flexibility and scaleability to an application development.

The discussed block diagram is one of three areas of the LabVIEW development process. LabVIEW programs are known as **V**irtual **I**nstruments (VI) and consist of three individual areas, including the discussed block diagrams as well as a **F**ront (*interface*) **P**anel (FP) and **C**onnector **P**anel (CP).
The FP can either be used as a programatic interface in terms of dragging and dropping components and objects onto a "canvas" allowing relevant positioning and scaling or they can be added to the block diagram as a node where its settings can be configured through dialogues and menu items (CP).

The graphical (*data-flow*) approach of application development that LabVIEW offers, is advantageous for non-programmers allowing them to build advanced DAQ applications in a simple drag and drop method. They can then use representations of equipment and components that they can relate to and understand (*recognise and identify*). This will additionally also benefit advanced programmers in speeding up application development processes as individual component and sections (*classes, interfaces and factories*) do not have to be hard-coded.

As a complete development tool, LabVIEW offers many benefits to DAQ applications over other existing and similar (*competitor*) tools making it a leader within its market and application domain. Some of these beneficial areas include:

- Extensive application support.
- Fast and effective application development for both experienced and non-experience programmers.
- Platform independent (*can run, and has been tested on all platforms*)
- Includes a built in compiler thus producing native code
- All graphical representations and data flow diagrams are translated (*converted*) into executable machine code (*using syntax interpretation*).

- Has many additional and advanced (*internal and external*) libraries available for inclusion, including a huge number of components, signal generation and conditioning factors, mathematics, statistics and analytical and graphical interface tools.
- Can include/import the ever popular "MathLab" (*from an external library*) in the form of a text based programming component that can be used to add additional functionality and handling factors such as signal processing and data analysis, further being integrated using script nodes within the application block diagram.
- Contains a built in run-time engine improving compile times, and thus development (*debugging*) and testing times
- Supports OOP allowing code re-use without any modifications and treating each node within a block diagram as an object, thus enhancing possibilities for project progression and expansion

A description (*walk through*) of the process required when creating a simple data based (flow) application using LabVIEW can be found in **[Appendix 8]-(Simple LabVIEW Application Development Walkthrough)**.

5.5.3.2 - MATLAB

MATrix **LAB**oratory (MATLAB) was invented and developed by "Cleve Moore" and is today maintained by a company called "The MathWorks", and like LabVIEW is a very large and powerful product, and can be separated into the following categories **Math and Optimisation**, Statistics and Data Analysis, **Control System Design and Analysis**, **Signal Processing and communication**, Image Processing, **Test and Measurement**, **Computational Biology**, **Financial Modeling and Analysis**, **Application Deployment** and **Database Connectivity and Reporting**).

MATLAB is a numerical based computing environment and DFP language and is an example of the discussed "mixed" development tool using a balance of both data flow and traditional (*coding*) based techniques.

MATLAB is described to be the language of technical computing, being a "*... high-level language and interactive environment that enables ... computationally intensive tasks to be performed faster than with traditional programming languages such as C, C++ and Fortran*" **[80]-(The MathWorks, 2009)**.

MATLAB can be used to develop a wide range of applications, including those that fall under the above categories and can be extended into specific areas. It can also be tailored towards specific application domains, using custom classes and available development aids and "add-ons".

An advantage that MATLAB has over LabVIEW is its flexibility. LabVIEW is very contained working with its own components and internal functionaries, whereas MATLAB provides additional features allowing work and projects to be both documented and shared (*integrated*) across different programming languages and applications. MATLAB additionally allows generated algorithms to be distributed and used with in other applications and areas.

MathWorks have documented the key MATLAB features to include:

- High level language for technical computing.
- Development environment for managing code, files and data.
- interactive tools for interactive exploration, design and problem solving.
- Mathematical functions for linear algebra, statistics, analysis, filtering, Optimisation and numerical integration.
- 2D and 3D graphic functions for visualising data.
- Tools for building custom **G**raphical **U**ser **I**nterfaces (GUI's).
- Functions for integrating MATLAB based algorithms with external applications and languages such as C, C++ Fortran, Java and Microsoft Excel.

[80]-(The MathWorks, 2009)

MATLAB provides a high-level DFP environment supporting both vector and matrix operations enabling advanced algorithms and applications to be quickly (*rapidly*) developed in comparison to traditional programming methods. This is because low level administrative tasks (*such as looping, variable declarations, typing and allocation memory management*) are not required, since MATLAB removes this functionality from the programmer taking care and handling them itself, however, at the same time having "*... all the features of a traditional programming language, including mathematic operators, flow control, data structures, data types*" **[80]-(The MathWorks, 2009)** plus it has OO and debugging features.

Another advantage that MATLAB has over LabVIEW is that is contains its own (*limited*) syntax known as "M-Code" that in its simplest form can be programmed and executed from the command line, or edited in the MATLAB M-Code editor (*which is basically just a simple text file editor*).
There is not much support in terms of documentation or more advanced M-Code editors existing as MATLAB (like LabVIEW) specialises in DFP. They appear to be trying to push both users and programmers down a visual programming path when developing MATLAB applications however, the fact that such syntax is both available and can be modified (*to an extent*) is in relation to the PS (*and personal feelings*) a positive and advantageous point.
An additional advantage that MATLAB includes is its built in and easy manipulation possibilities of matrices, advance algorithm creation and implementation. It allows the plotting of both data and functions, plus interfacing possibilities with other applications and programming languages.

The discussed matrix handling and manipulation is a very powerful feature in regards to data analysis, as visual components (*such as graphs and charts*) would have a **D**ata **P**rovider (DP) being of a matrix type (*in most cases being either a 2 or multidimensional array*) and it is this DP that is manipulated. The elements (children) within the DP can be modified and changed (*calculated, rounded, floored*) and/or removed or ignored depending on any configured filters or limitations in place.

The UI within MATLAB is created (like LabVIEW) using a **G**raphical **U**ser **I**nterface **D**evelopment **E**nvironment (GUIDE) acting as a canvas on which components can be "dragged and dropped", positioned and scaled. It further supplies a range components that can be used (*added to the canvas*) including graphs, menus, containers, sliders and filters **...** MATLAB does however, also allow the programmer to create GUI's at a programatic level, however limiting the code to components and functions specific to the MATLAB framework. Theoretically this has no real gain to that of using the provided GUIDE as using the GUIDE will produce the same front end result using (effectively) the same back-end code but in less time.

In relation to the PS, MATLAB support the development of DAQ applications throughout the complete life-cycle from the acquisition of data from external devices through to the presenting and rendering the data in the UI**.** It also in-between handles all the data processing and required numerical analysis, overall making it a very powerful development tool.
MATLAB is not only limited to data access from external DAQ devices, but can also read data in from a variety of other external sources including most popular file formats (".txt", ".csv", ".XML" and Microsoft Excel files) as well as a variety of DB's. Data can either be acquired from a devices (*computers*) serial port or in some cases can even be streamed directly into MATLAB and analysed immediately (*in real-time*).

As already mentioned, MATLAB has many built in components that can used and represented in a front end UI including data analysis visualisations that would be relevant for use within the PS providing graphs, charts and general data plotting components. Such components include both 2D and 3D powerful plotting and visualisation functionality, multi axis graphs and shape drawing tools all being possible to be exported to an external file for uses such as print or further forwarding (*via email*).

Like LabVIEW, MATLAB has its own built in mathematical, statistical and engineering functions contained in separate libraries that can be imported into specific projects as and when required. These libraries control factors such as data typing, matrix manipulations (*regarding components data providers*), interpolation, data analysis and filtering, data and code optimization, numerical integration and differential equations, all again making MATLAB a very powerful and flexible tool that could greatly benefit the PS (*and does similar systems*).

As discussed, a great feature of MATLAB is that it can share results (*outputs)* with other applications, for example results and algorithms created and calculated within MATLAB can be exported and used to provide and calculate data for use with other packages and reports. Additionally it will allow the results to (*in some cases*) be exported in HTML and LaTEX formats, and thus published to the web.
Additionally "*... MATLAB provides functions for integrating C and C++ code, Fortran code, COM objects and Java code*" **[80]-(The MathWorks, 2009)** DLL's, Java classes and active X controls can be used, and using the MATLAB library, such classes and controls can also call MATLAB from C, C++ and Fortran code.

[80]-(The MathWorks, 2009)

5.5.3.3 - Data Flow Programming Language Comparison

The two discussed DFP languages (LabVIEW and MATLAB) will be compared in relation to the PS, taking into account the set requirements of such a language for use with the PS comparing additional factors discovered and deemed relevant during further language investigation.
The following comparison table has been created with these arguments and factors in mind. The table will be used to compare the languages providing a clearer (*more visual*) understanding of each:

	LabVIEW	MATLAB
Access of external data source:	* * * *	* * * * *
Data plotting:	* * * *	* * * * *
Visual component availability:	* * *	* * * *
Component extensibility:	* *	* * *
Rendering engine:	* * *	* * * *
Measurement and calculation engine:	* * *	* * *
Numerical analysis:	* * * * *	* * * * *
OO support and handling:	* * *	* * *
Skinning and styling:	*	*
Data loading and handling:	* * * *	* * * *
Code interaction:	*	* * *
Flexibility and adaptability:	*	* * *
DAQ support:	* * * * *	* * * * *
Cost:	* * *	* *
Available support:	* * * * *	* * * * *
Overall Average:	*** * ***	*** * * ***

Both languages can connect to different types of external sources allowing external data to be “read” into a system the data being presented in a visual manner that can be used for analytical purposes, thus both languages meeting the “basic” PS requirements.
As seen in the above comparison table, both systems are reasonably strong with (*surprisingly*) MATLAB coming out as the better and more relevant language for use with the PS. LabVIEW appears to be the most popular and is the more commonly used language of the two, and it was therefore assumed would be better (*most suited to the PS*) but after further investigation it appears that MATLAB is the preferred/better language for use.
Both allow the ability to plot and render acquired data (*having strong graphical rendering engines*) onto provided (*built in*) visual components, including various charts, graphs and other types of interesting graphical representations, with MATLAB offering more powerful and detailed/advance components.
A powerful and advanced measurement and calculation engine is built into each, allowing calculations to be performed on large amounts of data with advanced numerical analysis solutions and accurate data plotting. They allow the generation and organisation (*controlling*) of large data sets and DP's for visual components and other similar (discussed) analytical components.
Both languages are aimed at developing applications of a data visualisation nature, and can therefore handle large amounts of data using various built in mechanisms such as data caching, quantising and filtering methods and calculations, all allowing for accurate and efficient data plotting and rendering.
The the programming style used is of a data flow nature, both supporting an OOP approach, treating each item that is created and added to an applications data flow diagram as an actual object each item being separated (*effectively having its own class*) extending other relative components and/or the application source further including other objects represented through item linking within the data flow diagram.
The support of OOP allows applications to be easily built (*extended*) upon, improving the effectiveness of future upgrades and creating a more structured and Organised OO application framework further improving performance and maintainability factors.
Although both languages support OO development patterns, they are limited in regard to code interaction and as a result both application and component flexibility and adaptability. Between the two, LabVIEW is affected the most as application development is tightly limited and controlled by configuration dialogues. This limits the programmer to only what is available within these dialogues, and in most cases these dialogues are very advanced. Also the lack of support for actual “hard” code interaction and adaption is believed (*at a personal* level) to be a great restriction, as within application development, code is both the heart and structure of an application. For “real” adaptability and “real”

control, permissions to access and modify (*work with*) code is an important requirement. MATLAB offers more code interaction, and is therefore believed (*for these reasons*) to offer more to the PS. However it is still limited as the code modifications are restricted to built in MATLAB components, preventing the creation of programmer/application specific components and therefore not being of much benefit.
The discussed code interaction limitations further affect application styling and skinning. Basic application styles can be controlled within both languages including sizes and colours, however as an application is limited to components only supported by the particular language, styling is also restricted to the application framework with selected components being used. Again MATLAB offers more in terms of application aesthetics, however, only because it offers more components and application options from which to choose.
Lacking from both is the ability to create custom components and skins that can be applied to any component, container or any other application object.
Both languages do however support and excel in the complete processing of the DAQ life-cycle, in terms of connecting to, or reviving data from a DAQ device (d*epending on the system setup*) handling, calculating and processing the data, further rendering the data relevantly at the front end in a very fast and efficient (*easy to control*) manner.
Both LabVIEW and MATLAB are expensive products (*costing over the 1000$ price mark*), however both having cheaper student editions (*not allowed for production use*) available.
Although expensive, both products provide a large amount of related support and documentation, thus developers theoretically should not be "lost" during application development.
LabVIEW is the slightly more expensive of the two, thus from completed investigation and comparison, it is obvious that for the PS, if such a DFP languages is to be used, MATLAB is better in terms of "value for money".

Overall powerful applications can be built using the discussed languages, in a fast and easy method (*even for a novice programmer*), they would be well structured incorporating OO design patterns. They would perform well, be reliable and user friendly. They are aesthetically pleasing and result in very effective applications.
In relation to the PS a DFP approach could be adopted and it is believed that a successful application would be produced meeting all its requirements, however, at a personal level, it is believed that for the best result and (rich) user interaction, components should be tailored to fit specific applications. This will allow the ability to create new, and extend existing components and classes, and furthermore the ability to customise themes, skins and styles of applications. It will also allow flexible and custom style sheet (CSS) creation and/or interaction, both of which are lacking within the discussed DFP application development.
As discussed, both products are not cheap, although open source alternatives exist including "FreeMat", "Mathnium", "GNU Octave", "R", "S-Lang", and "Scilab", that could be a more suited option if such a programming route is to be taken for the PS development.

Relative to the PS both products have been criticised and limitations within each have been found. This does not mean that the discussed areas are problematic areas for all applications, as there are many popular, powerful and successful applications (*even applications similar to that of the PS*) existing and in regular use today.
The most advantageous factor of both (*and other similar*) products is that they allow for the creation of powerful and effective data driven applications in a fast and efficient manner. They do not require advanced programmer knowledge and understanding and allow a novice developer to design and create a suitable application, thus saving on related development costs.

Application development using such frameworks and languages is restricted due to the discussed limitations. In order to obtain a better understanding of such limitations, and to uncover whether they are as restrictive as investigation has proven, and if there is no way around such restrictions, LabVIEW has been used and tested. A small "test" application has been produced in order to get a clearer idea and understanding of such programs helping to make a better and more accurate selection for use with the PS.
This discussed LabView application development process is described in **[Appendix 8]-(Simple LabVIEW Application Development Walkthrough).**
[Appendix 8], highlights and justifies both investigation and personal thought regarding the limitations of a data flow development cycle and eliminates such a framework and programming approach for use with the PS.

5.5.3.4 - Further Language Selection

Due to the discussed concerns regarding DFP methods and limitations, a choice has been made not to take this route for the development of the PS. Although due to the nature of the application, the use of either a specific data flow or numerical based language is considered as the "natural" and even the expected route. Most other similar (*and reviewed*) applications and systems take such routes, therefore an obvious "risk" is being taken in regards to the PS, as a different route will be persued.

Languages such as C, C++ or Java could be used, and of the three, Java is preferred. This is mainly for personal reasons and a greater knowledge and experience of the language over others, supporting all of the PS development requirements including OOP, custom and extendable classes and overall application control at a code level. Due to the low-level code based development of such languages, the creation of an interactive and "rich" user interface consisting of dynamic and animated components is (*although very possible*) a difficult and long winded process, as such languages are aimed at back-end system developing thus offering a limited (*and in some cases no*) graphical rendering engine.

A solution and preferred result (*final effect*) in terms of look, feel and richness (*engaging interactive user experiences*) is that of an Adobe Flash application, thus a ".swf" file that is run within the "free" downloadable Adobe Flash Player (AFP) to render its output.
Flash applications are generally used for web based purposes such as websites, banners, advertisements, or any other interactive components, and are in most cases reasonably small in size (*both physical and scaled size*) and do not handle large amounts of data (*in some cases may use a simple XML data provider or application descriptor file*).
Due to known issues related to the AFP in terms of memory handling and management (*leaks and poor garbage collection*) the use of Flash applications has been limited to smaller "components" as opposed to full scale applications. Such AFP problems have been and are being resolved as new players are released, with the new AFP 10 (*currently an Alpha release*) being far more advanced than its predecessors, allowing for more powerful AFP applications to blossom.

More detailed information regarding the Adobe Flex Player, its uses and possibilities including examples can be found at **[81]-(Adobe Flash Player, 2009)**.

AFP applications are usually and most commonly developed using Adobe Flash software, which is an advance visual authoring tool and environment that is programmed based upon a TimeLine.
Within Flash, due to the timeline based structure, applications and components are programmed at a "time" synchronised level, which has both strengths and weaknesses depending on the type of application required. For example, for small components or adverts including many animations, transitions and object tweening or any applications that rely heavily on the need for time synchronisation, the used of Flash to develop such applications or components is an ideal solution. However, for the control of more advanced, complex and larger applications the timeline is not the best solution, and can very quickly become very large, over complex and thus out of (*easy to handle*) control.
All Flash based applications are translated into and compiled as **A**ction-**S**cript (AS), an advanced scripting language from Adobe. Flash applications can also be programmed in pure AS thus eliminating the use of the timeline (*if and where necessary*). AS2 is a structured base scripting language, and has been for a while the most commonly used and popular version of AS (*and still is in many cases*). It has proved very successful as a scripting language for the use and control of smaller applications and components, however AS2 is not OO and thus proves problematic in terms of reusability and larger application development. This problem has been recently resolved with the release of an OO version of AS, AS3 changing how AS and AFP applications behave and perform. This offers a lot more in terms of power, flexibility, expandability and control, and results in more advanced application possibilities.
Even though when programming applications are very powerful using AS3, within the Flash environment this can become very complicated to control and understand as the development environment (*framework*) is built upon and around application development of a visual manner with the incorporation of the timeline.

More detailed information regarding the Adobe Flash including technical specifications, reviews, versions and examples can be found at **[82]-(Adobe Flash Professional, 2009)**.

The power of AS3 and the limitations of Flash in relation to programming advanced AS3 applications was a known issue within Adobe, and their solution to this problem was the creation and release of **Adobe Flex** (*Flex).* Flex, now in version 3, is a very powerful development tool for AFP applications.
The Flex framework is built upon the very successful and popular Eclipse framework that is used within many other successful application development environments (*mainly Java based*) offering the creation

and management of AS3 projects and libraries enabling an advanced structured OOP based platform for AS and AFP application development.

5.5.3.4.1 - Adobe Flex

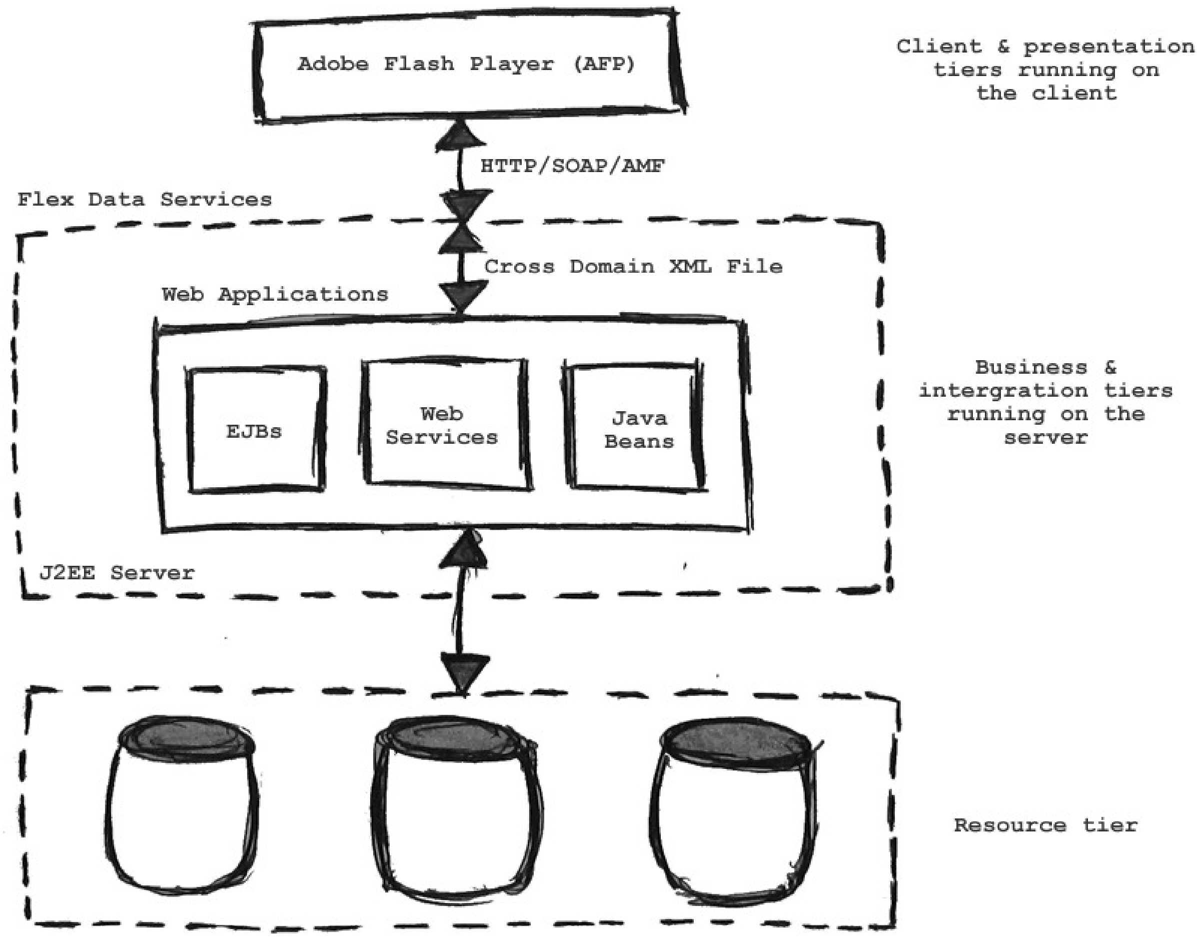
A (*additional*) reason that Flex has been selected for investigation is from personal experience since it is used on a daily basis and much knowledge has been gained in and around the technology (*compared with other programs, languages and technologies*) which is additionally believed to benefit the development of the PS in terms of speed, principles and best practice.

The fundamental concept of the Flex framework is to build applications that are compiled in AS3 and built into a ".swf" output file, further being rendered/ran in the AFP, thus being an advanced and powerful design paradigm compiling AS3 code into a ".swf" file that can be run almost anywhere (*on the desktop and web*).
Flex is the same as Flash in the concept that an application is programmed and compiled in AS of which a ".swf" (*application)* file is built. However, the way in which the applications are developed is completely different, and this is where the advantages of Flex (*in relation to the PS*) start.

There has been for many years a distinction between designers and developers, with designers tending to lean towards software such as Flash this being a more visual/timeline approach to application programing, and developers leaning towards more OOP based environments such as Java. Flex however caters for all types of user (*both designer and developers*).

Adobes aim when developing Flex was to create a paradigm enabling developers that are used to general OOP related paradigms and development methodologies (*such as Java, C, C++, Ruby and Python*) to learn and pick up the rudiments in a fast and easy manner, thus enabling them to adapt and program Flex applications that are rendered in the AFP.
Flex includes a built in and advanced compiler, debugger, and supports classes, components and libraries thus making for a very flexible and powerful framework.
All Flex code is compiled as AS, however it does not have to always be programmed using AS. for developers who are not so experienced with OOP principles and/or don't like getting their "hands dirty" with advanced low-level scripting. Flex uses an XML based tags notation (MXML) allowing for declarative markup supporting any built in (*or custom*) classes and components, making for a more simple, structured and easy to understand tag based coding standard to create Flex applications. All MXML code is built upon AS, and is thus translated back into AS when an Flex application is compiled. AS is however more advanced and has more application control, offering a lot of (*almost*) endless possibilities to an advance AS programmer when creating Flex applications. AS is, as its name suggests, a scripting language built from (*like JavaScript*) ECMAScript notations supporting the latest EX4 (ECMA) XML standard. This offers a lot of possibilities and power to XML descriptive applications.
The true advantage of Flex is that very rich and interactive applications can be created very easiliy with very little coding involved through the use of MXML. More advance AS applications require more time, but again in terms of time verses results, it is a very impressive technology.
As standard, Flex includes many advanced, interesting and useful components that can simply be included and used within an application with little effort. Such components consist of **controls** (*buttons, check boxes, data grids, sliders, combo boxes, steppers, labels, text inputs progress, bars ...*), **containers** (*boxes, panels, control bars, menus, forms, grids ...*), **navigators** (*accordions, tab bars, link bars, toggle bars, view stacks ...*) and even **charts** (*area charts, bar charts, pie charts, line charts, plot charts ...*). All of these components can be added and controlled at a basic level using both MXML and AS, however they can be further extended and built upon using AS, and even new "custom" components (*classes*) can be created using AS and again included using MXML.
Another advantage of Flex is that it is compatible with all SVN (*version control*) clients offering built in SVN management and HTTP Servers, as well as with any other server side scripting/programming languages. PHP, ASP.NET and JSP scripts as well as any servlets can be used as a type of back-end control, for example, allowing DB communication or interaction with any other external source and thus allowing Flex to program and handle the client, connecting the two with technologies such as both Client Side (*JSON, XML, RSS, SOAP, ATOM, AMF ...*) and Server Side (*HTTP, HTTPS, RTMP, Raw Sockets ...*), again opening up many options and possibilities for such applications.

The main aim of Flex was to enable the development of rich client side applications that run over the Internet for distribution through the AFP. The aim was not to replace Flash, but to focus on the development of more structured and powerful OOP related applications by providing a specialised framework rather than the development of the discussed banners advertisements and server side components (*leaving these for for Flash to handle using a more relevant timeline based approach*).



[Figure 36]-(Basic Flex Three tier Data Access Platform)

As shown in **[Figure 36]** the basic data access platform that Adobe have developed and that Flex follows for the development of **R**ich **I**nternet **A**pplications (RIA) can be separated into three tiers:

- Resource (Server Side) Tier
- Integration Tier
- Presentation (Client Side) Tier

The process builds (*concludes*) from a server side resource tier consisting of back-end (*server side*) control programming approaches (*such as ASP.NET, PHP, JAVA Python... and/or relevant web services or even built in Adobe Services*) used to interact with data sources further generating and supplying data fed from the server to the application (*known as an integration tier running on the server*). The data is sent (*transmitted*) to the client side application where the data is handled and the application is compiled and built into a ".swf" AFP file (*known as the Client and presentation tiers running on the client*).
The client side tier can be separated into different areas including the relevant OS used (*Windows, MoacOS, Linux ...*), the selected runtime environment (*AFP or Adobe Air or even both*), the Client framework (*Adobe Flex*) and the further content and applications used.
A more detailed technology platform can be found in **[Appendix 9]-(Adobe Technology Platform for RIA's).**

[83]-(T Patrick, 2008) describes that with the current supported release AFP 9 is "*... is installed on 95% of computers that use the Internet and has been installed more than 3,500,000,000 times at a rate of over 10,000,000 times per day ...*" developers want applications to "*... run compatibly in as many places as possible. AFP provides a solid foundation upon which to build these experiences without the need to install or updata anything on the end users computer ...*". **[83]-(T Patrick, 2008)** further provides an example of an online store application, stating the obvious, that the owner of the application (*store*) wants to have as many available customers as possible, thus due to the above figures, the AFP would

be a perfect solution in terms of platform and customer availability (*apart from standard HTML*). Additionally it will provide a great "rich" experience easily overpowering other solutions to 95% of the web. Most standard web-browsers (including Internet Explorer, Mozilla FireFox and Safari) either have the AFP installed by default or have a built in option and plugin for the player, as available on most up-to-date machines and browsers.
If a selected technology used to distribute applications and software in general is not widely and easily available (*not installed or is incompatible*) customers and potential users of the software are (*in many cases*) immediately being lost. This is due to the wide, easy and free availability of the AFP and it is believed to be a relevant and promising solution for the application distributor in relation to the PS.

Another great advantage of the Flex framework (*especially in relation to the PS*) is that it is not limited only to the web browser, and can also be run on the Desktop, opening further and obvious advantageous areas for applications. Adobe developed a new runtime called Adobe AIR, which is a runtime that allows applications to be developed and distributed to the Desktop (*for Windows, Mac and Linux **O**perating **S**ystems (OS)*) all using Flex and running from the same source code. This allows applications not only to be run in a web browser, but also a "true desktop application" allowing many more possibilities removing restrictions towards web based programming and applications.
Adobe AIR is again (*like the AFP*) a free download incorporating the AFP development specifically for desktop distribution. Air has its own built in installer, therefore applications can be obtained (*downloaded*) and simply installed with the Air installer handling all the required (*packaged*) files installing and organising them to a users desktop. "*Air applications are real native desktop applications and provide APII's for writing files, drag-drop, system notifications, network detection and more. AIR empowers developers to write desktop software that leverages the Internet*" **[83]-(T Patrick, 2008).**

Flex can even be distributed to mobile devices, although not relevant for further investigation in relation to the PS. This however is an interesting concept and an additional area for project expansion.

A concern regarding the distribution to the AFP is related to the data quantity, handling and rendering of the AFP. As discussed the PS will acquire a lot of data (*up to thousands of rows*) at any one time from a motorcycle as it travels around a racetrack. Due to personal experiences regarding memory handling and management and poor GC. such large amounts of data can be both very CPU and memory intensive, stacking up and using large amounts of system memory in a small amount of time. If the AFP cannot handle this it could have dramatic results (*either crashing the application itself and/or the system that the application is running on*). Justifying this worry is the lack of existing applications handling such large amounts of data within the AFP, thus assuming that developers have chosen to take other routes when creating data intensive applications. However, specification wise (*as documented by Adobe*) Flex should in theory be capable of handling (*and has the potential to handle*) such data quantities (*if used and handled correctly*).

Due to many reasons including the possibilities and rich interactive results that can be achieved using Flex, as well as existing experience and knowledge in and around the technology, it is assumed to both speed up and improve development, resulting in a better documented, programmed and stable format. Flex will be the chosen technology for the development of the UI application of the PS.

There are, as described concerns related to the capabilities of the framework and AFP that could hinder and even stop the development process, however, an attitude will be adopted in that nothing is impossible, and again a further "risk" will be taken, dealing with such areas as and when (*if*) they appear.
If the development process is successful using Flex and the expected outcomes are met, it is believed that the final application result will (*should*) be very impressive in terms of appearance (*aesthetics*) and overall interactivity and richness of content.

5.5.4 - Research and Investigation Conclusion

The construction and writing of this chapter has evolved into a very in depth and extensive study covering all the required areas believed necessary to enable a successful project solution and development.

The knowledge gained during this chapter (*the knowledge and understanding level achieved at the end of chapter in comparison to the level at the beginning of the chapter*) is great and detailed. It is considered to be very valuable, changing initial thoughts, opinions and directions of how the project would progress. It opens up and identifies additional problematic areas that need to be considered, finding solutions and re-visiting areas that were not completely understood before this chapter was commenced. The chapter has not only been a great learning process, but has additionally aided further PS development to a higher level, which it is believed will make for a much smoother and stable process.

The main project requirements have been taken, and all possibilities that are believed relevant in terms of enabling a successful project completion, along with areas that are believed to improve project development (*even if not directly associated/required by the system itself*) have been investigated in great depth and detail. The chapter has additionally taken and expanded upon sources that were identified within the LR, that were highlighted to be of be interest and relevant to the PS. These were postponed for further investigation within this chapter, and have been investigated to what is believed to be a high standard. Additional areas and existing systems have also been identified that are deemed to be both relevant and beneficial to the project.

The completion of this chapter has made clear what is and what is not possible in relation to the development of the PS. It highlights both advantageous and problematic areas that may (*or may not*) exist during development, furthermore identifying methods and solutions to appropriately handle such areas.

A solution has been identified along with relevant methods required to achieve this solution (*which will be discussed in more detail the next chapter*). In certain ways the identified solution although similar to what was initially thought/planned, is also very different due to the areas identified within this chapter.

As discussed, it is believed that all areas have been covered/investigated, additionally identifying new areas that have further been addressed, with a suitable solution being identified.

The basics of Location tracking systems have been covered, identifying the different types of relevant location based systems that could be used for the PS further investigating and comparing existing systems, architectures and technologies within such systems. A great level of knowledge and understanding of each was gained finally selecting a method of mobile object tracking that is most suited to the PS.

Not only acquisition methods of locational position coordinates have been investigated, but also the calculations and mathematical foundations required to obtain such measurements. Although not required for project completion, it was believed (both at the time of investigation, and still now) to be relevant to understand, the fundamental workings of such systems and additionally open and identify areas to be aware of as well as areas that can be further taken advantage of, thus improving accuracy and overall system performance.

Much has been learned and taken from individual and existing system investigation, which is thought to be one of the most valuable areas (*in terms of a learning process and aid to the PS development*) when looking into the direction that such systems have taken and understanding the reasons for these directions. Each system was compared to others and to the PS. This further helped to identify the best routes and practices to be applied and looking at ways which would benefit the PS. During system investigation areas were found that are not requirements of the PS but could be used for the benefit of the PS. Of these discovered (additional) areas, depending on the beneficial level and required implementation times, some will be added to the system requirements and implemented (*even at the prototypes stage*) with the others being relevantly documented and left as further project expansion possibilities.

Different Hardware and DAQ devices have been selected that are deemed relevant for use with the PS and investigated further to both try to obtain a better understanding of the role of such devices, and also to try to find a suitable device that could be used with the PS. The most relevant devices were selected from a long list, and were investigated in detail highlighting the strengths and weaknesses of each. Along with technical specifications and beneficial areas these were further compared one against the other and the PS, the conclusion being an identification of the relevant device for use.

An important requirement documented in the Project Proposal is that of the lean angle which when included in the system is believed to take the PS to an additional level due to its “motorcycle” nature making the analytical process more valuable, reliable and worthwhile with such (*motorcycle lean angle*) measurements included in the analysis process. For this reason lean angle acquisition techniques and devices have been investigated to a decent level, with relevant possibilities that can be adopted to the PS being identified. This further enabled the analysis of such measurements which met the project requirements and removed any initially based doubts.

Finally investigation into application development tools was completed identifying relevant

programming approaches, languages and frameworks that could be used to create DAQ data based (*driven*) analytical ASW relevant to that required by the PS. Again much was learned form this section of the chapter in regards to programming such SW, as at a personal level, DAQ ASW has never been completed or even attempted. Therefore it is a new and interesting area.
Surprisingly due to personal experiences and knowledge within other areas, the standard and documented "best routes" to take when developing such SW are not being followed for the PS development. A risk will be taken and a route that is more believed in, and understood will be used. Therefore it will be interesting to see the results of the PS application in comparison to existing applications created using a more standard "better suited" approach to data based DAQ programming, identifying the beneficial and limited areas of each.

This chapter has proved, comprehensive, detailed and very valuable for identifying new and interesting areas that will benefit the PS development. IT also identified problematic areas that may affect the PS development with additional solutions and methods being established.
The chapter has also proved very valuable in terms of a learning process, since much has been learned establishing a vast amount of additional knowledge and understanding both in and around the subject area.
All areas required to complete the next part of this report (*the Solution, Design and Implementation chapter*) along with the rest of the project development and completion cycle have been identified and understood through the investigation completed within this chapter.

6 - Artifact / Solution Chapter

This chapter will review and reflect previous work completed within both the LR and R&I chapters, highlighting valuable sources discovered, further discussing relevant and beneficial that helped to identify a potential solution, further highlighting sources and technological areas used, and the paths chosen to achieve the final result.
The selected approach (*solution*) for the creation of the PS artifact will be discussed highlighting the reasons why it was chosen and how it could be implemented.
As discussed, this project is separated into two main areas consisting of a "back-end" and "front-end". The back-end system cannot be implemented due to financial and resource restrictions, however a solution for such a back-end system has been identified and will be documented accordingly within this chapter.
The front-end of the PS consists of an application relying on data being provided from the back-end system. The selection development solution (*method*) of this application will additionally be discussed within this chapter, being further followed, with an implementation process will taking place and documented relevantly.

Before the artifact is created it will be clearly defined in relation to what it is, and what it achievements are, consisting of an in depth design process hilighting what the application will do, its overall goals and requirements, and how it will meet these requirements, thus giving scope of its complete coverage.
Once the design has been established, the (*front-end*) artifact will be implemented with these implementation stages being documented at each step and sub-steps accordingly.

Overall, the methods in identifying and accomplishing the PS solution will be detailed, including the steps and directions taken to both realise and achieve the solution, from the investigation and reasonings for using specific technologies, hardware and languages through the complete design and implementation phases, finishing with the testing of the final artifact, with additionally any problematic experienced being documented.
Although there is a high level of confidence in the realised solution, problems are expected during further phases, and these areas will be documented accordingly within this chapter.

Once the artifact has been created, a testing process will take place. The results from this testing process will be discussed in relation to "what" has been created and if it meets the initial PS retirements and expectations, thus justifying the artifact against the projects aims and objectives, further proving if the pPS meets its expectations, and if not, why not.

6.1.1 - Previous Work Completed

The completion of the LR and Research and investigation chapters has established a solution to the PS that can be "theoretically" used and implemented meeting the PS requirements.
The initial requirements and project goals were specified and have been further documented and discussed throughout the report in detail, resulting in the understanding of a theoretical project solution. This solution is required to be translated and implemented into a "real" artifact that can be further tested and evaluated accordingly.

As discussed, an extensive and in depth process (*review, research and investigation*) has been completed to enable this solution to exist. This process commences with large literature search in which the most relevant results, sources and projects discovered were selected and further reviewed and discussed (*in detail*) within the LR chapter.

6.1.1.1 - Literature Review Summary

Different literatures, systems and various other sources were selected for review, focussing on areas relevant to the PS, comparing and contrasting each, highlighting potential possibilities and sources in relation to current approaches and possible project expansions, further identifying areas that are both required, and that could be used (*adopted*) within the PS, thus overall improving the projects development and final outcome.

The first area covered (deemed necessary) within review was different DAQ methods, technologies and architectures relevant to the PS, with additional literatures and sources regarding relevant DAQ SW also being reviewed, consisting of existing systems, and selected design and development methods of such SW, including both DSW and ASW approaches.

Due to the mobile nature of the System (LBS), sources regarding different and existing WSN's and even NGWN based solutions were covered, along with the available architectures (*both passive and active*) available within such networks in relation to how the PS could both use and benefit from such networks and architectures.

During the LR it soon became clear that GPS is the main concept (*strongest and most popular*) used for mobile object tracking, with most of the reviewed sources both focussing on, and discussing the use of GPS within such areas, and how the technology is successfully used (*and can be used*) to track mobile objects in a variety of different environments. Therefore many GPS literatures, related to the technology itself (*its foundations and fundamentals*) and how it is used within relevant tracking systems were reviewed. This enabled a great insight into the technology, offering examples of how it is, and could be used with, and to benefit the PS.

The PS tracking possibilities were not “wanted” to be limited to a GPS based solution, as different methods were desired to be compared and contrasted, thus identifying the best (*possible*) suited technology for use.
A more in depth literature search was performed and other relevant LBS methods were discovered. The methods were further rewind, gaining an advanced insight into how they are used, and for what reasons, enabling a platform for comparison between the identified (*and different*) LBS systems and technologies.

Many literatures were discovered regarding existing systems, in relation to the development phases of such systems, and the purpose(s) of the systems. This proved very valuable during LR, identifying key areas (*both advantageous and problematic*) that should be acknowledged and handled during the development of such a “mobile” tracking system.
Not only literatures regarding similar systems (*motor racing related stems*) were reviewed, but LBS in general, as a broader understanding of such system foundations was believed necessary to obtain (*outside the subject box*) not only **in** the direct fields, but also **around** these field(s).

Authors opinions and arguments have been obtained and compared against both one-another and personal opinions and arguments, helping to provide a more accurate conclusion, further allowing more reliable (*supported*) personal opinions, furthermore providing for a more reliable understanding and selection of directions and domains (*technologies, architectures, SW and HW...*) to be taken for further (*more in depth*) research and investigation.

At the end of the LR chapter, the opinion was obtained that the best method for the development and completion of the PS, was to follow and create a custom “hybrid” solution, taking preferred (*most relevant*) areas from different domains and “mixing and matching” these areas to build an overall and complete system solution; **“***Based on the reviewed system and literatures “mixing-and-matching” advantageous areas related to the PS thus forming a hybrid styled foundation that is believed to be a sufficient and relevant direction to pursue for further research and development*”.

A lot has been covered within the LR chapter allowing plenty of scope for additional research along the development cycle of the PS. The literatures studied have provided a great advantage and “body of knowledge” regarding possible uses of varying technologies, additionally justifying these solutions with other authors and schools of thought regarding the subject area.

Many research inputs, points and domains were identified and highlighted within the LR in which it is believed relevant to obtain a better/more in-depth knowledge and understanding to determine a more successful project completion.
Additional areas an possible investigation leads have been opened within the LR chapter that should be followed up and closed (*tied up*). This required further investigation into these domains to enable the selection of a single domain. Therefore these areas (*along with others*) were "passed" into the next chapter and highlighted for further investigation.

6.1.1.2 - Research Approach

It has been established in relation to the research and development of the PS, no "particular" methodology will be followed, as due to the discussed development situation, a more relaxed, free and less strict and structured working process/environment is both preferred and believed to be more productive. Therefore a hybrid methodology will be followed taking relevant and necessary areas from each selected and other existing methodologies, glueing them together allowing a general and consistent structure to be maintained, further ensuring a successful result, at the same time remaining comfortable, following a (*flexible*) structure that is preferred and believed best suited for a successful project completion.
The research approach and directions taken have been documented and justified within the R&I chapter, and this process will continue within the following design and implementation sections.

6.1.1.3 - Research and Investigation Summary

The identified "key" areas from both the PP and LR, have been taken and used as a template/guideline for the R&I chapter. Due to both the amount of identified investigation domains, and the additional amount of relevant areas and sub-areas identified within the R&I, has again made for another very extensive chapter.
The aim of the chapter is to identify solutions that can be used to enable a successful project completion. The potential system solutions will be investigated further within the chapter, with additional domain comparisons being completed enabling strengths and weaknesses of each to be identified, in relation to meeting the PS requirements to the highest and most efficient standards possible.

The chapter commences with the large domain of "mobile location tracking methods" regarding the fundamental mathematics behind such systems, driving and providing location based measurements (*calculations*) and the layers (*levels*) required within a systems architecture in relation to the acquisition of such data.
The understanding of this domain is not required to enable a successful project completion, however, it is believed to provide a better understanding of the underlaying methods used by such technologies and devices that the PS will use and adopt, and with this additionally gained knowledge, further opinions, views and decisions made throughout the project may be affected and even changed (*hopefully for the better*).

The chapter continues to investigate the possibilities (*solutions*) available in regards to general mobile object tracking solutions and practices.
Both SBS and GPS based systems have been investigated and compared in relation to the PS, including examples of existing successful systems within each domain, and additionally the requirements, possibilities, expectations and limitations of each to the high and in depth level.
GPS proved the most beneficial and advantageous technology relevant to the PS, and was therefore taken and investigated at a more "low-level" and in depth standard, highlighting areas of the technology and preparing (*identifying solutions and best practices*) for any problematic areas that could arise.
The RaceFX literature and system discovered in the LR and considered to be a both very relevant and beneficial resource in regards to the PS development, (*being referenced and referred to throughout the project*), was investigated in detail as it also uses a GPS concept, although to meet slightly different requirements. The investigation of this system (*along with many other systems*) proved very beneficial, identifying many areas that can be taken advantage of and avoided as an where possible, thus aiding for a more successful system result.

Once a stable system foundation was established, the different types of available hardware relevant to,

and that could be used with the PS were investigated.
Many different DAQ HW (*receivers, loggers and sensors*) and technologies exist for many and varying purposes. More relevant devices (designed *towards the acquisition of vehicle measurements*) were reviewed with the favoured and most relevant devices being identified and selected for further investigation.
Each device has been investigated to a high level identifying strengths an weaknesses of each providing a reelable foundation for comparisons, further allowing for the most relevant device to be selected and used with the PS. This investigation process is believed to achieve not only be the most reelable and accurate results, but additionally the most reasonable in regards to the systems purpose and nature, thus many other factors are required to be (*and have been*) taken into consideration when selecting such devices and sensors.

A project requirement documented early in the project, believed as a very valuable measurement to obtain in relation to the PS's success and overall usability and practicality, is that of the motorcycles lean angle measurement.
None of the investigated systems or devices “specifically” calculated such measurement. It was therefore deemed relevant to investigate the likeliness of acquiring this measurement in regards to available methods, additional sensors and overall possibility *(complexity, cost and time overheads involved*), establishing if it is a reasonable and possible measurement to obtain, therefore requiring further investigation.

Possible (*theoretical*) solutions to the complete DAQ process have been established within the R&I chapter, therefore achieving the initial documented (back-end system) project requirements.
An additional project requirement is the design and implementation of a prototype (POC) application that can be used to render the acquired data in a way that is can be used for analytical purposes (*demonstrate how data can be represented)*.
The R&I chapter therefore proceeded looking into ASW possibilities (*languages and environments*) that can be used to develop such applications.
Due to additional learning and obtained information regarding the acquisition of such data, new application requirements have been understood, and documented within the chapter, further being used as an investigation and comparison guide of the selected ASW in relation to the PS.
Not only programming languages and environments were investigated, but also programming approaches and methods (*including the obvious data-flow programming approach*) and data storage possibilities, identifying best practices and most efficient methods to develop an accurate and stable application that can be used to analyse such data.
Surprisingly the chosen environment selected for the PS application was of a different nature to the environments both recommended and identified as being most relevant, due to both added application requirements (*rendering and tweening capabilities*) and personal feelings and experiences with the selected development solution. Therefore a “risk” is being taken within the application development, as it is not following a standard “recommended” development route of such a (*data-driven*) application, however, it is a risk believed (*at a personal level*) to aid the PS development process, producing a more impressive result in terms of possibilities and user experiences.

6.1.2 - Identified System Solution

The completion of previous work (LR and R&I chapters) have identified possible solutions to the documented PS requirements enabling a successful “back-end” system and “front-end” application design, and further application implementation.
To recap and help further discussion and understandings within this chapter, the initially set PS requirements and aims and objectives have been summarised re-iterated below:

Data that could be collected and used for relevant measurements, features and analysis may include the following fundamental information:

- Overall racetrack lap times.
- Object positioning at different sections of a racetrack.
- Identification of racing lines used/taken during individual laps.
- Different speeds at different sections of a racetrack.
- Average Speed calculation.
- Fastest and slowest speed detection.
- Individual corner speed detection.
    - Speeds when entering corers.
    - Speeds maintained mid corner.

- Speeds when exiting corners.
- Braking detection, including braking points and strength.
- Acceleration detection, including acceleration points and strength.

Problematic and challenging areas to consider/focus during investigation further design and implementation could include:

- The system will be operational in an outdoor environment.
- The system will (*potentially*) have a centralised architectural structure.
- A high speed object (*motorcycle*) movements are required to be tracked.
- A small lightweight device is required to be installed on the mobile object (*motorcycle*) that will not affect handling or performance in any way.

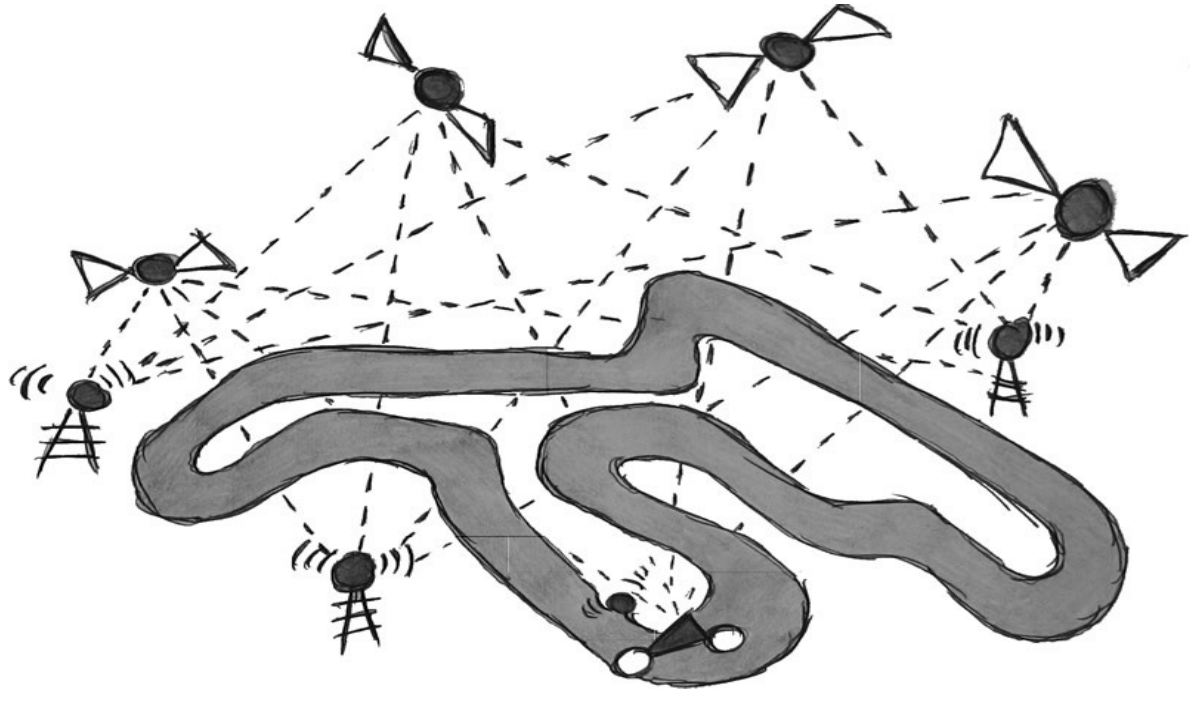
It is believed (*at this stage of the project*) that previous work (*review and investigation*) completed has been successful and to a high standard covering all documented project requirements, as well as many additional project possibilities and directions, to an in depth and detailed level.
Solutions to the documented PS requirements, and thus to the complete project have been identified and can be “theoretically” used to enable a successful project completion. These system solutions will be discussed further within this chapter.

The complete PS will be re-capped (*summarised*) including additional factors and possibilities (*potential uses*) discovered during the previous chapters, as a reminder of purpose and requirements of the system, before “broken-down” individual system solutions are further discussed.

The aim of the PS is to track motorcycles as they travel round a racetrack. Such tracking involves acquisition of not only a motorcycles locational measurements in relation to the racetrack, but also additional measurements including speed, acceleration, braking, RPM, and throttle positions and any other relevant measurements.
Once data has been acquired it is to be translated and rendered as useable and understandable information. An analytics based application (UI) is to be created, that can be used as a learning aid by riders, teachers and even technicians, to help improve a riders and motorcycles performance in relation to a specific racetracks.
The PS application will enable advance learning mechanism for riders in relation to hilighting weak and strong areas at specific times, speeds and locations during a lap, that can with be improved upon or taken advantage of in other areas, further highlighting any motorcycle performance issues (restrictions) in relation to a motorcycles set-up.

Overall the complete PS can be used to reduce lap times, thus making for a very powerful, useful and valuable DAQ system within the motorcycle racing domain, in relation to improving rider skills and motorcycle performance.
The system can be used within learning applications (for new riders), improvement/enhancement applications (for experienced riders) and even within race teams to improve both rider and motorcycle setup/performance in relation to different racetracks within a racing series. The system options are almost endless regarding the uses for such a system within the motorcycle racing domain.

As discussed a potential solution has been found and will be used to develop the system further. As already described, the proposed “back-end” solution will not be implemented at a practical level (*due to costs, time and hardware resource availability*), however the identified back-end solution will be discussed at a theoretical level.
In depth detail will try to be avoided within the “back-end” system solution identification as large amounts of detail into the relevant areas has already been proved, and can be refereed to within previous chapters and the project appendices where relevant.

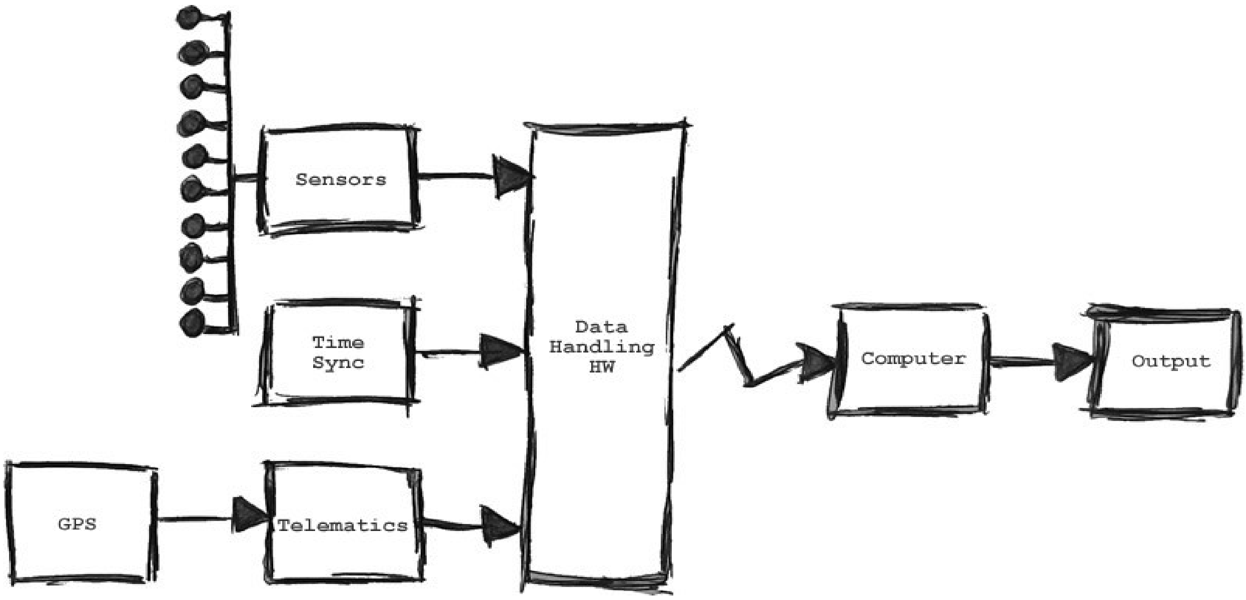


[Figure 37]-(Basic MODAQ System Concept)

[Figure 37] Illustrates the basic PS concept that will be used to achieve and meet the DAQ requirements of the "back-end" system, in relation to the acquisition of relevant data from a fast moving motorcycle within a racing environment (*racetrack*).
A GPS receiver will be installed on the motorcycle, and this receiver will communicate with the GPS satellites orbiting earth. There are in total 24 satellites, and from these satellites a minimum of four are required to obtain and calculate accurate location measurements of a motorcycle at a specific time.
Due to both the high speeds a "tracked" motorcycle could (*potentially*) travel and objects (*buildings, walls and bridges ...*) that could block, reflect or interfere with GPS signals within a racetrack environment, therefore not always being possible to obtain and maintain signals with all required satellites, further DGPS stations and receivers will be positioned relevantly around the racetracks. The DGPS receivers will also communicate with both the GPS satellites and the system enabling more accurate and precise location measurements to be acquired, further improving and even (*in some cases*) eliminating any GPS DOP that may exist.
The DPGS receivers will therefore be positioned at "fixed" locations allowing corrections to be calculated and any offsets removed that may exist between the mobile GPS receiver (*motorcycle*) in relation to the tracking environment (*racetrack*). The DGPS should be additionally positions where a clear line of sight to the sky (*and thus the satellites*) is available, for example on the top of buildings (*grandstands*) or even using specialised (elevated) platforms, where they will be used to transmit continuos differential measurements and corrections.
The number of the DGPS receivers required will depend on the specific racetrack in terms of both size (*area covered*) and the area/environment that the racetrack is located (*the more obstacles and signal interference, the more DGPS receivers required to improve (magnify) the signal accuracy*).

The "back-end" system will be separated into three sub-systems including GPS, telemetry and time synchronization. As discussed and shown in **[Figure 37]**, the motorcycle will be fitted with a GPS receiver (*the receiver used will be discussed further in the chapter*).
The GPS receiver will (*as well as the additional DGPS receivers*) communicate with GPS satellites and in turn calculate (*and correct*) the motorcycles location measurements in relation to distinct time, allowing the location of the motorcycle in relation to a racetrack at and distinct time (*during the tracking process*) to be established.
The acquired locational and other statistical data measurements will be packeted and transferred over a DSL line, and handled by an advance telemetry system, to a CBS where the data is unpacked, synchronised and handled further appropriately.
The PS will include the already discussed DGPS technology which will improve upon standard GPS results providing measurements accurate to within 30 centimeters (*more than accurate enough for use and to meet the PS requirements*).
DGPS will be used along with an advance telemetry system thats job will be to handle the transfer of data from the motorcycle (*both the locational and statistical data*) to the CBS at a transfer rate of

approximately 5 times per second.



[Figure 38]-(MODAQ System Design)

[Figure 38] Illustrates the overall PS design in terms of how the data is acquired and "further handled". The figure shows the the GPS data being conveyed from the receiver (*installed on a motorcycle*) through the telematics system through to the UI (*output*). As discussed the data is compressed into small data packets before transmission, to improve the performance of the transmission. To improve performance further, any required data calculations will be performed after transmission (*when received at the CBS*).
The GPS Receiver will be contained within a specialised DAQ device that is installed on the motorcycle. This DAQ device will acquire additional information from the motorcycle using specialised sensors and engine management communication techniques/ The used of the DAQ device will allow for projects expansions in terms of supporting the acquisition of additional measurements from additional sensors and sources that can be compressed, transferred and synchronised with the existing (*already*) acquired measurements.

The GPS (*locational*) and sensor (*statistical*) measurements are synchronised using a built in "time-synchronisation" layer that is accurate to within one millisecond, providing essential reliable and accurate results that are essential, especially when an object is traveling at such high speeds. The time synchronised data is compressed and transmitted to the CBS where it is handled using specialised data handling SW. The SW translates and structures the the data into relevant information, performing any necessary calculations before it is passed onto a computer (*processor*) for further (*client side*) handling and finally used and rendered within the application UI.

The discussed thematic system consists of TBS positioned around the racetrack with additional telematic transceivers located within the discussed DAQ device installed on the motorcycle.
The telematic system has time multiplexing capabilities (*both analogue and digital*) allowing data to be transmitted from the motorcycle to the CBS in an error free, timely, accurate and predictable manner. To achieve the best results from the system, and to provide continuos coverage between the motorcycle (*telematic transceivers*) and the TBS's, for similar obstruction and distance reasonings as within the DGPS system, the number of TBS installed with depend on the environmental conditions/settings of a racetrack.
Within the MODAQ system, the "advised minimum" amount of TBS's required will be the same as the amount of DGPS receivers in place as conditions will be the same, thus if one system is successful, theoretically so will the other.
The common/standard amount of required TBS within a racetrack environment is estimated to be 3 to 4, with one of these stations being the "master" TBS that will work/communicate with the discussed time synchronisation system to format, synchronise and time slot the acquired data. The correctional offset data calculated by the DGPS receivers is assigned a relevant time slot, and transmitted by the TBS's to the motorcycles DAQ device where the locational measurements are constructed and transmitted along with the statistical data (*acquired from the DAQ device and other onboard sensors*) back via the telemetry channels to the CBS where the data is handled relevantly and any redundant

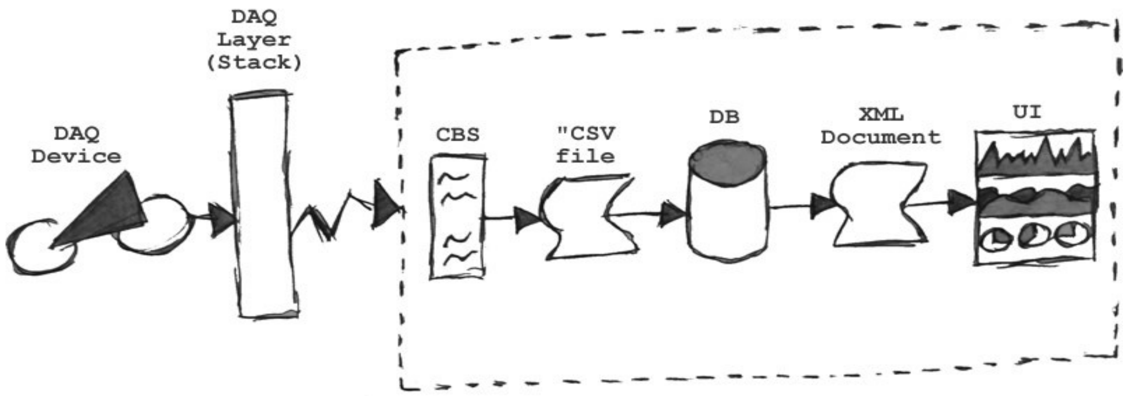
data packets are removed.

The discussed and selected DAQ device that will be used is the "Nology G-Dyno Plus". The Nology device has the built in and discussed hardware allowing such system capabilities and results to be possible, including an onboard accelerometer, telematic transceivers, an advance and powerful GPS receiver (*SirfStar 2 GPS receiver)* and additionally an 11 DAQ sensor input channels, allowing 11 data measurements to be acquired and transmitted through the telemetry sub system, additional to the standard acquired measurements that the devices provides, including speed, distance, horsepower, torque, and G-Forces, thus providing a very useful and powerful data logging system.
The system can measure both the power used in different gears and/or the power and speed/time used throughout the gears, all making for very interesting analysis results. Furthermore, braking times and forces over distance can be recorded with cornering and handling forces all being measured at the same time, thus with some advanced maths can be used to calculate additional factors such as lean angles.
The discussed measurements will be stored in a simple ".csv" format local to the device before transmission. The Nology device has Flash memory allowing up to 128mg of data can be stored locally, translating into up to 12 hours of lap monitoring time (*being more than satisfactory*) before the storage is to be formatted for further use. In most cases the data will be removed from the DAQ device after each "tracking" session as once the data in transmitted, it is stored local to the CBS, and is therefore no longer required on the device itself.
The device additionally uses INS that with the aid of the acquired locational (GPS) measurements can create (*build*) and map an accurate racetrack model, allowing the ability to analyse data in relation to the motorcycles (*visible*) location on the racetrack, again in relation to time.
The system further measures a vehicles altitude, thus depending on the chosen ASW used, and its capabilities, this data (*along with the location information*) can be used to create a 3D racetrack model.

The built in accelerometer allows both accurate power and torque measurements, as well as braking measurements (calculated in a similar way, however using "deceleration" measurements) to be calculated from both already acquired longitude and latitude measurements.
The system can obtain not only the actual power being output from an engine (*acquired form the engine management system*), but also using this data and advanced algorithms with other acquired data (*including that from the accelerometer*) the power that is displaced to a racetracks surface can be obtained, enhancing analysis data highlighting where power is not being used enough, and also being wasted.
Cornering measurements are calculated using the acquired lateral (*sideway*) acceleration data, allowing the G-force data to be collected thus when displayed (*and if displayed correctly*) showing the forces experienced by a vehicle during cornering.

A solution to an "important" project requirement that has been referred to throughout the report, is acquisition of a motorcycles lean angle. The acquired measurements that the Nology device provides is enough to calculate the motorcycles lean angle, and although additional, specialised sensors have been found, and can be used to calculate lean angles, it is believed irrelevant to include additional hardware and thus additional "possible" complications when the measurements required to calculate such angles already exist.

The required is transmitted to the CBS, and the lean angle calculations will be performed client side to save both transmission times and additional "back-end" system work/complexities.
A client side algorithm will be created and performed on the relevant data measurements, using the concept of distance, speed and acceleration to obtain the motorcycles lean angle relevant to each stage/part and time of a lap.
Using the "*... acquired location measurements (-X and -Y coordinates) the exact position of a motorcycle can be known, and in relation to the lean angle, how fast the angle is changing can be calculated. With the angle change speed and time measurements determined, the motorcycles lean angle can relatively easily be calculated. For example a default "start" angle would need to be provided (assumed as zero) and from this start angle, using time and location measurements and the angle change speed, the current lean angle can be calculated*".



[Figure 39]-(MODAQ Data handling process)

[Figure 39] shows the life-cycle of the acquired of the data from the motorcycle (*Nology*) device throughout the relevant stages to the final application UI.
As discussed, the acquired data is stored locally on the DAQ device before it is compressed and transmitted to the CBS. The data travels through and is processed using a DAQ layer (c*onsisting of three sub layers including a reception layer, abstraction layer and synthesis layer*) and is then further transmitted to the CBS where is can be translated into meaningful and understandable information and therefore rendered relevantly within the application UI.
The application UI is the second of the two project stage, thus the next step within the PS solution.

The acquired data is stored in a simple comma-separated ".csv" file format. The selected framework that will be used for the "front-end" application UI development is Adobe Flex.
Flex offers many advantages in relation to application development in terms of its AFP rendering and competent extension capabilities along with supporting the creation of custom components, although Flex does not support "direct" DB communication. A stable, reliable and efficient data handling that the Flex framework does however support is XML document handling (*being both advised for use by Adobe, and being a preferred "personal" format*), therefore an additional "intermediate" stage is required that will convert the ".csv" into a valid XML document.

As the DAQ device cannot store unlimited files locally, and reasons that it is not ideal (*in terms of both practicality and organisation issues*) to store large amounts "individual" files on a computers HD, an additional step has been added to the PS solution consisting of a DB to load the ".csv" files to (*as a local storage and backup*), and additionally the ability to export the data from the DB to an XML document.

An advanced script will be used (*and discussed further within the chapter*) to take the contents form the ".csv" and upload them to a relevant DB (*that can be stored either locally or online depending on the system/user requirements*). The data will be extracted from the DB and exported (*using the same "one process" script*) into a valid XML document that can be further used as a DP for the Flex application (UI).

NOTE (Application Change): The initial requirement of the PS was to create (*design and implement*) a "front-end" application that can be used to analyse the acquired data, however, after additional (and in depth) thought it has been acknowledged that a better (and possible) more rounded system solution is not to provide "just" an application, but a complete application styled framework in relation to the PS fundamentals and the acquired data.

This a big step in comparison to the requirements initially set at the beginning of the project, however is believed to be a step in a more advance and correct direction, that if is successful could change the system as a whole in terms of possibilities, usages and further expansions, overall making for a more powerful and useable final result.

This change will not affect the initial PS application requirements, as an application can still be created, however using the potential and described application framework (*further justifying its power and possibilities*) still meeting the set requirements.
The framework will allow more custom components and even additional applications to be developed (*possibly by third party vendors*) that are application/user specific, thus not limiting a system to particular features or fundamentals. but allowing them to be expanded relevantly into their own domains meeting "specific" and individual application requirements.

The idea of such an application based frameworks came from the fact that, the system (ideas and possibilities) and grown rapidly during the work completed in relation to the initial set PS requirements. A "back-end" DAQ system solution has already been established, along with the described additional data handling layer and conversion mechanism, converting and storing the acquired data into a valid XML document format, therefore constructing a powerful system foundation in which there is no reason why such required mechanism cannot (*if programmed correctly*) be both dynamic and flexible accepting any data input and source, and in turn the creation of dynamic and flexible (*individual*) applications.

Further application (*framework*) development will be affected by this change due to the fact that it is no longer being programmed/tailored towards a specific "single" application, but towards a framework that can support multiple (*relevant*) applications and therefore application requirements.
The system will provide an XML document consisting of DAQ measurements that can be fed (*pushed*) into the framework and used as a specific application DP.
Additional and relevant analytical components can be created and provided as standard within the framework (*graphs, charts, gauges and any other relevant component*) as well as specific custom (*external*) components that can be brought/wired into the framework can be sleeted and modified/customised for use with a particular application.

The owner/developer of a system in regards to the "selected" data acquired from the system, will/should know the types/format of the data (*i.e the data that has been acquired in terms of speed (number), acceleration (number), suspension travel (distance), lean angles (degrees) ...*) and can therefore select (*or provide*) a relevant component(s) supplied by the framework to represent the different data types, thus creating and customising a specific analytical dashboard to suit both individual user/application requirements.

This discussed change should *(in theory*) not be such a complex task (*no more complex than the design and implementation of the initial PS application*), however being a lot more advantageous and powerful opening many additional system possibilities and future expansions, and for this reason is deemed as the direction to take for further application design and implementation.

7 - Application Analysis and Design

The "back-end" system has been documented and a solution has been established meeting the PS requirements. Due to dissuades reasons (*costs, time and resources*), the back-end system will not (*and at this stage cannot*) be implemented, however the steps required to potentially acquire such data from a motorcycle has been investigated and discussed in great detail, and a valid solution has been identified and documented that can be used and referred to throughout further application design stages.

As discussed, The PS is split into two main system areas, the "back-end" DAQ system, and the "front-end" application UI that will be used to render the squired data.
A prototype application can however be implemented, and therefore will be designed within within this section of the report, and further implemented producing an system artifact (POC) that can be used to demonstrate the possibilities of the PS, Illustrating how such acquired data can be handled and represented to a user within a "front-end" application (UI).

The fact that the "back-end" system cannot be implemented, limits the "front-end" development in that no "real" data can be acquired to feed the application with, therefore requiring the use of (*less accurate*) test or simulated data which is at a personal level not preferred, and is further documented within previous chapters as an inaccurate and unpredictable solution to application development and testing.
Luckily for the "front-end" development process, this route does not have to be taken. As discussed, a contact was established within the LR with one of the developers of the RaceFX system (*Ken Milnes*), who has kindly provided a data sample that has acquired using the RaceFX system during a (*competition*) IndyCar race that can be used as a DP during the development and testing phases of the PS application. (*this data will be discussed further within the chapter*).

This section of the chapter will focus on the design of the "front-end" prototype application that will be implemented and used as a "demonstration" (POC) tool for the PS.
Although the application requirements have been documented and are understood, before further application design continues, an application "Mission Statement" will be provided describing the applications purpose, summarising its overall requirements, aims and objectives.
The "Mission Statement" can and will be referred to throughout the application design process as a guide to keep the design heading in the right direction, thus helping for a successful result:

7.1 - Application Mission Statement

The PS "front-end" application will be used to render data that has been acquired using the documented "back-end" system. The application aims to handle the data relevantly, translating it into meaningful and understandable (information) data representations in the form of visual components. Such visual components could be anything from graphs, charts or gauges... (relevant to the data type the component represents) and should be easy to understand, highlighting what happened (in regards to the relevant DAQ measurement), at what time and at what location (in relation to the racetrack).
The application will be aimed for delivery on a standard computer using todays common OS's (Windows, MacOS, Linux...) and should be both easy to install and use (understand).

7.2 - Human Computer Interaction

Human **C**omputer **I**nteraction (HCI) is an important factor in regards to application deign, and therefore relevant HCI issues that affect (*change and improve*) the application design will be taken into consideration throughout the design process.
Due to the importance of HCI in relation to PS application design, it is deemed relevant at this stage to take a step back (*following a more R&I manner*) and investigate relevant HCI issues further that could affect and improve the overall PS application usability.

HCI is a large and very broad subject therefore issues assumed relevant to the development of the PS will be selected for further discussion, looking into different HCI iareas that could be utilised to improve the PS applications usability during the design and development of the application.

Poor application usability can (*and in most cases will*) put users "off" using an application (*drive the users away*), therefore application usability is an important factor that needs to be addressed and taken on-board during the design of the PS application in relation to the applications overall (*and measurable*) success. The main usability goals that HCI aims to improve include (the three E's), Effectiveness, Efficiency and Enjoy-ability when using an application.

To create a usable application, both its context and setting are required to be known and understood (*identified*), thus requiring some relevant questions to be asked before any further design is continued. Such questions include:

- Who will the application be used by (target audience)?
- What tasks will the user be doing?
- What is the users environment like?
- What tools and techniques are available to develop the application?
- What technology is feasible?

The design challenge is knowing what can be done (f*unctionally*), and the manner in which it should be completed to meet users requirements (*usability*). Therefore, the appropriate input and output tools and interaction style should be chosen, the user fit with an application should be investigated (*how the users improvise an application*), how the application should adapt to the user (*customisation*) and how the user adapts to an application (*training / frequent use, ease of learning*). The application should further include the relevant guidance options (*help, error handling, and documentation*) all-aiding in overall usability.

In relation to the PS (*and as described*), the application will follow a design route towards a prototype application, being aimed towards an "alpha" release and thus an "alpha" target audience (*user base*), therefore gaining further user feedback at his "alpha" stage.
This method is preferred for the PS in terms production speed and release, additionally as the application (*in its prototype stage*) will be aimed primarily as a **P**roof **O**f **C**oncept (POC) therefore not requiring so much feedback. If the PS is successful further design will be accomplished and further more in-depth HCI issues and methods will be addressed.

A HCI issue that is however deemed relevant for inclusion within the application design is the use of colour. The colour used should be appropriate and easy to view (*not causing eye strain*). When used appropriately colour can be effective and aid in usability (*for example highlighting or giving certain feelings and effects*). No more than 7 colours should be used within an interface and the use of colour should be consistent throughout. To enhance accessibility colour could be put under users control/preference, therefore the applications skin and colour scheme(s) should be user controllable or at least modifiable.

A common accessibility issue is colour blindness. One in twenty people have a colour deficiency's finding it hard to distinguish between certain colours, therefore a colour scheme(s) that caters for these issues (*allows visibility / readability*) needs to be included.
Colour blindness is an issue that needs to be handled within such an analytical applicant (*like that of the PS application*).
Charts and graphs will consist (*at times multiple*) series each having different colours representing specific data (*Speed, Acceleration, Braking...*) and if these colours can not be distinguished between, the whole analytical process (*and application purpose*) could be dramatically affected, again justifying the need for colour schemes and general colour control to be taken on-boards within application design.

The three main types of colour blindness including, **Deuteranope**, **Protanope** and **Tritanope.** Image examples representing these different colour blindness types can be found in **[Appendix 10]-(Colour Blindness Image Examples).**

Another important HCI issue relevant to the PS application is that of the application navigation design and layout. The navigation should be simplistic and easy to follow. Links and menus should be positioned and grouped accordingly (*usually in a horizontal or vertical manner*) and should be clear and easy to understand.

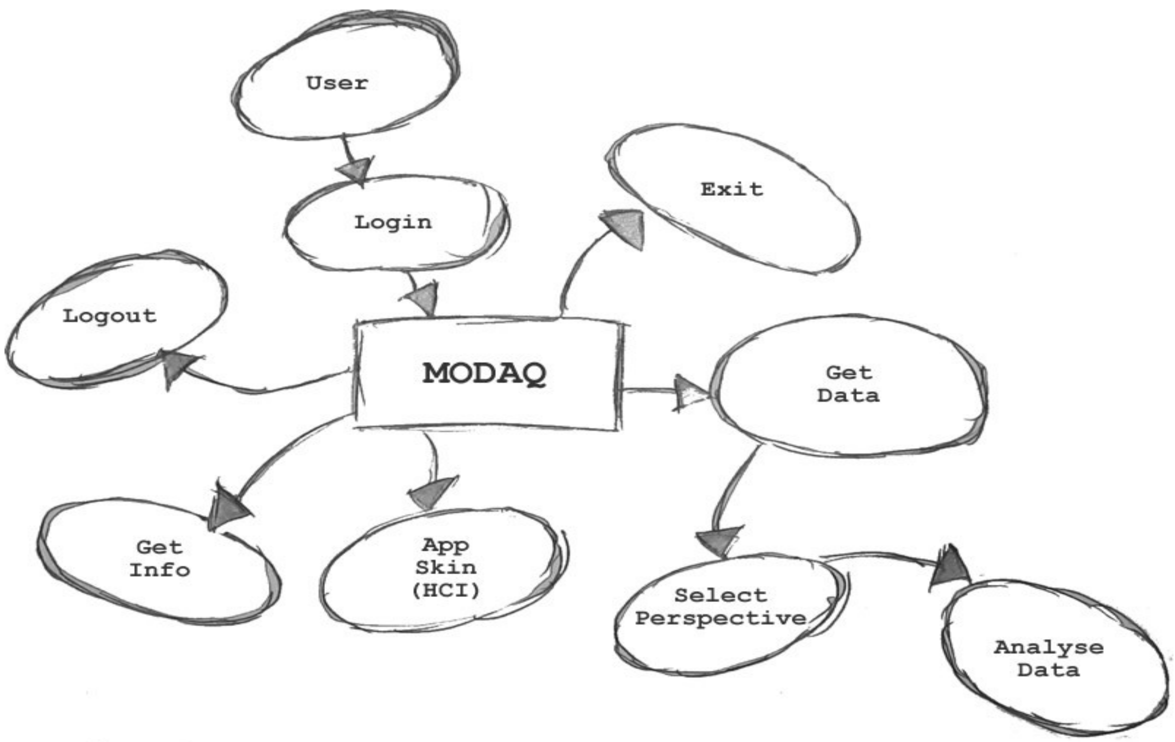
An additional key issue to usability issue is application consistency. The application design should be kept as consistent as possible, therefore as a user "uses" and navigates through an application and becomes familiar with how it works and flows, it becomes easier for them to understand as how they

would execute a task, is how they expect to execute the task, improving speed and efficiency of task completion.
If an application was not consistent it could cause user confusion and even frustration therefore creating poor usability. *"Effective applications are both consistent within themselves and consistent with one another"* **[91]-(Dix et al, 1993)**.

Due to the "POC" nature of the PS application, HCI and usability in general is not considered as a "high priority" system requirement, however, HCI is an area that can "make or break" an application (*even at such a prototype level*) and for this reason is an important area to cover and include within the application design. The discussed HCI issues along with additional relevant HCI related information regarding application design issues can be found in **[Appendix 11]-(Further Human Computer Interface Considerations)**.

The design of the PS application should not be too complicated for two reasons.
The first being that it is only a prototype application, therefore will be be released as an "Alpha" candidate, thus being tested further after release gaining feedback and thus developing the application further in relation to identified application and user requirements after it has bee put into practice.
This is a preferred method of design as it is a lot faster, being more beneficial in relation to beta and final application releases, especially in comparison to applications that are heavily designed to a strict specification before they are even seen/used, thus in many cases (*when unsuccessful*) requiring a complete new design and implementation process, thus wasting both time and money. For this reason, the PS application will follow a more "loose" and flexible design process, highlighting problems and improvements by the people who will be using the SW, thus designing and aiming the application towards both the "alpha" and prototype release.
The second (*and more "actual" design related*) reason is that the PS application is not aimed at a specific user type (*administrator, general user...*) and therefore does not require being designed specifically to meet "different" user requirements, making for a straightforward "single" user based design process, further eliminating multiple user classification requirements, and multiple user class descriptions. However, a user will be using the application for a specific reason, and this user and their "usage" reasons are required to be identified and understood.

7.3 - User Classification



[Figure 40]-(MODAQ Application User Model)

The user based application processes should be (*at this stage*) separated into a number of activities performed by a user. The basic application activities for the PS include:

- Application login
- Application logout
- Exit application
- Skin application (HCI requirement)
- Get application information
- Get application Data
- View application Data
    - Select data perspective
        - Analyse selected Data

The application users (*the conceptual who*) and their requirements (*the conceptual what*) have been identified in relation to the MODAQ application. **[Figure 40]** highlights the users activities for the defined "single" application user type.

7.3.1 - User Class description

A user will be required to login to the application, as the application framework can support multiple application instances, thus multiple "different" users could potentially use the same framework to view different data. The application should allow different user to login to different accounts (*supporting different user accounts*) and analyse specific data. A login capability further requires logout capabilities further allowing users to log in and out of separate accounts.
This user account functionality is not deemed necessary for the application at this prototype stage, however it should still be implemented and supported to a "minimum" level due to security reasons in regard to preventing unauthorised users to access "confidentially" acquired data. Additionally, it is useful to be in place thus making wiring of the "complete" feature easier as and when it is required at a later development stage.
Once logged in, the main "user" task is to easily and efficiently access the acquired data and analyse it. Additional (*and simple*) functionalities deemed relevant for inclusion is the ability to skin the application (*choosing a colour scheme for the applicant*) thus allowing the user to personalise the application (*using company colour schemes ...*) and/or allowing users with poor visibility or users suffering from colour blindness to change the colour scheme of the application to a scheme more useable for themselves (*thus meeting the discussed HCI issues*).
As discussed, the acquired data form the "back-end" system is stored as a ".csv" file which should be locatable from within the application, allowing the file to be selected and further loaded into the application where the data and converting it into a relevant DP for the visual components and thus used to render the data.
Once the data has been loaded, various (*user configured*) perspectives should be enabled for selection, containing various "interactive" visual components, including graphs, charts, gauges and even textual table based renderings that can be used to analyse and interact with the data.

Data interactivity within the application will consist of various filtering possibilities, allowing the user to selected (*filter*) data achieving specific analytical results, for example:

- Only show data measurements of speeds over **x** MPH.
- Only show data measurements of lean angles over **x** degrees.
- Only show data measurements when the engine revs are over **x** RMP.
- Only show data measurements of cornering forces of over **x** G's.
- Only show data measurements between location **x** and **y**.
- Only show data measurements between the times **x** and **y**.

The latter of the above filtering examples is highlights an interesting concept that should be included within the application design. The complete DAQ process is based upon time based measurements (*measurements acquired at a specific time base in relation to a lap*), thus enabling the creation of a "timeline" (*time based*) component that can be used to filter time by changing both start and end times, and additionally allowing data to be "zoomed" in and out on a time based platform, making for the creation of an application control component that is believed to aid data analysis dramatically.

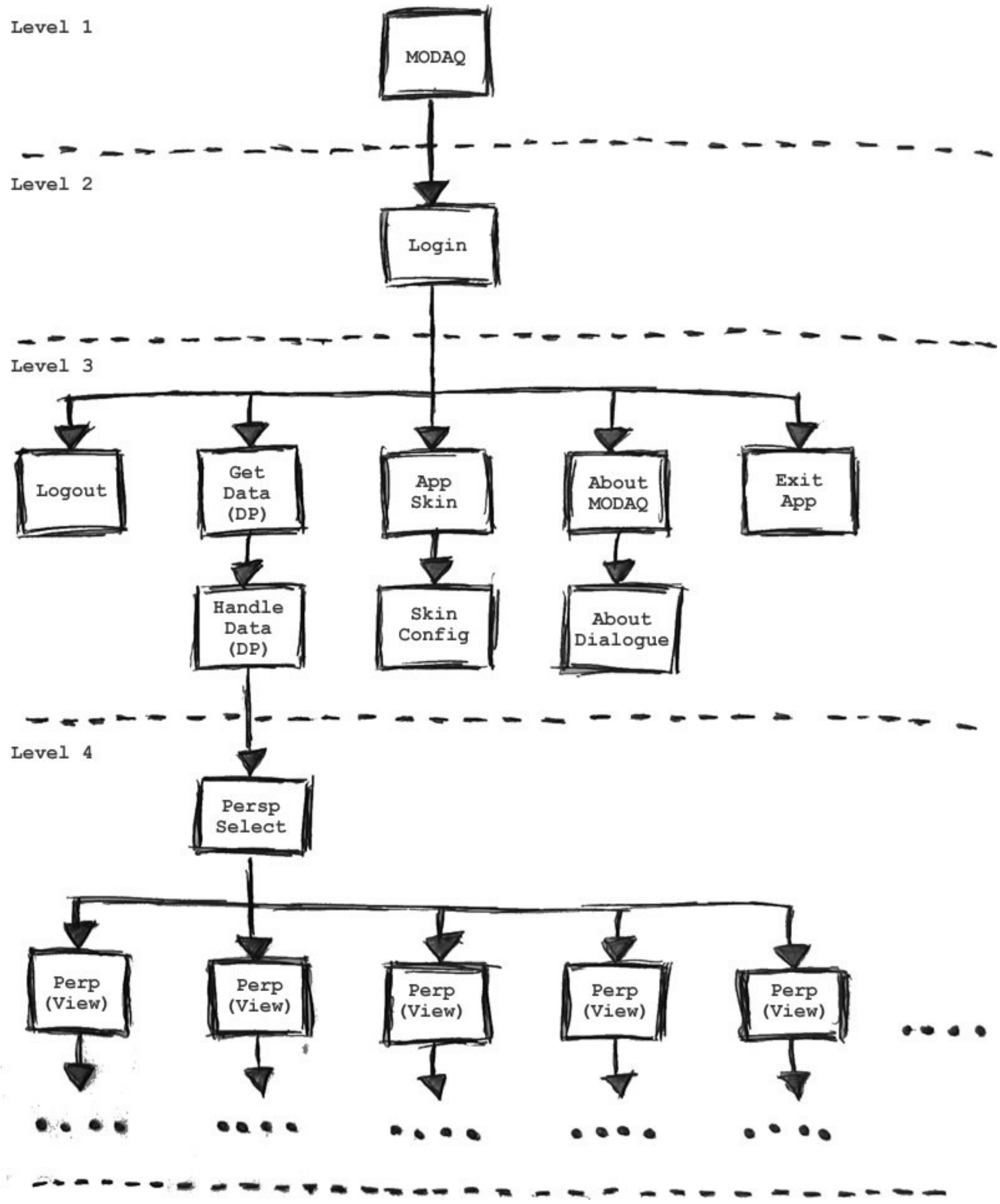
The idea of the discussed application perspectives came from the expansion from a single application to an application based framework. An application can contain many perspectives, and a perspective can contain many views, with the views being components and used to render specific data. Perspectives will be customizable and extendable classes that can be both default and/or user specific (*modified and custom perspectives can be created for different application instances*) thus allowing perspectives to be dynamically tailored to meet specific applications and requirements.
For the PS application, default perspectives will be provided containing visualisations relevant to the PS requirements, further demonstrating how the perspectives can be used and what they can offer to other application instances.

7.4 - Conceptual Design

The application will be approached at a conceptual level, thus dealign with the "what and how" of application design, and the conceptual structure of the application model identifying how the user can navigate through and used the identified application areas.
Further application requirements and considerations could potentially be identified throughout this conceptual process, and along with existing requirements, a "high-level" description will be provided that like the mission sternest, can be used and refereed to throughout further design and implementation procedures.
Commonly within application design, various and generic task models and navigation tracks are created highlighting navigation routes and tasks that can be taken are produced at this stage of the design process, however, for the PS it is deemed more relevant (*and is preferred*) to follow a slightly different (*yet still similar*) modeling method that is believed (*and known due to previous personal experiences and uses*) to be more beneficial and relevant for the application design.
The chosen method for use is taken form the TRIuMPh Methodology, taking and improvising on the **Linear Task Model**, **Non-Linear Task Model** and **Navigation Model** in a way relevant to the PS, highlighting and separating the the navigation possibilities (routes) that can be taken through the PS application along with the identification possible tasks that can be completed, similar to what is illustrated in **[Figure 40].**

A **L**inear **T**ask **M**odel (LTM) will be created following an advance **H**ierarchical **T**ask **A**nalysis (HTA) setting that will be further (rigorously) analysed and grouped logically into areas allowing the identification of how the identified tasks should (*and could*) be both separated and grouped.
The logical groupings performed within the LTM allow the the model to be viewed as "uniformed elements" in relation top the PS application.
The groupings are numbered and further used to create a navigational model, in which navigation operators are added to the distinct groupings enabling the determination of "sequence interaction" between the individual groups, thus providing a navigational flow, hiligting the possible routes that can be taken through the PS application.

[89]-(R Campion et al, 1995)

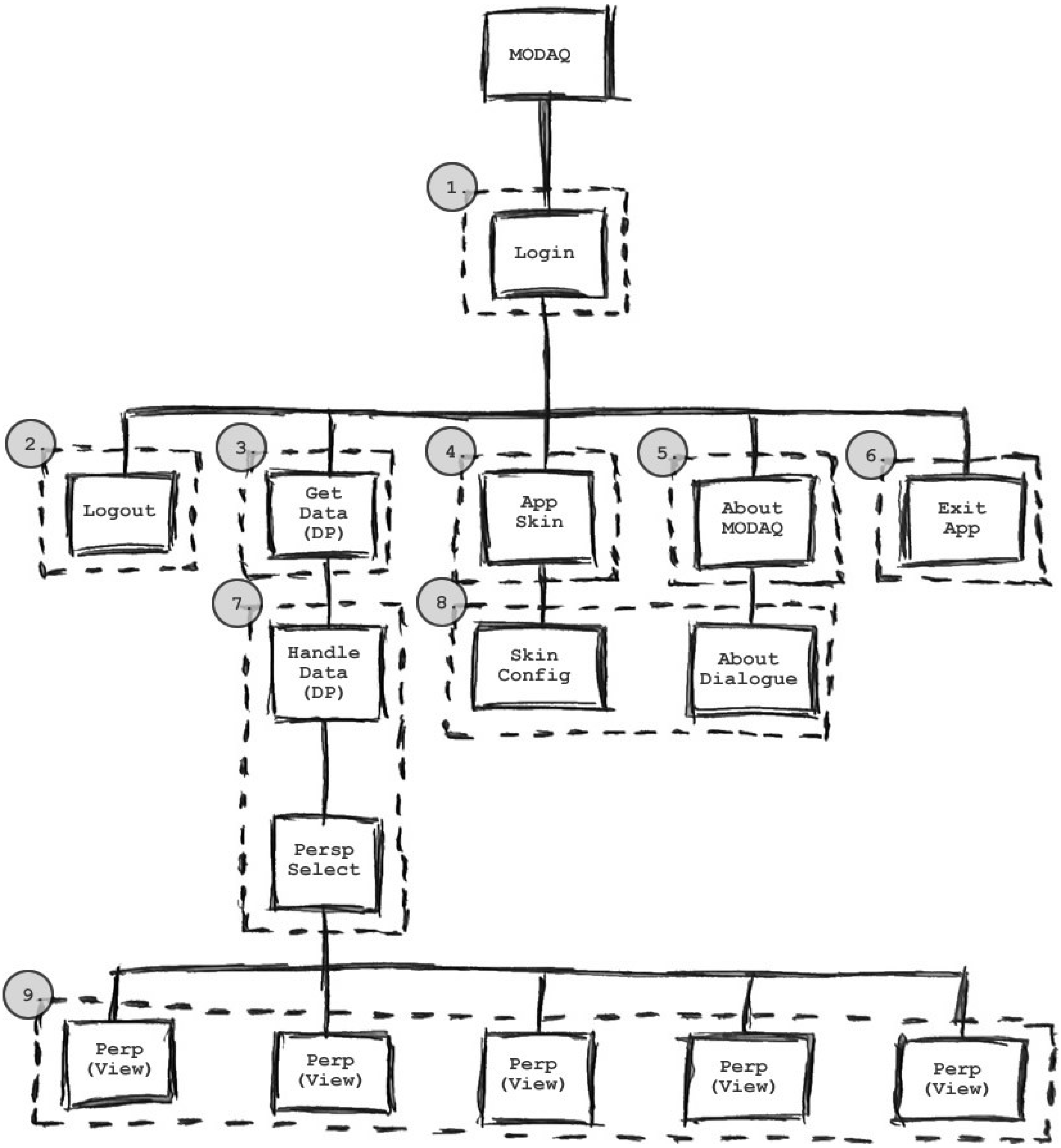


[Figure 41]-(MODAQ Application Linear Task Model)

[Figure 41] Illustrates the PS applications LTM, visually showing the previously discussed application activities and possibilities in a HTA method that further separates the activities into four levels (*areas*) relevant to other activities within the model.

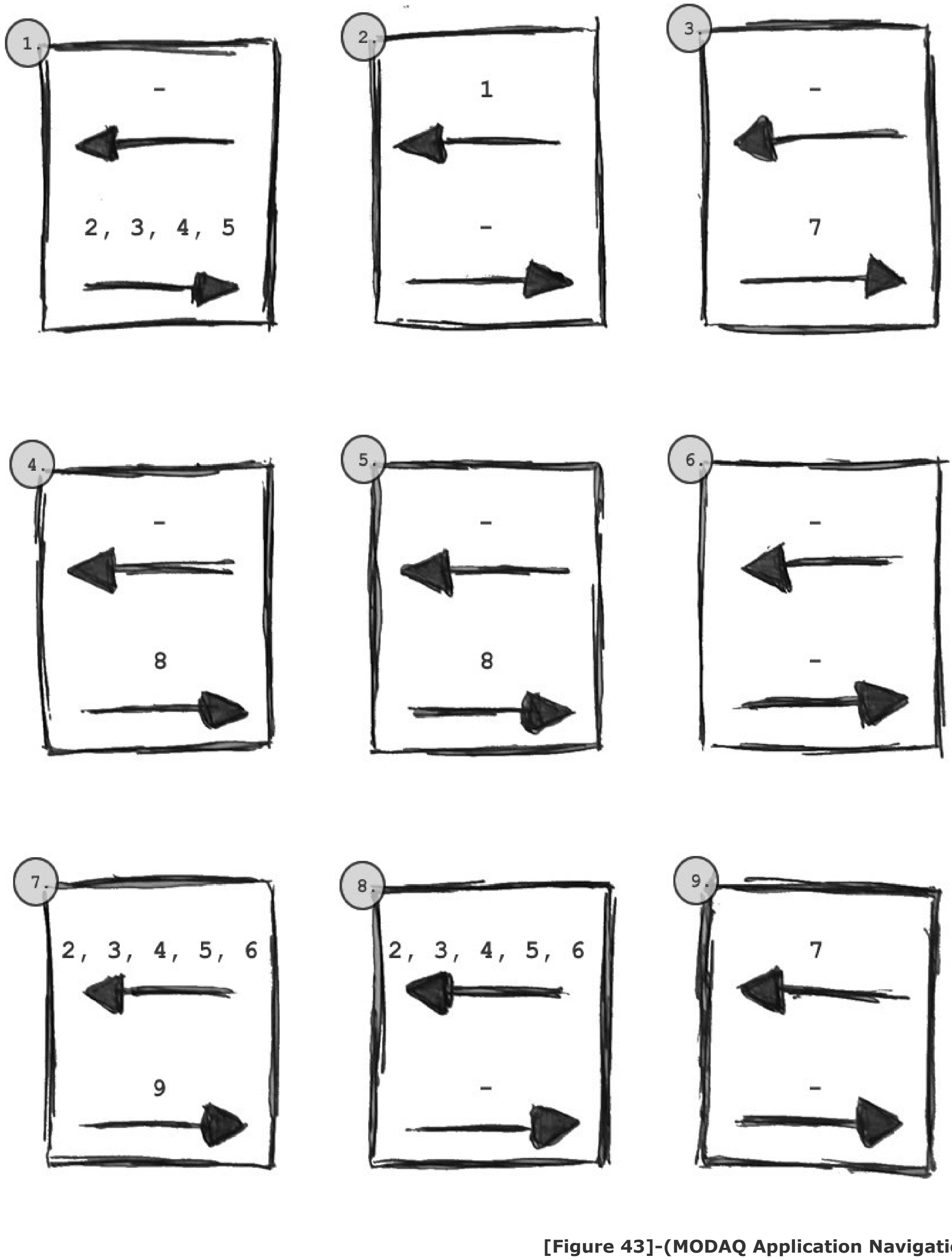
Level 1 consists of the application root (*application start-up*) that once complete will enable **Level 2**, the application Login. **Level 2** will consist two step handler, including a successful login a non successful login. If the login is successful, the user will be forwarded to **Level 3**, however if the login fails (*an incorrect username or password is entered*) the user will be alerted of the problem and sent back to **Level 2.** Once **Level 3** has been established, the user has accessed the main body of the application in which all application possibilities (*tasks*) can be pursued including (*at this prototype stage*) application logout, application exit (*closing the application*), application skinning options (*included for HCI purposes*), application information (*about the application, and application help*) and the option to select a data source to use as a DP for visualisation components.

When a data-source has been selected, it is handled relevantly (*converted into either XML object(s), array(s), vector(s) or any relevant and efficient format that can be used to render the data within components*), and **Level 4** is entered. Within **Level 4**, the user can select a perspective (*from a menus system or list*) containing specified views (*components*) that gets data from the associated DP further plotting and rendering, thus allowing the data to be further analysed.



[Figure 42]-(MODAQ Application Non-Linear Task Model)

[Figure 42] shows the PS application as a **N**on-**L**inear **T**ask **M**odel (NLTM). The LTM from **[Figure 41]** has been taken and spirited into groups in relation to activity similarities. For the PS application these similarities are taken further than an "application" level, however such activity similarities can be translated into object reusability (*form an OOP perspective*) thus re-using classes and reducing code. **[Figure 42]** shows the most of the higher-level activities, mostly ungrouped "individual" thus having individual processes and expectations different to the other activities within the application.
Group 8 (*skinning and information*) can however reuse the same dialogue class (*however with different content*), and **Group 7** (*data handling and perspective select*) can be dealt with in one step. It is **Group 9** that really highlight the advantages of an OOP approach, as each perspective can use (*extend*) a specialised "perspective" based interface consisting of all the default and required methods and functions that such a perspective class would require/use, thus reusing this interface for each class, not only providing for code efficiency and improving maintainability, but also providing consistency within all perspectives throughout the framework.



[Figure 43]-(MODAQ Application Navigation Model)

[Figure 43] shows the logical groups identified within the NLTM from **[Figure 42]** as a **N**avigational **M**odel (NM) consisting of navigational operators assigned to each group and the relevant group numbers that a distinct group can navigate to, further being assigned the the operators within each group, thus enabling the determination of a "sequence-interaction" between individual groups within the PS application, furthermore providing a navigational flow hilighting the routes that can be taken and the tasks that can be preferred at each stage within the application.

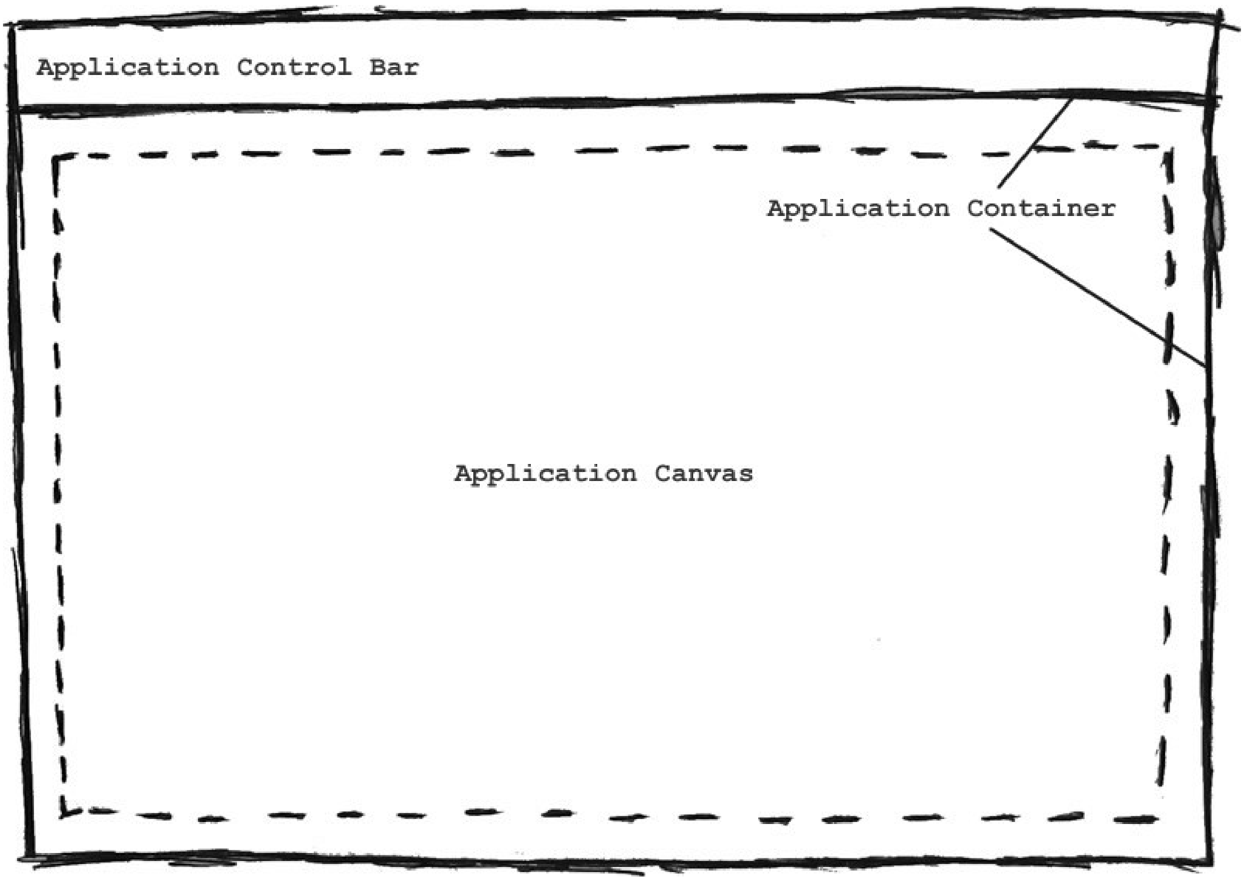
The design for the prototype application (*itself*) does not required a back-end DB. As disused, a simple DB will be used winter the PS, however is added as an "additional" (unrequited) step consisting of simple tables in which the structure is controlled by the acquired data, thus the DB and table structure cannot be "specifically" designed towards apart from using a very "generic" deign method, thus for the PS application (*at this stage*) is is deemed irrelevant to pursue any fundamental DB design methods (***E**ntity **R**elationship **M**odels (ERM) or DB Data Models*) further.
For the same reasons, and as the design is now being tailored more towards an application framework (*rather than an individual application*) further navigational separation including conceptual and non-conceptual flow(s) between links is deemed irrelevant (*the provided NM is enough*) as well as the creation of any additional (*more detailed*) models (*structure or composition*) that would expand on the existing models. Both the framework and OOP selection has reduced the application design required dramatically compared to a static "single" application design, as each element of the application is being designed towards re-usability and efficiency optimisation.

7.5 - Application Storyboard Designs

Storyboards have been created visually showing the basic application design/structures that is wanted to be achieved. Story-boarding is like the task models, a techniques that is highly valued at a personal level, due to previous uses and experiences with such a technique, it has proved a very helpful and worthwhile process allowing for a better (*more visual*) perspective of what the application will look like, and how the separate activities (*as previously identified*) and components can be laid out and used.

For further design, a "traditional" story-boarding technique will be used, allowing for a better understanding process in terms of further interface and layout design and implementation. The storyboards will be used as a template giving the overall feeling for each individual layout, navigation systems/menus and content structure(s).

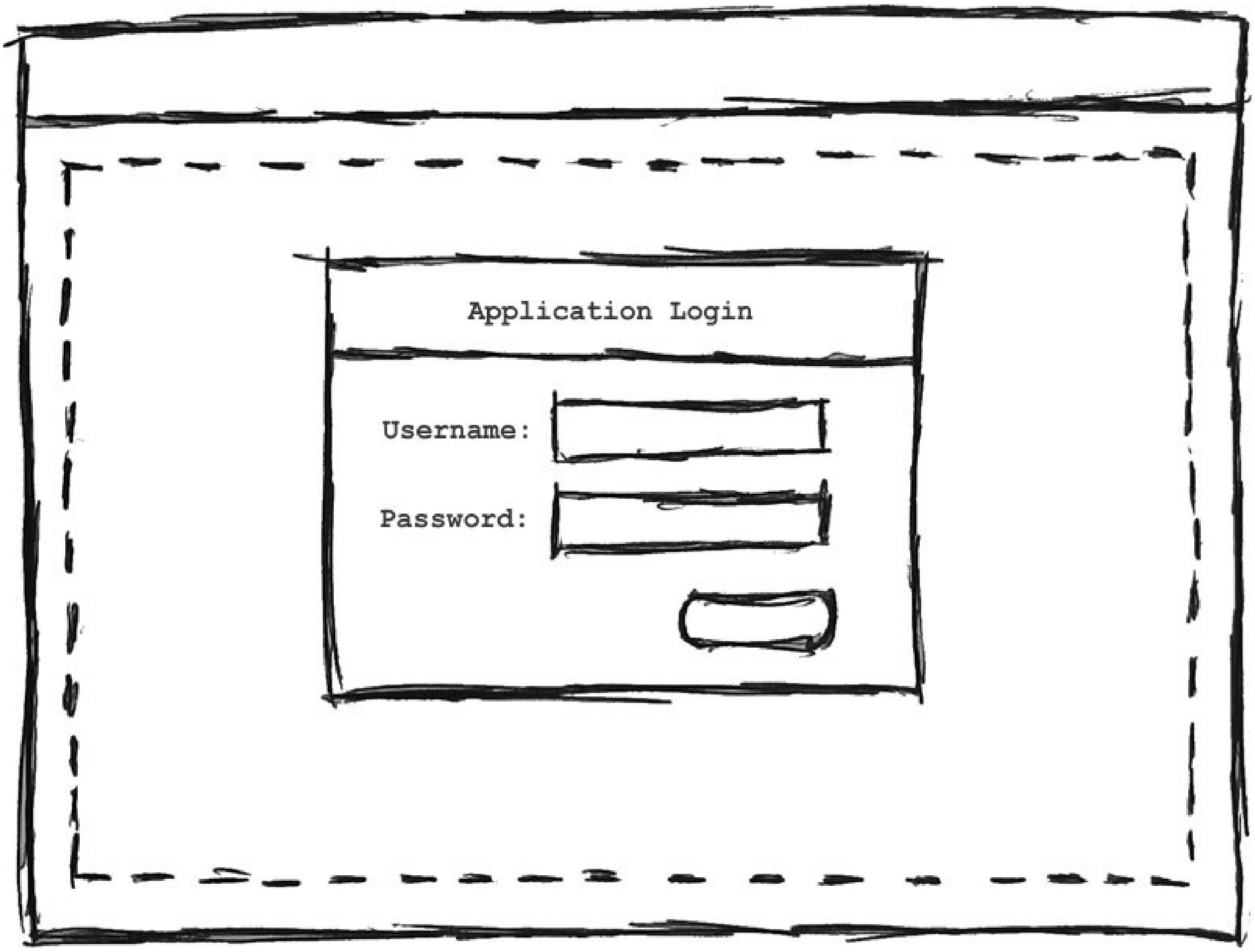
Many different story-boarding techniques exist and could be used for the PS application deign, however it is believed more beneficial to the design flow to follow a simple (*yet effective*) paper-based "mock-up" design, allowing for additional flexibility within the implementation process, thus being an ideal solution for the PS.
The created task and navigation models focus on the different areas and movement through the application, whereas the storyboards will concentrate and allow a more detailed illustration and descriptions of the content of each element (*object*) within the application.
The storyboards contain a rough design (*sketch*) of the visual steps that have been identified and believed necessary for inclusion within the application design, including overall component (*menus and objects*) layout, possible information presentations and any other relevant visual or interaction elements.
The storyboards will offer a a "broad" design layout for the complete application framework, additionally taking in consideration the discussed HCI issues that may arise.
The storyboards will be relevant to expand upon, and will be used in correspondence with the produced task models, thus providing a non-linear application design and development approach.



[Figure 44]-(Storyboard: MODAQ Application Structure)

[Figure 44] Illustrates the initial application framework structure that further application objects and components can be built upon and be integrated within.
The whole application will be wrapped in a canonizer relevant to the runtime environment that the application is being used (*is deployed*) within. In the case of an AIR runtime, the container will be an AIR (*desktop "windowed" based*) application container that is deployed, and ran locally on a users machine, whereas if a Flex based runtime is being used, the container will be a web-browser (*Internet Explorer, Mozilla FireFox, Safari ...*) and will be deployed via the internet (*web*) or on a local web server. The application container itself will have a built in (*runtime specific*) close method (*button*), thus handling the applications exit functionality. A built in exit functionality could also be included within the implementation section if believed necessary.

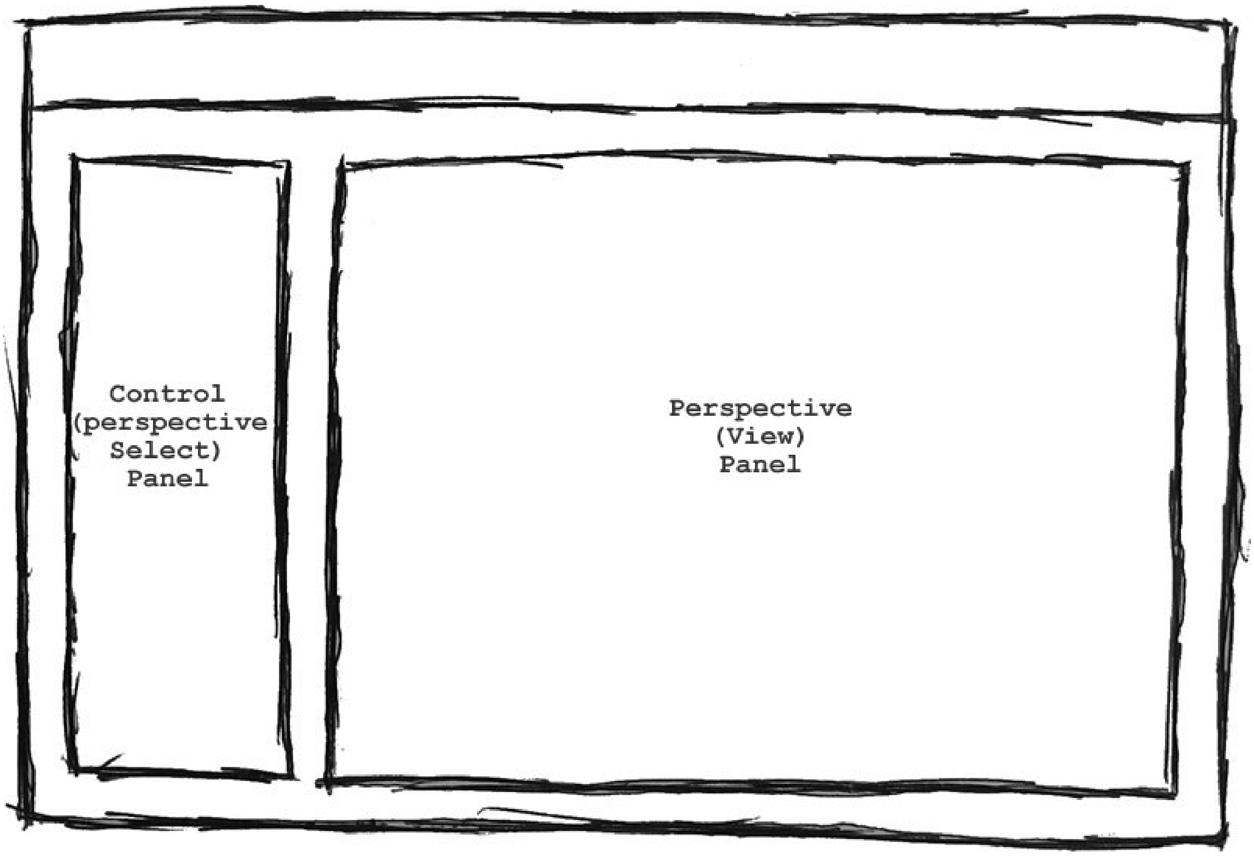
A horizontal application control bar will be positioned at the top of the application in which various application controls can be added. The control bar will be in the form of a high-level based menu system consisting menu options and activities aimed towards the top of the application structure. In the case of the PS the default controls will be added to the control bar consisting of **Get Data**, **Skin Application**, **About Application** and **Exit Application** menus items.
The application container will have a vertical layout thus the child after (*beneath*) the control bar will be the "main" application canvas, where all of the lower-level application elements will be rendered. The canvas itself should ideally support an absolute layout thus allowing complete flexibility within the framework and therefore the efficient extension and addition of further application objects and components (*perspectives and visualisations*).



[Figure 45]-(Storyboard: MODAQ Application Login)

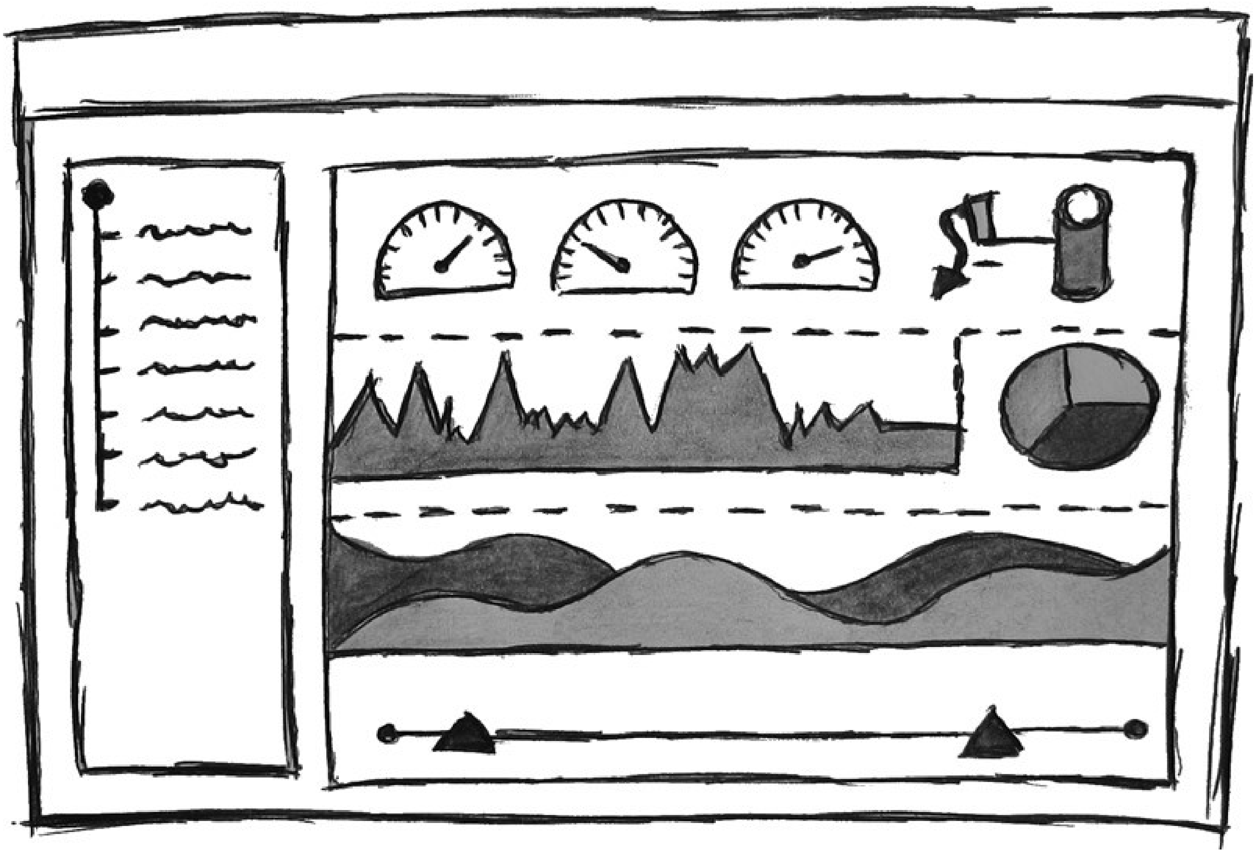
[Figure 45] Illustrates the default application login dialogue that will be used (*and is required*) for all applications using the MODAQ framework. At this prototype stage, the login will consist of a simple (standard) dialogue expecting a users valid "username" and "password", which when entered correctly will remove the dialogue (*as an "Application Canvas" child*) and open the main application body. The dialogue creation and removal requires the use of "window" based containers, Thus opening an additional investigation areas into if and how a specialised window manager could be implemented, and how such a manger could benefit the application.
If however, incorrect login details are entered, the login control should handle the error relevantly and not log the user into the application.

Some sort of user "session" or global variable could (*and should*) be used to set (*communicate*) the application login status throughout each application class and sub-class, as the application should "potentially" (*depending on implementation complexities for a prototype release*) support and handle multiple instances of users logins within a single runtime (*without the application being shut down*) thus users can login, logout and further login again, at different time,s and even using a different accounts (*could be a different user*).



[Figure 46]-(Storyboard: MODAQ Application Component Panel Layout)

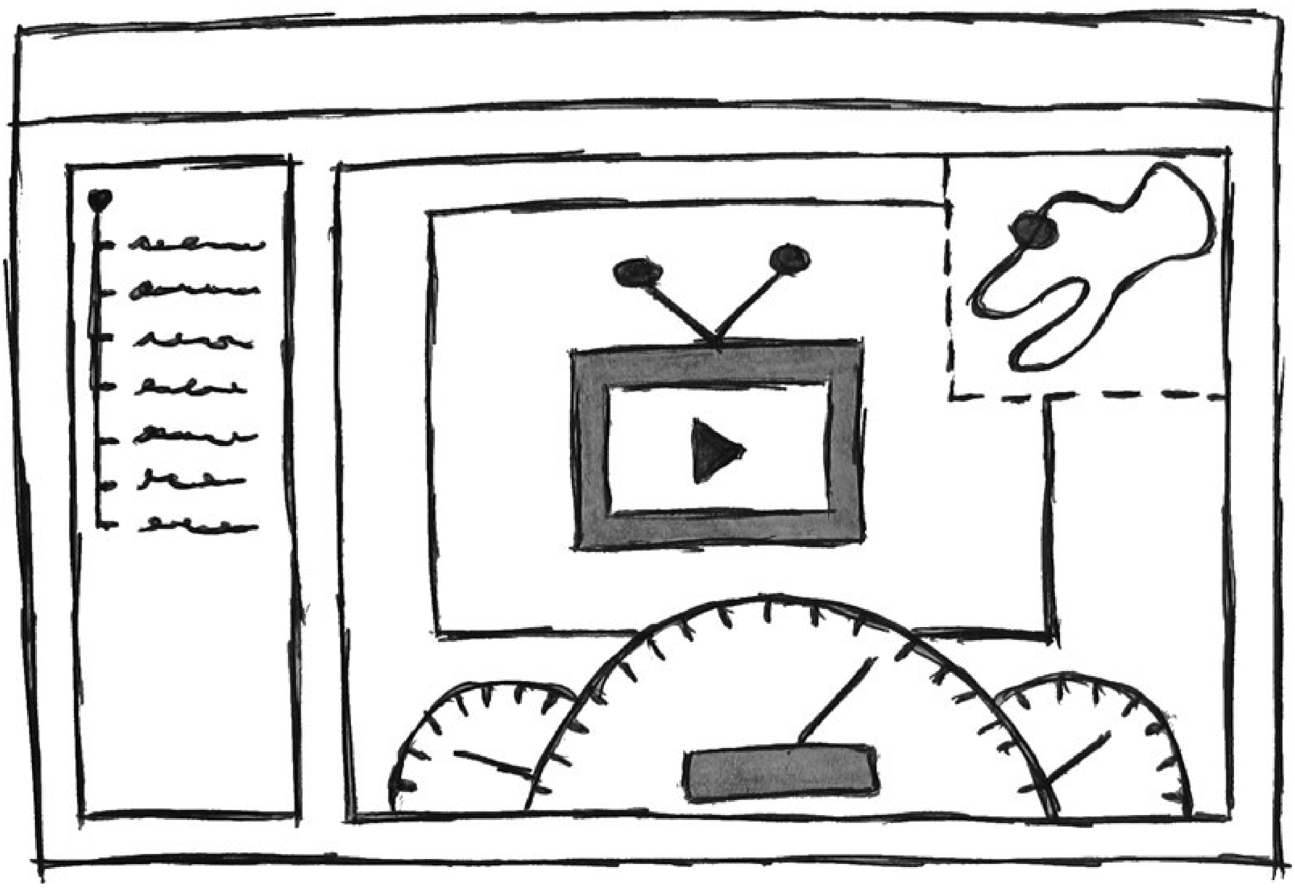
[Figure 46] Illustrates how the application canvas, as shown in **[Figure 44]**, will be potentially divided and laid out. The canvas will consist an embedded horizontal container (*or could simply be changed from having an horizontal layout rather absolute layout*) containing two panels (*containers*). The right hand side container will be be a wrapper containing a control panel, in which the different perspectives (*relevant to an application instance*) will be listed, and can be further selected (*clicked*) with the sleeted perceptive being added (*as a child*) to the left hand side panel.
The advantage of the proposed framework is that each panel will be given an interface that components (*either provided or custom*) can extend and build upon. The interface will be passed a relevant DP, thus in the case of the control (*menu*) panel, a DP containing the relevant perspective list in relation to the currently logged in user, therefore the component class that extends the interface for control panel will have access to this DP, thus rendering the data in a method suitable to a specific application instance. A "simple" default menu component will be provided with the PS in the form of a standard "list" or tree structured view containing the list of perspectives.
The same (*consistent*) method will be applied to the perspective view panel, with a standard interface being provided (*although more complex than the control panel interface*) and being passed a relevant DP, alloying both provided and custom component classes to extend and thus again, render the acquired data in a manner relevant to a specific application.
For the PS application a perspective interface will be written containing sub visualisation classes that further extend the perspective class, containing visualisation components that are relevant to the application, rendering the data and allowing it to be efficiently and accurately viewed.
As the framework is tailored towards DAQ Applications, standard visualisation components will be provided as default (*charts, grids and graphs*) that can be used (*included*) with any perspective, requiring simply a relevant DP, and thus plotting and rendering the data accordingly.



[Figure 47]-(Storyboard: MODAQ Typical Dashboard Perspective)

[Figure 47] Illustrates a "potential" and typical dashboard perspective that can be achieved using the discussed application framework and included as a perspective within the PS application.
The left hand panel contains a "tree" styled menu component that will list the different perspectives available within a specific application, and the right hand panel illustrates a PS perspective view. The view will (as discussed) be provided a DP containing all of the acquired data from a motorcycle lap, and this data will be used to plot and render the different visualisation components within the perspective.
Within this particular perspective design, a "dashboard" styled layout has been targeted, consisting of both vertical and (*in places*) horizontal layout flow containing visualisations relevant to the acquired data. Gauges can be used to represent measurements such as the motorcycles speed, RPM, and engine temperatures during a lap, with various charts and graphs being used to viksualise almost all numeric based data, and even overlapping such data on the same graph (including *multiple graph series*) enabling better analysis and data comprises in regards to different times an locations around during a lap around a racetrack.
The component in the top right hand corner of the perspective is an example of a custom visualisation component that has been created to represent the petrol level of a motorcycle during a lap (*a petrol gauge component*). The component will consist of a "fill" level that will (*in theory*) reduce during a lap(s). This example shows the possibilities of other custom components that can be created (*programmed*) just requiring a little designer imagination and intuitiveness.
Further more examples of such components are illustrated in the following figures.

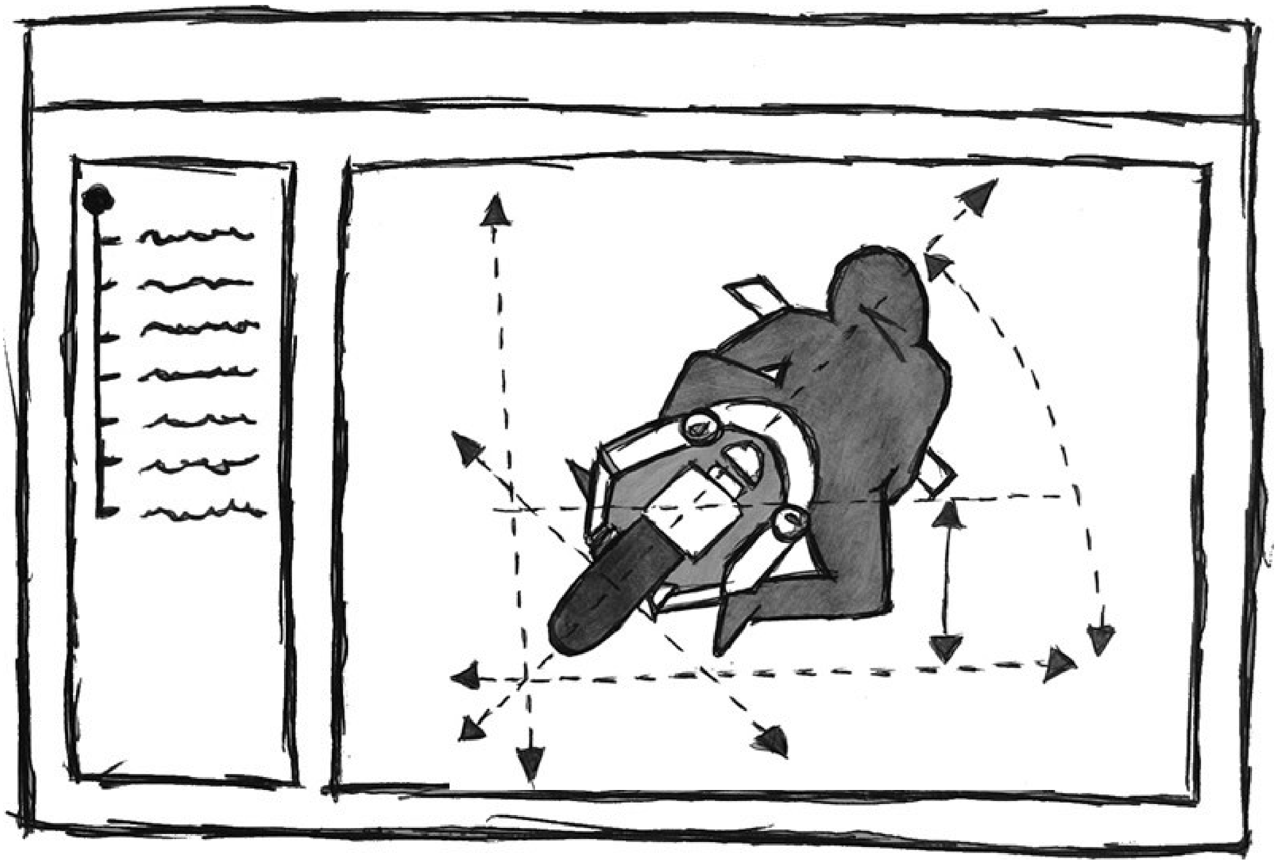
All the acquired data measurements are tie based, thus a simple timeline component (*control*) can be created (*as illustrated at the bottom of the perspective panel*) and used to filter and control the start and end times of the data within the complete perspective, thus making for avery powerful analytical tool, allowing users to filter data to specific time regions and thus racetrack locations.



[Figure 48]-(Storyboard: MODAQ Application Potential Video Perspective Design)

[Figure 48] Illustrates an additional perspective design that embeds a video control. As discussed within the R&I chapter, an expansion to the project could be to install a specialised (*small*) video camera either on a motorcycles petrol tank, or on the riders helmet, and record (*video*) a lap(s). This video data can be compressed, synchronised and further transmitted with the other required measurements. Additional and relevant components added to the perspective can include a motorcycles dashboard representation consisting of a speedometer, rev-counter and temperature/petrol gauge, and a track map/model (*as discussed previously within this chapter*) all being populated by the same data source (SP) and thus being time synchronised with the video player itself, overall making for a very interesting and useful component.

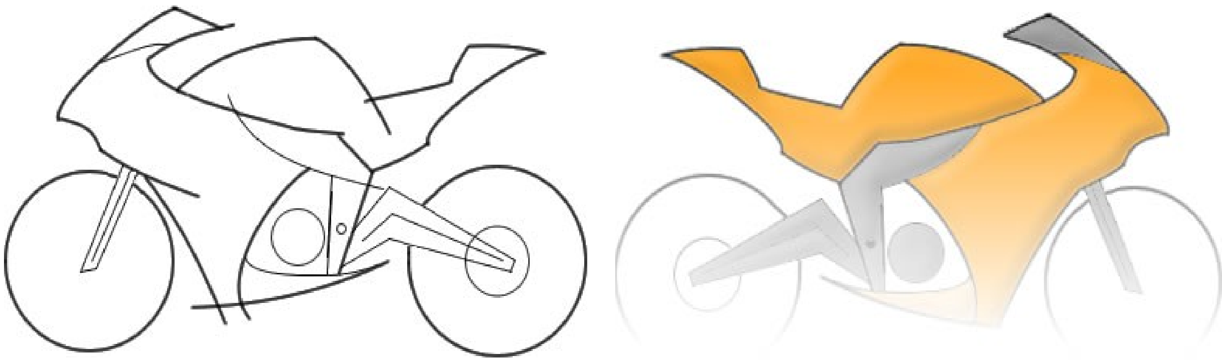
The video player can, for example include "standard" video player controls (*play, stop, pause, fast forward and rewind ....*) therefore not only controlling the video, but also controlling the videos DP, and thus further controlling any other (*linked*) components within the same perspective as they will all use, and be connected to the same DP, therefore the components can be controlled, and remain "in-sync" with the video player, even when it is stopped, paused or skipped.



[Figure 49]-(Storyboard: MODAQ Application Potential Lean Perspective Design)

[Figure 49] Illustrates the final perspective design deemed relevant for use within the PS. The perspective is designed to represent a motorcycles lean angle(s) during a lap using the discussed and calculated lean angle measurements.
The perspective consists of a custom component containing a motorcycle model (*representing a real motorcycle*) and is synchronised to both the time and location measurements in relation to a racetrack lap. The perspective could also benefit from additional (*relevant*) components such as a timeline control and also a racetrack model/map being included, as when all the components are working together (*and are in sync*) would again make for a very impressive and effective analytical result
The component will have a default angle of "0" thus the motorcycle model would be rendered straight (up-right) and will further use the calculated lean angle measurements to re-render the models (*motorcycle*) lean angles accordingly in relation to the "current" lap time being analysed. The model will animate (*use the AFP tweening behavior*) between measurements, thus making for a smooth and effective overall result.

7.6 - Application Logo Design



[Figure 50]-(Application Logo Designs)

[Figure 50] Illustrates the logo designed for use and incorporation with the PS application. Although not relevant in regards to meeting the application requirements, and/or not affecting the creation of a successful application development (*and for these reasons alone will not be discussed in any further detail*), a logo is believed (at a personal level) relevant to all systems and applications, thus has been designed and created.
The logo represents the PS application, and has been created (*for more marketing related reasons*) enabling and creating application recognition. The logo will represent the application as an entity, and consists of an image/shape (*content*) that is both rememberable (can be easily associated with) and relevant to the PS application.

The logo design process can be found in **[Appendix 12]-(Application Logo Design Process)**.

It is believed that enough design work and design has been completed to enable both the creation of a prototype application that meets the PS requirements, and therefore to successfully and confidently move onto the application implementation phase.
The design of the application has been discussed and justified in terms of **user classifications**, **user and application tasks**, application navigation and flow, application components and children, **application structure** and **application layout** to a high and in depth level believed satisfactory for further project implementation to proceed.
A good knowledge and understanding has been established in relation to what the application will look like, how it will flow and how the overall layout of the application will be structured. It is additionally (*believed to be*) known and understood how the proposed design can be achieved in relation to the application implementation, regarding a solution of how the different design areas will be “attacked” and programmed providing a successful final artifact.
For these reasons, the PS application development process will move into its next phase, and ths applicaton implementation process will commence.

8 - Application Implementation

Due to both financial and practical reasons, a real and complete "back-end" system cannot be produced for this project. For such a system access to a motorcycle, racetrack and all of the investigated and selected hardware and technologies would have to be available, which is unfortunately (*in this case*) not possible. However, a prototype front end analytical application can be developed as an example (POC) representing what the end result of such a system can achieve using "test" data.

8.1 - Application Data-provider Creation Implementation

The developer of the investigated system (*RaceFX*) and contact ("ken Milnes") that was established during the lLR chapter has kindly provided "real" sample data that was taken using the RaceFX system during an IndyCar race at the famous St Petersburg racetrack, thus proving very beneficial to the prototype application as accurate, reliable and real data and be used as a DP (although being related to a car rather than a motorcycle) allowing the possibilities and results of the application, being a lot more realistic and believable to that of a real world situation opposed to the use of wither test or simulated data.
The data was acquired from an IndyCar with using discussed DAQ devices within the RaceFX system. The devices were installed on the car during the race, continuously measuring and acquiring the following data types (areas) from a 12 lap race:

- Time
- Lap
- Lap fraction (*LapFraction*)
- Distance to wall (*DistanceToWall*)
- Latitude
- Longitude
- North
- East
- Altitude
- Speed
- Heading
- Steering
- Linear Acceleration (*LinearAccel*)
- Lateral Acceleration (*LateratAccel*)
- Vertical Acceleration (*VerticalAccel*)
- Position Standard Deviation (*PositionStandardDeviation*)
- Number of satellites tracked (*NumSatellites*)
- Revs Per Minute (RPM)
- Brake Pressure (*brake*)
- Throttle Pressure (*Throttle*)
- Gear
- Fuel

The data is relatively self explanatory, with the speed being in MPH, the longitudinal and latitudinal angles being in decimal degrees and the altitude being in meters above the WGS44 ellipsoid. The naming conventions within the brackets of each area, represent the headings that will be during the development of the PS application.

A summary/selection of the raw data transmitted from the cars DAQ device, and used as a DP throughout the following implementation and testing application development phases can be found in a tabular format in **[Appendix 13]-(Sample DAQ Application Data)**. Not all data can be included within the report due to the amount of data acquired (4688 rows), however the complete data file(s) can be found on the CD enclosed with this report.

The data from transmitted from the DAQ devise is saved in an ".csv" file format, thus to start with, the sample data will be put back into its original ".csv" format to produce an accurate and relevant development foundation. A summary/selection of the data in its original ".csv" file format can be found in **[Appendix 14]-(Sample DAQ ".csv" Data File)**.

As the PS application will be developed in Flex, the easiest and most efficient (*best perform-ant nature*) is for an XML file to be created containing all of the data required to be fed into the Flex application therefore use the XML file as an application DP. However this is not so efficient in relation to creating an XML file from a ".csv" file.
Applications and tools exist to create XML from such ".csv" content, however it is (*at a personal level*) preferred if complete control over this process can be adopted, and additionally allowing for any structure of ".csv" file to be handled and converted into a valid XML document, thus making for a very flexible and useable application that can be used for all different analysis possibilities and differing types of DAQ devices.

As the ".csv" is going to be handled and processed manually, it is believed good proactive to sore the data in a DB as an intermediate step (*although due to the use of flex a DB is not required*). This step will enable tracking and a backup of all uploads that can be further selected within the application.
The ".csv" conversion will therefore follow the process of taking the ".csv" file, creating a relevant table within the DB and inserting the data relevantly into the created table, further exporting this inserted data into a structured and valid XML document.

To support such DP handling, the creation of a custom script is believed the best (*and only*) reasonable method to support such a "custom" and dynamic process.
For the script to be as useable as possible, it is required to be additionally as flexible and dynamical as possible. The main job of the script is to:

- Take a ".csv" file that can be in any format and any location (*user specific*), thus some sort of navigation/selection option allowing a user to point to (*select*) a specific ".csv" file to use as a DP for the PS application.
- The selected ".csv" file should formatted and used to create new table in a DB with relevant column headings within the file. The table should be named accordingly (*relevant to the application and ".csv" file name, and the data will be stored in the DB for access and backup until a user specifically chooses to remove the table*). In many " default" and standard cases, the column headings are contained within the first row of a ".csv" file, and if such headers do not exist a default naming convention and creation for column headers needs to be handled and implemented.
- For flexibility, the selected ".csv" files can be of any format (*support different amounts of rows and columns and additionally different data types,* ***int, string, boolean*** *...*) thus additionally requireing the table generation process must also be flexible, creating dynamic tables in the structure (*format*) of the ".csv" files.
- The final step of the process, is to take the inserted data from the DB table and dynamically create and structure and XML file from the data.

As discussed, a relevant script is required to be created to enable such a DP generation process. A web-based scripting language is preferred as it can be deployed both locally (*depending in a relevant server is installed on a local box*) and if not over the internet, being the preferred location as it is therefore accessible from any machine and location and at any time.

There are many dynamic scripting methods and possibilities that could be used and adopted (*all achieving the same/similar result*). Such possibilities are general separated into two main areas including Server Side Scripting (***P****re* ***H****ypertext* ***P****rocessor* (*PHP),* ***J****ava* ***S****erver* ***P****ages (JSP),* ***A****ctive* ***S****erver* ***P****ages (ASP) and ASP.NET ...*) and Client Side Scripting (***J****ava****S****cript (JS)*), and again further separated into three main (*programming approach*) areas including Programatic, Template and Hybrid approaches.
Due to the amount of scripting possibilities available, a selection process has been performed with the most relevant scripting languages in relation to the PS development, and further investigation has been completed with these selected areas.

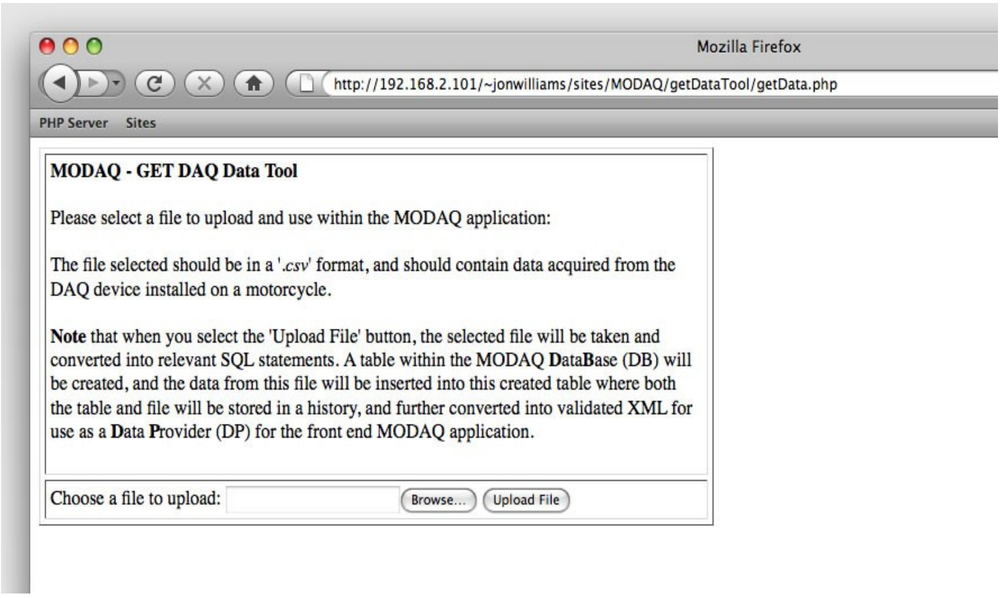
Investigation into these areas has not been documented within the report body as it is deemed irrelevant, as this stage is (*currently*) not only a POC and experimental stage of the development cycle, and is not compulsory in regards to the applications success.
After investigation, the preferred language for use is **PHP** due to cost (*open source*), power, flexibility and OOP support and additionally existing knowledge in and around the language, thus improving the speed and quality of the script development.

An additional (*and discussed*) step and requirement that has been (re)-added to the development process is that of DB interaction, requiring an DB acting as an "intermediate" step during the creation process of the application DP.
There are many possible and available DB's that can be used with the PS. There are not many requirements of the DB in relation to the PS, apart being easily interacted with from an external script (*in this case a PHP script*) and other general DB requirements common in most applications including good performance, accessibility, and the possibility to support the amount of data required by a particular system (*in the case of the PS application, up tp thousands of rows*).
Due to the fact that PHP is the chosen scripting language that will be used to interact/communicate with the cheesed DB, MySQL is the obvious choice as it is the most compatible DB for use with PHP (*they were designed for use with one another*) having many built in MySQL related functions, thus removing the need for such DB interactions having to be programmed as these provided interaction classes and methods that can be used/called, simply passing in relevant and required variables as and where necessary, saving both time and complications during the creation of such scripts.
Another advantage that MySQL has is that is (*like PHP*) open source further having a vast amount of online support and documentation averrable.
In its latest released version, MySQl is also very powerful (*to a more than satisfactory level for the PS*) and being further comparable to to that of proprietary DB's.
If for whatever reasons during any project expansion processes, it is believed that MySQL is not sufficient, further investigation into a more advance proprietary DB's will be investigated further, however, at this stage of development, and for what is required for a prototype system, it is believed that MySQL is more than sufficient to meet the PS requirements and thus further DB investigation is deemed irreverent at this stage.

The following "implementation" related sections within this chapter regard complex procedures and processes. Most of the areas will be at a programming (*coding*) base regarding the steps and programming approaches/methods sued to create the final artifact, again being at a very in-depth and detailed (*complex*) level.
For these reasons, the following implementation sections descriptions will follow more of a visual description using images (*screenshots*) during showing stages and results during the PS application implementation. This method will make for better and easier reading and understandings regarding the implementation phase and its achievements in comparison to an in depth technical complex and descriptive method, explaining in detail the actual code itself.
The scripting code (*MXML, AS and PHP*) and the DP files (*".csv" and created XML document*) can however be found on the CD enclosed with this report.

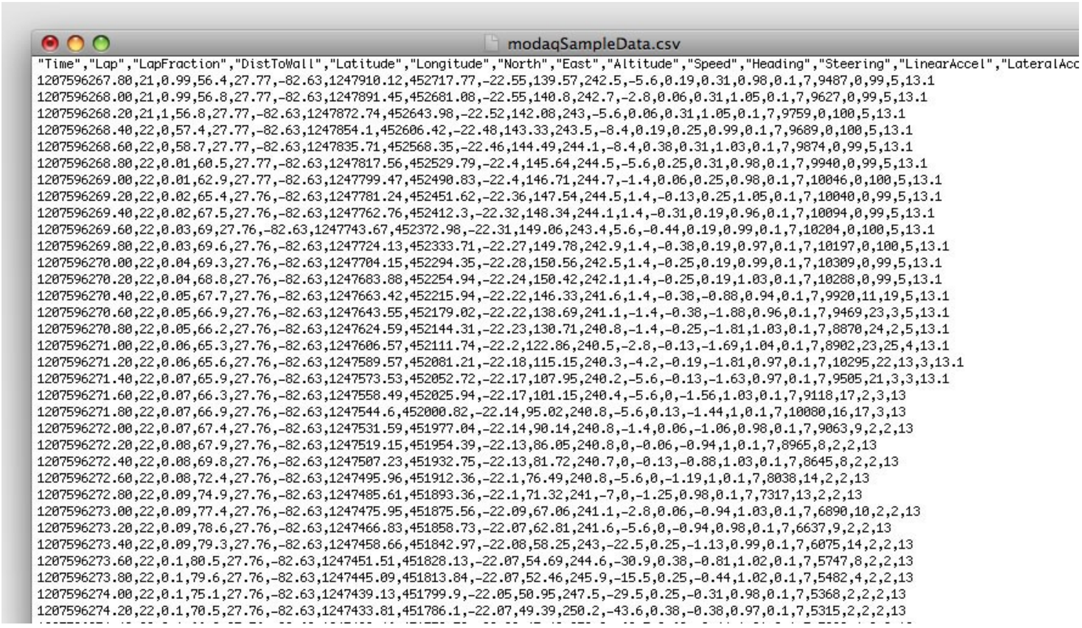
8.1.1 - Dynamic Data Provider Creation Script

A very simple HTML/PHP based UI has been created to allow a user to select and upload a ".csv" file, containing all the acquired motorcycle measurements that will be used as the application DP.



[Figure 51]-(Simple Data Provider Selection User Interface)

[Figure 51] Illustrates the initial UI with a simple and easy to use file select input and submit button. The UI is not skinned (at this stage) and therefore has a very basic appearance, which is fine at this (*testing*) prototype stage of development.



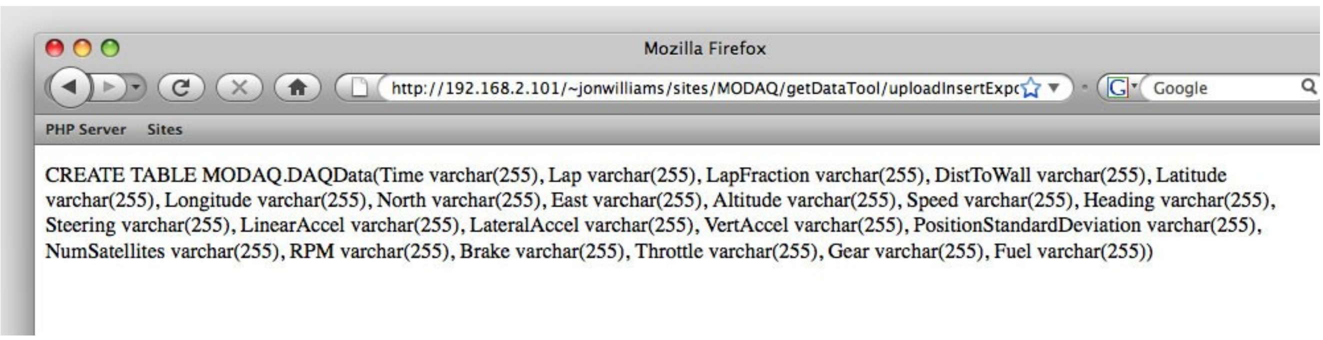
```
modaqSampleData.csv
"Time","Lap","LapFraction","DistToWall","Latitude","Longitude","North","East","Altitude","Speed","Heading","Steering","LinearAccel","LateralAcc
1207596267.80,21,0.99,56.4,27.77,-82.63,1247910.12,452717.77,-22.55,139.57,242.5,-5.6,0.19,0.31,0.98,0.1,7,9487,0,99,5,13.1
1207596268.00,21,0.99,56.8,27.77,-82.63,1247891.45,452681.08,-22.55,140.8,242.7,-2.8,0.06,0.31,1.05,0.1,7,9627,0,99,5,13.1
1207596268.20,21,1,56.8,27.77,-82.63,1247872.74,452643.98,-22.52,142.08,243,-5.6,0.06,0.31,1.05,0.1,7,9759,0,100,5,13.1
1207596268.40,22,0,57.4,27.77,-82.63,1247854.1,452606.42,-22.48,143.33,243.5,-8.4,0.19,0.25,0.99,0.1,7,9689,0,100,5,13.1
1207596268.60,22,0,58.7,27.77,-82.63,1247835.71,452568.35,-22.46,144.49,244.1,-8.4,0.38,0.31,1.03,0.1,7,9874,0,99,5,13.1
1207596268.80,22,0.01,60.5,27.77,-82.63,1247817.56,452529.79,-22.4,145.64,244.5,-5.6,0.25,0.31,0.98,0.1,7,9940,0,99,5,13.1
1207596269.00,22,0.01,62.9,27.77,-82.63,1247799.47,452490.83,-22.4,146.71,244.7,-1.4,0.06,0.25,0.98,0.1,7,10046,0,100,5,13.1
1207596269.20,22,0.02,65.4,27.76,-82.63,1247781.24,452451.62,-22.36,147.54,244.5,1.4,-0.13,0.25,1.05,0.1,7,10040,0,99,5,13.1
1207596269.40,22,0.02,67.5,27.76,-82.63,1247762.76,452412.3,-22.32,148.34,244.1,1.4,-0.31,0.19,0.96,0.1,7,10094,0,99,5,13.1
1207596269.60,22,0.03,69,27.76,-82.63,1247743.67,452372.98,-22.31,149.06,243.4,5.6,-0.44,0.19,0.99,0.1,7,10204,0,100,5,13.1
1207596269.80,22,0.03,69.6,27.76,-82.63,1247724.13,452333.71,-22.27,149.78,242.9,1.4,-0.38,0.19,0.97,0.1,7,10197,0,100,5,13.1
1207596270.00,22,0.04,69.3,27.76,-82.63,1247704.15,452294.35,-22.28,150.56,242.5,1.4,-0.25,0.19,0.99,0.1,7,10309,0,99,5,13.1
1207596270.20,22,0.04,68.8,27.76,-82.63,1247683.88,452254.94,-22.24,150.42,242.1,1.4,-0.25,0.19,1.03,0.1,7,10288,0,99,5,13.1
1207596270.40,22,0.05,67.7,27.76,-82.63,1247663.42,452215.94,-22.22,146.33,241.6,1.4,-0.38,-0.88,0.94,0.1,7,9920,11,19,5,13.1
1207596270.60,22,0.05,66.9,27.76,-82.63,1247643.55,452179.02,-22.22,138.69,241.1,-1.4,-0.38,-1.88,0.96,0.1,7,9469,23,3,5,13.1
1207596270.80,22,0.05,66.2,27.76,-82.63,1247624.59,452144.31,-22.23,130.71,240.8,-1.4,-0.25,-1.81,1.03,0.1,7,8870,24,2,5,13.1
1207596271.00,22,0.06,65.3,27.76,-82.63,1247606.57,452111.74,-22.2,122.86,240.5,-2.8,-0.13,-1.69,1.04,0.1,7,8902,23,25,4,13.1
1207596271.20,22,0.06,65.6,27.76,-82.63,1247589.57,452081.21,-22.18,115.15,240.3,-4.2,-0.19,-1.81,0.97,0.1,7,10295,22,13,3,13.1
1207596271.40,22,0.07,65.9,27.76,-82.63,1247573.53,452052.72,-22.17,107.95,240.2,-5.6,-0.13,-1.63,0.97,0.1,7,9505,21,3,3,13.1
1207596271.60,22,0.07,66.3,27.76,-82.63,1247558.49,452025.94,-22.17,101.15,240.4,-5.6,0,-1.56,1.03,0.1,7,9118,17,2,3,13
1207596271.80,22,0.07,66.9,27.76,-82.63,1247544.6,452000.82,-22.14,95.02,240.8,-5.6,0.13,-1.44,1,0.1,7,10080,16,17,3,13
1207596272.00,22,0.07,67.4,27.76,-82.63,1247531.59,451977.04,-22.14,90.14,240.8,-1.4,0.06,-1.06,0.98,0.1,7,9063,9,2,2,13
1207596272.20,22,0.08,67.9,27.76,-82.63,1247519.15,451954.39,-22.13,86.05,240.8,0,-0.06,-0.94,1,0.1,7,8965,8,2,2,13
1207596272.40,22,0.08,69.8,27.76,-82.63,1247507.23,451932.75,-22.13,81.72,240.7,0,-0.13,-0.88,1.03,0.1,7,8645,8,2,2,13
1207596272.60,22,0.08,72.4,27.76,-82.63,1247495.96,451912.36,-22.1,76.49,240.8,-5.6,0,-1.19,1,0.1,7,8038,14,2,2,13
1207596272.80,22,0.09,74.9,27.76,-82.63,1247485.61,451893.36,-22.1,71.32,241,-7,0,-1.25,0.98,0.1,7,7317,13,2,2,13
1207596273.00,22,0.09,77.4,27.76,-82.63,1247475.95,451875.56,-22.09,67.06,241.1,-2.8,0.06,-0.94,1.03,0.1,7,6890,10,2,2,13
1207596273.20,22,0.09,78.6,27.76,-82.63,1247466.83,451858.73,-22.07,62.81,241.6,-5.6,0,-0.94,0.98,0.1,7,6637,9,2,2,13
1207596273.40,22,0.09,79.3,27.76,-82.63,1247458.66,451842.97,-22.08,58.25,243,-22.5,0.25,-1.13,0.99,0.1,7,6075,14,2,2,13
1207596273.60,22,0.1,80.5,27.76,-82.63,1247451.51,451828.13,-22.07,54.69,244.6,-30.9,0.38,-0.81,1.02,0.1,7,5747,8,2,2,13
1207596273.80,22,0.1,79.6,27.76,-82.63,1247445.09,451813.84,-22.07,52.46,245.9,-15.5,0.25,-0.44,1.02,0.1,7,5482,4,2,2,13
1207596274.00,22,0.1,75.1,27.76,-82.63,1247439.13,451799.9,-22.05,50.95,247.5,-29.5,0.25,-0.31,0.98,0.1,7,5368,2,2,2,13
1207596274.20,22,0.1,70.5,27.76,-82.63,1247433.81,451786.1,-22.07,49.39,250.2,-43.6,0.38,-0.38,0.97,0.1,7,5315,2,2,2,13
```

[Figure 52]-(Example ".csv" data file)

[Figure 52] Illustrates the format of test data used and acquired within the RaceFX system converted into the relevant ".csv" file format for use with the PS (*and how it would be stored and downloaded from the Nology DAQ device*). It can be seen that the first row (*line*) in the file contains the required column headers, and it is these headers that will be used when creating the DB table and the nodes within the outputted XML file. The rest of the rows (*lines*) within the file are simple comma "," separated (CSV) strings relevant (*in terms of order*) to the column headers within the same file.

When a file is selected, and the submit button is "clicked" within the UI, the file type and content are checked and handled accordingly, for example, if the file type is incorrect or the file contains unexpected (*or corrupted*) content. However, if the file appears ok, an advance PHP script is called and performs the necessary processes required to create the XML based application DP.

The first step of the PHP script is to connect to the relevant server and DB and create a table that containing all of the data from the selected ".csv" file. The PHP script reads the ".csv" file taking the headers from the file and building a valid SQL Statement that will be used to created a DB table. Each contained **column** named is looped through and pushed into the SQL string thus constructing a complete statement that is, after a validation process, executed and thus if successful, a table is created within the DB.

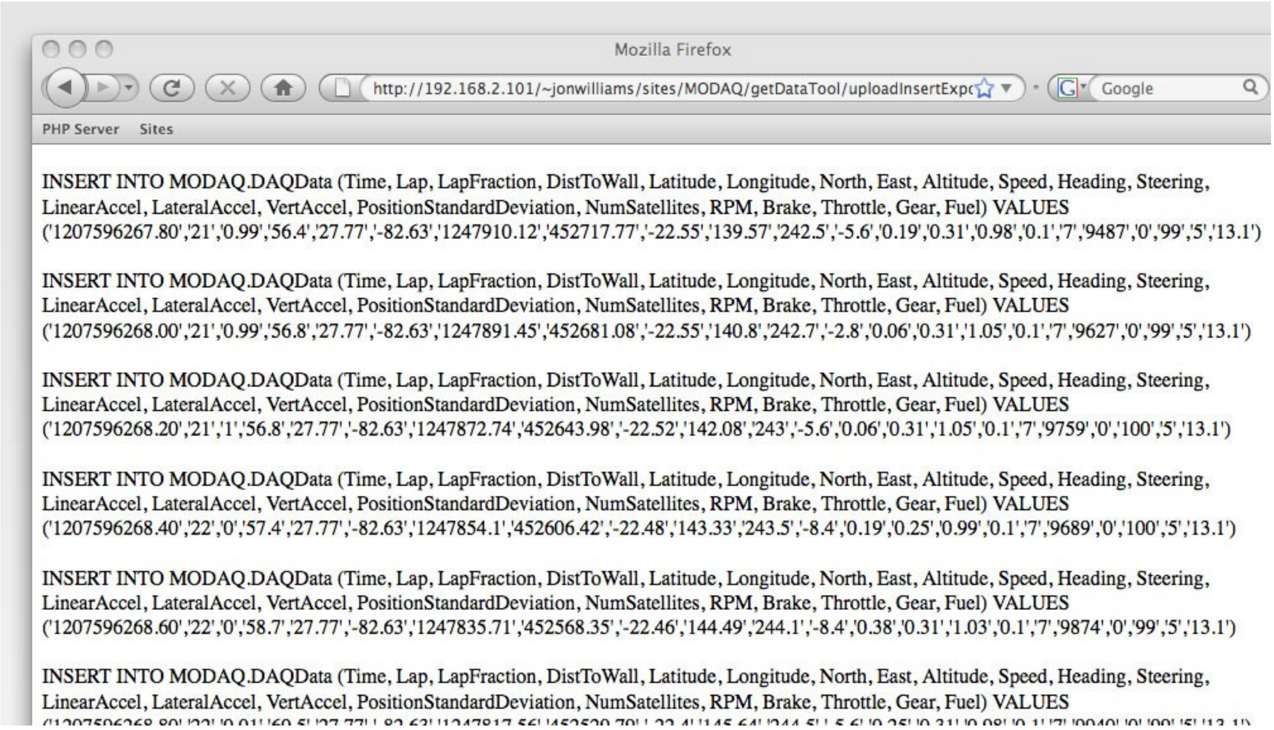


CREATE TABLE MODAQ.DAQData(Time varchar(255), Lap varchar(255), LapFraction varchar(255), DistToWall varchar(255), Latitude varchar(255), Longitude varchar(255), North varchar(255), East varchar(255), Altitude varchar(255), Speed varchar(255), Heading varchar(255), Steering varchar(255), LinearAccel varchar(255), LateralAccel varchar(255), VertAccel varchar(255), PositionStandardDeviation varchar(255), NumSatellites varchar(255), RPM varchar(255), Brake varchar(255), Throttle varchar(255), Gear varchar(255), Fuel varchar(255))

[Figure 53]-(Constructed "create table" SQL statement created from a ".csv" file)

[Figure 53] Illustrates the constructed "create table" used for the test data. All columns are set to a type string (varchar(255)), however in the case of the PS, the types can be checked within the client and further typed if required. This is enough for the prototype application, however a future improvement could be to enhance the PHP script to check and type the data before it is inserted into the DB.

So far, a table has been created in the DB, however this table is currently empty as no records (*rows*) have yet been added (*inserted*). The next stage within the PHP script is to loop through all of the data **rows** within the ".csv" file, and in a similar way to how the "create table" SQL Statement was constructed, an "insert into" SQL Statement will be constructed for each of the rows, therefore **x** amount (*depending on the amount of rows contained within the ".csv" file*) of "insert into" statements will be constructed and executed, inserting each row synchronously (*one after the other*) into the previously created DB table.



[Figure 54]-(Constructed "insert into" SQL Statements created from a ".cvs" file)

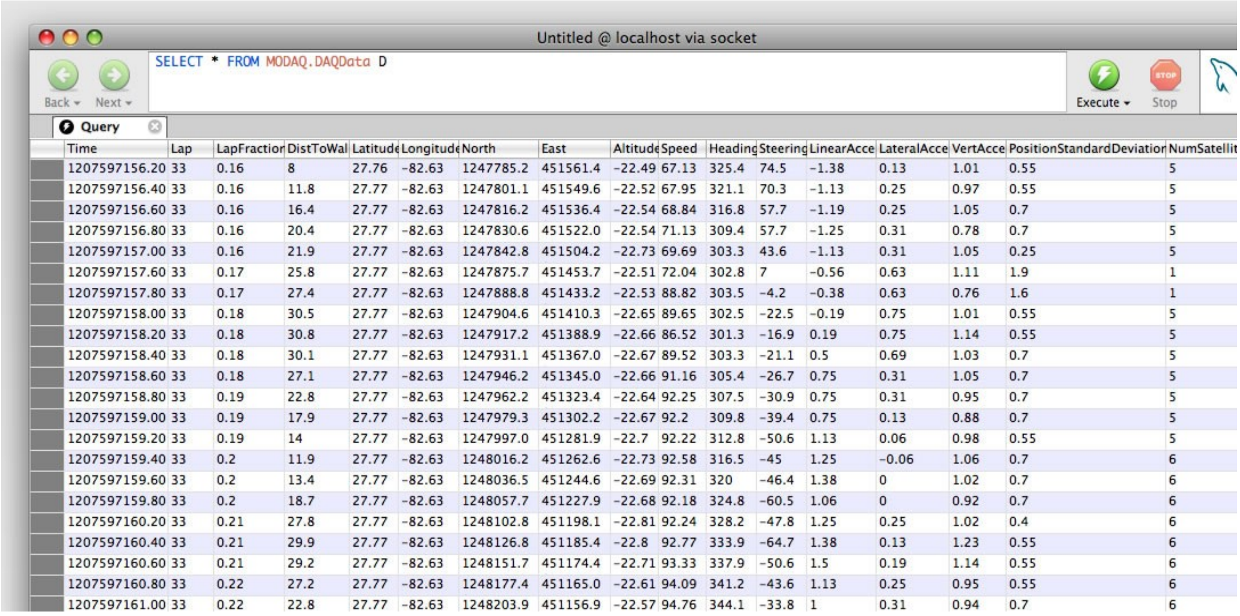
[Figure 54] Illustrates some of the discussed PHP script created "insert into" SQL statement in relation to the test data file.

Inserting each data row individually as separate SQL statements is probably not the best (*most efficient*) method of data insertion into a DB. A more efficient method could be the construction of a single SQL statement that creates and adds the data into rows in one complete process.

A single process insert requires the construction of an advanced, complex and large (*especially for use with such amounts DAQ data*) statement to be created, thus proving difficult to create. Although difficult, is still completely possible (*and still relevant*) to create dynamically within the PHP script, however the main reason this "complete" statement approach was not taken is due to maintainability and resource intensiveness reasons. Smaller individual statements are easier to maintain, debug and control in caparison to an extensive single statement. As (*in the case of the created script*) if an SQL error is to occur within the statement, it will be traced wthilnth the PHP output and assigned a relevant execution line number to refer to, thus allowing to debug the error referencing and "attacking" a single statement opposed to a large advanced statement, being similar to the popular phrase: "*finding a needle in a haystack"*.

Additionally, if such a large statement is to be ran in one attempt (process), it can be very (*depending on the amount of data required to be inserted*) resource (*memory, CPU and DB*) intensive and could possibly cause crashes and failures on the server, whereas if the statement is separated into individual "row" related statements (*although being more time intensive*) will be less resource intensive as each statement will be executed synchronously, opening the connection, executing the statement and the closing the connection again, thus freeing up resources after each execution, and additionally handling errors in a more friendly and efficient manner as the loop can be broken or skipped when an error occurs (*allowing problems to be solved in a synchronous manner*).

In relation to the test data used, the acquired data file contains 4688 rows, thus requiring 4688 "insert into" SQL statements to be executed. On the MODAQ application development box (*containing 3gb of memory and running on MacOS*) being used for this project, these 4688 SQL statements require less than a second for all to be executed, which is both very reasonable and a lot better than what was initially expected, additionally proving and supporting the justification and efficiency of not only the PHP script but also the chosen MySQL DB.

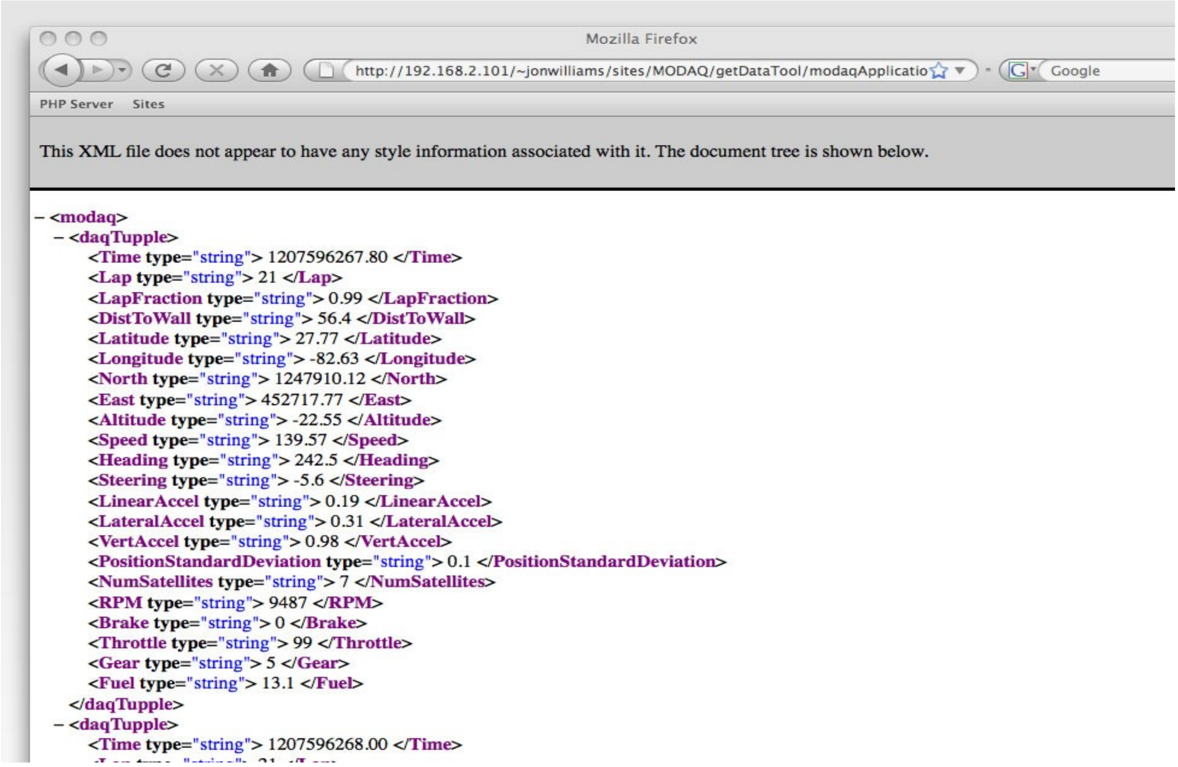


Time	Lap	LapFractior	DistToWal	Latitude	Longitude	North	East	Altitude	Speed	Heading	Steering	LinearAcce	LateralAcce	VertAcce	PositionStandardDeviatior	NumSatellit
1207597156.20	33	0.16	8	27.76	-82.63	1247785.2	451561.4	-22.49	67.13	325.4	74.5	-1.38	0.13	1.01	0.55	5
1207597156.40	33	0.16	11.8	27.77	-82.63	1247801.1	451549.6	-22.52	67.95	321.1	70.3	-1.13	0.25	0.97	0.55	5
1207597156.60	33	0.16	16.4	27.77	-82.63	1247816.2	451536.4	-22.54	68.84	316.8	57.7	-1.19	0.25	1.05	0.7	5
1207597156.80	33	0.16	20.4	27.77	-82.63	1247830.6	451522.0	-22.54	71.13	309.4	57.7	-1.25	0.31	0.78	0.7	5
1207597157.00	33	0.16	21.9	27.77	-82.63	1247842.8	451504.2	-22.73	69.69	303.3	43.6	-1.13	0.31	1.05	0.25	5
1207597157.60	33	0.17	25.8	27.77	-82.63	1247875.7	451453.7	-22.51	72.04	302.8	7	-0.56	0.63	1.11	1.9	1
1207597157.80	33	0.17	27.4	27.77	-82.63	1247888.8	451433.2	-22.53	88.82	303.5	-4.2	-0.38	0.63	0.76	1.6	1
1207597158.00	33	0.18	30.5	27.77	-82.63	1247904.6	451410.3	-22.65	89.65	302.5	-22.5	-0.19	0.75	1.01	0.55	5
1207597158.20	33	0.18	30.8	27.77	-82.63	1247917.2	451388.9	-22.66	86.52	301.3	-16.9	0.19	0.75	1.14	0.55	5
1207597158.40	33	0.18	30.1	27.77	-82.63	1247931.1	451367.0	-22.67	89.52	303.3	-21.1	0.5	0.69	1.03	0.7	5
1207597158.60	33	0.18	27.1	27.77	-82.63	1247946.2	451345.0	-22.66	91.16	305.4	-26.7	0.75	0.31	1.05	0.7	5
1207597158.80	33	0.19	22.8	27.77	-82.63	1247962.2	451323.4	-22.64	92.25	307.5	-30.9	0.75	0.31	0.95	0.7	5
1207597159.00	33	0.19	17.9	27.77	-82.63	1247979.3	451302.2	-22.67	92.2	309.8	-39.4	0.75	0.13	0.88	0.7	5
1207597159.20	33	0.19	14	27.77	-82.63	1247997.0	451281.9	-22.7	92.22	312.8	-50.6	1.13	0.06	0.98	0.55	5
1207597159.40	33	0.2	11.9	27.77	-82.63	1248016.2	451262.6	-22.73	92.58	316.5	-45	1.25	-0.06	1.06	0.7	6
1207597159.60	33	0.2	13.4	27.77	-82.63	1248036.5	451244.6	-22.69	92.31	320	-46.4	1.38	0	1.02	0.7	6
1207597159.80	33	0.2	18.7	27.77	-82.63	1248057.7	451227.9	-22.68	92.18	324.8	-60.5	1.06	0	0.92	0.7	6
1207597160.20	33	0.21	27.8	27.77	-82.63	1248102.8	451198.1	-22.81	92.24	328.2	-47.8	1.25	0.25	1.02	0.4	6
1207597160.40	33	0.21	29.9	27.77	-82.63	1248126.8	451185.4	-22.8	92.77	333.9	-64.7	1.38	0.13	1.23	0.55	6
1207597160.60	33	0.21	29.2	27.77	-82.63	1248151.7	451174.4	-22.71	93.33	337.9	-50.6	1.5	0.19	1.14	0.55	6
1207597160.80	33	0.22	27.2	27.77	-82.63	1248177.4	451165.0	-22.61	94.09	341.2	-43.6	1.13	0.25	0.95	0.55	6
1207597161.00	33	0.22	22.8	27.77	-82.63	1248203.9	451156.9	-22.57	94.76	344.1	-33.8	1	0.31	0.94	0.7	6

[Figure 55]-(DAQ Data Loaded into MySQL Database)

[Figure 55] Illustrates the data as it is in the MySQL DB (*after the discussed SQL statements have been processed*). Once in the DB, the first part of the PHP script is then complete.
The data is stored in the DB where is will stay (*as a backup*) until the users specifies it is not longer required and is deleted. The next job of the PHP script is to read this data from the DB and convert it into an valid XML document that will be used as the DP for the PS application (*this process is discussed further within this chapter*).

Once the PHP script has complete the process of talking a ".csv" file, creating and new DB table from the file and inserting the contents of the file into the created table, further exporting the data from the DB table and converting it into valid XML, saving the converted XML data in a valid XML docment. When complete and the XML document is complete, the user is promoted that the final XML DP file is available and can then further locate the document form within the MODAQ application and further laod and render the data accordingly.



[figure 56]-(Completed XML Application Data Provider File)

[figure 56] Illustrates the final generated XML output file created within the PHP script. The parent child of the XML document is the "**<modaq>**" containing (4688) "**<daqTupple>**" child nodes with each of these "**<daqTupple>**" nodes parenting child nodes relevant to the initial headers set in the ".csv" file.

In the case of the test data used for the PS development, containing the child nodes include: "**<Time>**", "**<Lap>**", "**<LapFraction>**", "**<DistToWall>**", "**<Latitude>**", "**<Longitude>**", "**<North>**", "**<East>**", "**<Altitude>**", "**<Speed>**", "**<Heading>**", "**<Steering>**", "**<LinearAccel>**", "**<LateralAccel>**", "**<VertAccel>**", "**<PositionStandardDeviation>**", "**<NumSatellites>**", "**<RPM>**", "**<Brake>**", "**<Throttle>**", "**<Gear>**", "**<Fuel>**" all being relevant to the previously discussed and identified file column headers.

A more detailed extract of the produced XML document can be found in **[Appendix 15]-(Generated XML Data Provider Document Extract).**

There are many ways and possibilities of creating an XML document from DB results, with the chosen (and preferred) method for this project being the use of a PHP script.
The PHP script created selects all the data from the DB by executing a "Select * From" SQL statement. Any filters on the data that is required in the final XML data should be appended to this SQL statement as the results from this steaming will be looped through to create the final XML document.
After the statement has been executed, the variable **$num** is set that represents the number of rows that are in the DB table, this variable is measured using the built in PHP MySQL function "**mysql_num_rows**". The **$num** variable is referenced throughout the XML data creation as it represents the presence of row data from the SQL query. If the **$num** variable is greater than '0' then it is assumed that data exists in the DB table, and therefore it is relevant for the process to continue. If however the **$num** variable is equal to '0' then there is no data in the DB table and therefore there is no reason to continue and create an empty XML document, thus this is considered as an error and is handled appropriately.

If data exists, the variable **$xmlFileName** is created and set to the name of the file that the created XML document will be named. The script checks if a file with that mane already exists on the server, and if it does it is deleted, and a new (*empty*) file is created and named accordingly (set *equal to* **$xmlFileName**) and its permissions are set so that the file can be written to, thus allowing content to be printed to the file from the PHP script.

This process is satisfactory at this prototype stage, however in a "real" situation when being ran on a live and online web-server such process (*in its current state*) would have some security issues that would have to be dealt with. To make the process more secure for real world applications, a full file path should (*at least*) be provided to a directory that contains the relevant files that are to be handled (*in this case opened or written to*) additionally making sure that the directory is positioned at a level "above" the servers web-root.

Once the file has been created, the XML content can start to be constructed and written to the file. A connection to the file is established within the PHP script and the file is opened (*through the established connection*).
The variable **$_xml** is created and is initially set equal to an empty string ("") that will be appended upon (*constructed*) throughout the XML document process with nodes and sub-nodes (*parents and children*) being created and added to the string in a format relevant to that of a valid XML document.
All valid XML documents require a "document descriptor" consisting of a header tag containing the XML version, and the XML encoding that they use. For example, the following string is appended to the **$_xml** variable as the first line (*the header*) of the XML document: "**<?xml version=\"1.0\" encoding=\"UTF-8\" ?>**".
Once the document descriptor is in, the content body can be written to the document in the style of nodes (*tag based elements*) following parent and child architecture. The complete document will be wrapped the an application parent node "**<modaq>**" containing child and sub-child nodes, thus the first node appended to the **$_xml** variable is the "**<modaq>**" node.
Each row within the DB table is looped through using another built in PHP MySQL function "**mysql_fetch_array**" and is linked with the initially executed selection query.
For each row iteration a node is added "**<daqTubble>**" and within this node (*and iteration*) the DB columns are additionally looped across and the column (*header*) names and data types are acquired, again using another PHP MySQL function "**mysql_fetch_field**" (*thus justifying the reason of choice for both PHP and MySQL for such a process in relation to saving time and complexity due to the built in/provided functions*) with these values being used to create **x** amount of child nodes (*depending on the number of columns in a DB table*) of the parent node "**<daqTubble>**". As each node is created it is appending the the **$_xml** string, and as each iteration is completed the child nodes are closed relevantly "**</daqTubble>**", and at the end of the document the parent node is closed relevantly "**</modaq>**".

Once the iteration process is complete and all children and parent nodes have bee closed relevantly. The final **$_xml** variable consists of a sting containing valid XML content and is thus written to the file (*represented as* **$xmlFileName**) and the established connection to the file is closed.

Overall the PHP script results in the backup (*and local storage*) of data acquired form the "back-end" MODAQ system within a DB, and further the creation of a valid XML document containing structured content of a parent and child node nature of the data from the created DB table.

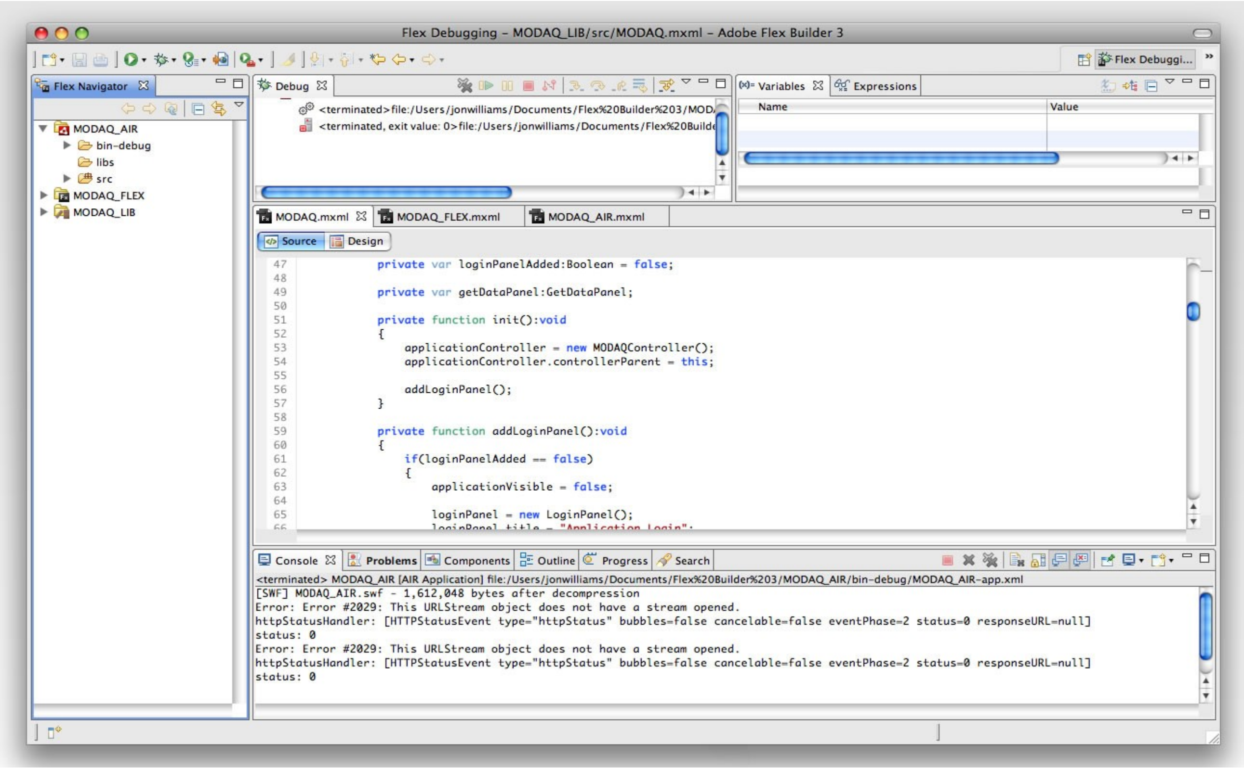
The complete (*and commented*) PHP script that is used for this process can be found in **[Appendix 16]-(Data Provider PHP Script)**.

8.2 - Flex Application Implementation Process

The previously discussed PHP script has taken the data acquired and provided the "back-end" system, and used it to create a DB and XML documents that can be further used as a DP within the "front-end" application.

This section of the chapter, will discuss and document the implementation of the PS application using the Adobe Flex technology. As with the previous implementation section, in relation to programming the "front-end" application, most of the areas will be at a programming (*coding*) base regarding the steps and programming approaches/methods used to create the final artifact, being at a very in-depth and detailed (*complex*) level.
The development of the "front-end" application will be a lot more "in-depth" and a lot more and complex code will be required due to its initial OO application (*framework*) setup. The implementation will require many foundation and application related objects consisting of different **interfaces**, **factories**, **classes** and both **AS** and **MXML components**, resulting in a lot of coding to be completed that it deemed irrelevant to discuss in detail within the report body. Therefore the process will not be discussed at a "code" level, however the process will be summarised and discussed at a visual level using relevant illustrations and screenshots where appropriate to represent the work that has been completed during the application implementation phase(s).
This approach will make for better and easier reading and understandings regarding the implementation phase in comparison to a complex and descriptive approach.

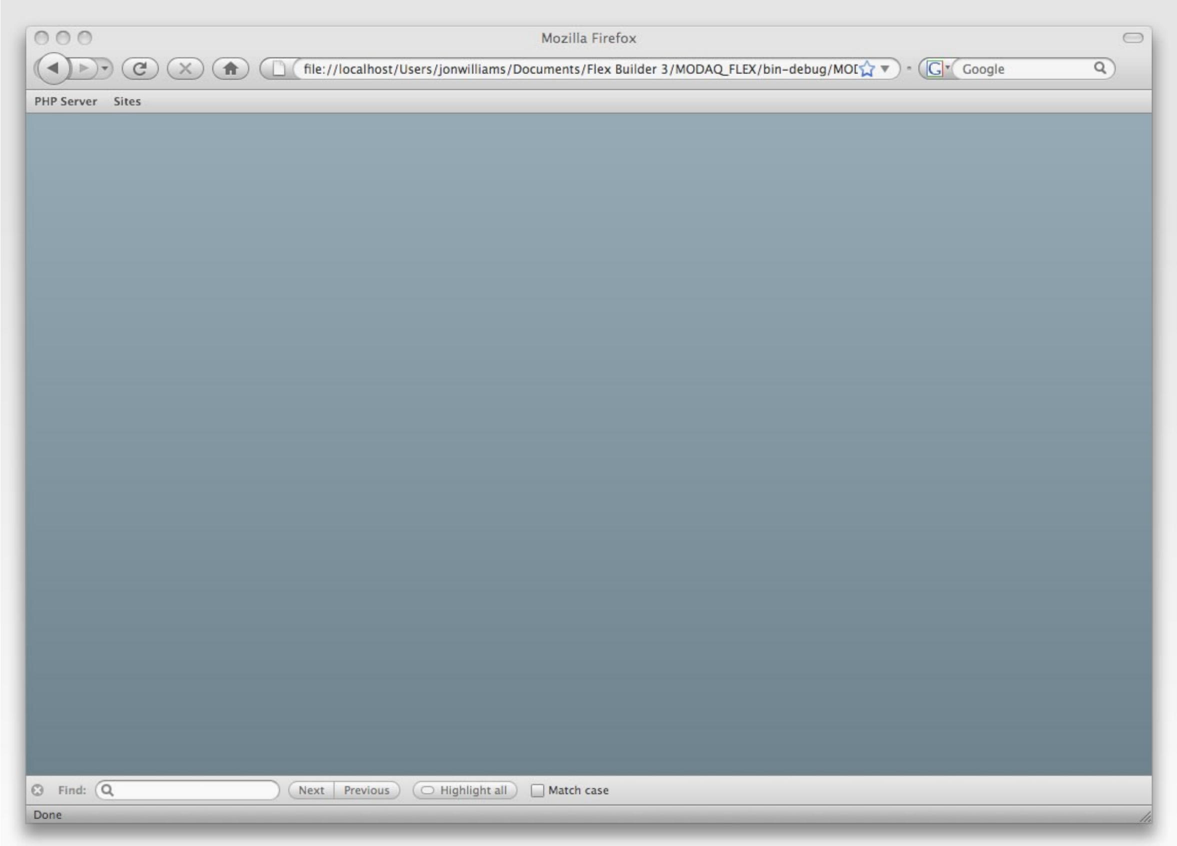
The fist stage of the UI development process is to create a foundation to build the framework and further applications from. As discussed, the application is to run in both a desktop and web based environment, therefore must support the relevant runtimes of each.
Flex supports both Flex (*web-based*) and AIR (*desktop based*) runtimes, therefore a project for each is required.
Code is not wanted to be duplicated between the two projects for obvious maintainability and code control issues, therefore a separate library will be created that will contain all of the application code.
The Library will contain a "root" application class that the complete application development will build and compile from.
The two projects (*Flex and AIR*) will include this library within their project build paths, adding the discussed root application class to there distinct application runtime files, therefore "wrapping" the class within the relevant runtime builds (*one for Flex and one for AIR*).
The Flex and AIR projects will be very small containing very little code (*apart from the library wrapper*), with all code and work being completed in the library project.
The two projects will only include a simple MXML application files containing the application root tags relevant to the particular runtime:
The Flex MXML application root tag is "**<mx:Application … />**" and the AIR MXML application root tag is "**<mx:WindowedApplication … />**".



[Figure 57]-(Application Project Setup)

[Figure 57] illustrates the simple project setup within the Flex Navigator panel on the left hand side of the screenshot. The top project is the AIR application ("**MODAQ_AIR**"), the middle id the Flex application ("**MODAQ_AIR**") and the is the discussed library project containing the source code for each ("**MODAQ_LIB**").

This method will result in two applications being created that can be compiled and exported for use within two separate runtime environments, however using the same source code (*library*) thus making code maintenance very efficient, as code only has to be changed and maintained from one location.
The use of Libraries and using code between applications is standard OOP practice, however the ability of these practices within the AFP for use within two completely different runtimes (*including a web-based*) runtime, has (*before Flex*) never been seen within the AFP, and thus making it a very powerful tool (*regarding its already acquired possibilities*) in relation to application development.



[Figure 58]-(MODOQ Application Flex Runtime Environment)

[Figure 59]-(MODOQ Application AIR Runtime Environment)

[Figure 58] and **[Figure 59]** Illustrate the application foundation executed within both of the runtime environments. At this stage of development no actual code has been added to the project library (*apart from the relevant constructer classes*), therefore the application has no content.

8.2.1 - Application Window Manager

The application structure has been created, however further "framework" related code needs to be implemented before any actual "application" specific code can be written.
All application instances will require some sort of windows for tasks such as application login, file selection and any other relevant window related task.
In general, windows are very helpful objects that can be used for many different and relevant tasks and purposes within an application, therefore it is believed that some sort of window manager should be implemented within the framework that can be used throughout further application development and expansions.
A set of default (*custom*) windows will be provided for the application to handle the PS applications "default" tasks (*login and DP file selection*), however an additional and more flexible window manager should be provided, allowing for the support of custom window based controls and components.

A disadvantage to the Flex framework is that it lacks a built in Window Manager, therefore an custom manager has to be either created or included as an external project.
A new release of an open source MDI framework for used with flex application has however recently been released by **[93]-(B Holmes et al, 2009)** and is know as the "FlexMDI" framework.
The FlexMDI framework has a built in and extensible window manager. For this reason the it will be built into the PS application framework, and the window manager will be taken advantage of and used, thus saving additional and unnecessary coding time, as a complete, solid and tested manager is available, that can be modified and extended to meet a specific application requirements.
The FlexMDI window manager is a robust, extensible, open source solution that can be used for **M**ultiple **D**ocument **I**nterface (MDI) based interfaces. "*... FlexMDI is a framework for easily creating MDI interfaces in Adobe Flex. It can be setup and running with FlexMDI in a matter of seconds ...*" the framework is "*... focused heavily allowing developers to easily extend and customise the framework to fit their specific needs*" **[93]-(B Holmes et al, 2009)**.

The FlexMDI window manager offers the standard and well known features of a window manager along with additional features related Flex application development including:

- Draggable, resizable (*from any edge/corner*) windows
- Default functionality for minimise, maximise/restore and close
- Extensive event model on a window manager level
- Externalised effects classes for transitions
- Cascade, tile and "tile plus fill space" window management
- Context menu functionality
- Ability to modify/customise default behaviors
- Construct UI's in wither MXML or AS or both

The FlexMDI window manager can be added to the project structure as either a separate library project and included within the "**MODAQ_LIB**" library projects build path, or by simply including and referencing a compressed ".scw" library file that contains all of the required framework classes. The window manager is compatible for both Flex and AIR applications, thus making it a powerful and very beneficial manager that can and will be included within the development of the PS application framework.

The FlexMDI framework and window manager will be included and used within the PS application framework. A special MDI Canvas "**MDICanvas**" class will be added to the application constructor, that will be further added and used as the internal application container with the application objects being added to this canvas.
Within the **MDICanvas** class, an instance of the FlexMDI window manager will be instantiated as a global variable that will be used throughout an application instance.
The **MDIManger** is responsible for opening (*adding windows to the* **MDICnavas**), closing (*removing windows from the* **MDICanvas**) and controlling and keeping reference of any currently open windows and handling them appropriately.
A simple AS based "Pseudocode" example of adding a "login" window using the FlexMDI window manager would include:

```
private var MODAQ_MDIWindowManager:MDIWindowManager;
private var MODAQ_MDICanvas:MDICanvas;

// artificial constructor class
private function constructor():void
{
        MODAQ_MDICanvas = new MDICanvas();
        MODAQ_MDICanvas.width = 100%;
        MODAQ_MDICanvas.height = 100%;

        MODAQ_MDIWindowManager = new MDIWindowManager();

        MODAQ_MDICanvas.addChild( MODAQ_MDIWindowManager )
        this.addChild( MODAQ_MDICanvas );
}

private function createWindow():void
{
        var newWindow:MDIWindow = new MDIWindow();
        newWindow.title = "Application Login";
        newWindow.width = 400;
        newWindow.height = 300;

        MODAQ_MDICanvas. MODAQ_MDIWindowManager.addWindow( newWindow.height )
}
```

Similar methods exist for other window manager controls (*closing/removing windows and tiling and cascading windows ...*) that can be added and called relevantly.
The above example shows how easy it is to mange windows through the use of the FlexMDI window manager, with all that is required, is to set up and initialise instances of the manager classes (*windows*) and then pass the control over to the **MDIManager** itself.

8.2.2 - Application Event Model

An additional area that is believed to be required, and is "wanted" to be included, is an application event model. The window manager has its own built in event model, however such an event model is believed (due to chosen the framework route) to be very beneficial both enabling and supporting flexible and controllable application instances to be further developed.

There are many reports and literatures existing regarding event based programming with positive results and conclusions, as well as many successful and efficient event based systems existing.
Many development frameworks support and use events in the "background", with in some cases the code itself is separated from the event model, thus programmer is not aware that events are being fired and handled behind the scenes.

At a personal level, although event based languages have been used, a specific and "pure" custom event based and driven development model has never been used within application development, and is therefore an interesting area to explore and obtain experience in.
The fact that Flex is an event based framework and supports event driven application development, makes for the ideal opportunity to follow the event driven application route.

An event represents a change in state and is commonly described as a "notification" in relation actions occurring. A common example of events within programming is "mouse events". Within most frameworks, mouse events are dispatched when a mouse notification occurs, for example on **mouse move**, **mouse down** (*button down*), **mouse up** (*mouse up*), **mouse click** and **mouse double click**, with these notifications being known as events.

Within event based programming, custom events can be created and generally be given a name or ID, and some event models even allow events to be extended, assigning variables/parameters to an event. When an activity occurs a new instance of a pre-defined event is created, and dispatched using a built in framework "Dispatch Event" method. "Event Listeners" are declared, usually within the application root or in a specific "event" class, that listens for the the dispatched events (*adding and removing them as appropriate*).
Event listeners usually consist of a method or routine, that receives and handles a control (*Command*) appropriately when the specific event has been fired that is being listened to.

8.2.3 - Cairngorm Application Framework

Adobe provide a special framework known as the "cairngorm" framework containing an advance AS based event model known as the "**C**airngorm **E**vent **M**odel" (CEM) following a **M**odel-**V**iew-**C**ontroller (MVC) based architectural pattern. The CEM can be used (*rather than having to create a custom event model*) for use with the PS application at this prototypes stage.

The Cairngorm framework is an architectural based framework that was created (*by Adobe*) to try and push AS projects and applicants (and even developers) into a more OO structure and practice. The framework offers a starting point for the development of RIA applications directing (*and focussing*) towards application consistency, reliability, and modularity (OO) during the development of Flex and AS based applications by implementing and using reliable design patterns.

The Cairngorm framework additionally supports applications to be separated by producing an application wrapper that extended modules can be built upon. This is similar to the initial concept that has been designed (*and started to be implemented*) for the PS application, thus justifying its need. The Cairngorm framework could therefore also be used the PS application framework if and where problematic development areas arise.

The inclusion Cairngorm framework is believed to benefit the PS application greatly, and will therefore be incorporated and used within the application development.
The area of the framework that is deemed most relevant (*at this stage*) is the CEM, therefore it is deemed relevant to investigate the CEM in more detail to obtain a better understanding to how it works before it is implemented.
The complete Cairngorm event model consists of several entities. These entities are summarised and explained well by **[94]-(C Giametta, 2007)** as:

- **Value Object:** *The Cairngorm Value Object (VO) acts as a marker interface to help improve readability of code. The VO is nothing more that a simple mapping to a service bean, and is used to help transform incoming object lists to map to Flex components such as charts or DataGrids.*
- **Event:** *The event describes the type of object that will be processed when an event is fired. The event sets an event type to be listened to by the controller. The Cairngorm Event Class is required for any event processing within an application (and is the class of interest within the framework).*
- **Front controller:** *This class controls user actions by handling the event to a specific command. The marriage of these two components allows Flex to detect user gestures and return data to the model.*
- **Command:** *The command class enforces the binding of the event between the Front Controller. The Command is responsible for initiating external calls through the Business Delegate.*
- **Business Delegate:** *The Business Delegate implements the Response Interface that handles data returned as the result of a service call. Best practice for delegates is to match them to the server side method implementation that is used (by following some sort of interface). A delegate can call other services as required however can cause code maintainability issues.*
- **Service Locator:** *The business delegate locates the application level service and passes references of the Commands Result and Fault Handlers.*
- **Model Locator:** *The Model Locator is used to instantiate the applications model and/or models. The locator grants access to data objects that the Command will apply results to for the View layer to assign to Flex components (charts, combos, grids ...).*

[94]-(C Giametta, 2007)

Cairngorm events are therefore not the same as "standard" events, as they follow a different "specialised" and distinct model that can be advantageous to application development if used correctly and consistently from the beginning of the development process, and if not, can (*and have been*) proved to be problematic with the model getting confused resulting in unexpected event handlers being fired.
Cairngorm events are required to be registered within a specialised Cairngorm controller class, and the event therefore needs to be dispatched thorough a specialised Cairngorm event dispatcher instance in order for the associated event commands to execute and handle the correct event.

Below is a simple AS based "Pseudocode" example of dispatching an application "login" event using the Cairngorm Event Model:

```
// username and password would by dynamic variable
// entered into, and obtained from the applications login form
private var username:String = "username";
private var password:string = "password";

// events and event handlers would be set up in separate
// specialised classes within the final application
private var event:LoginEvent;

// artificial constructor class
private function constructor( usr:String, psw:String ):void
{
       event = new LoginEvent( usr, pwd ) ;
       CairngormEventDispatcher.getInstance().dispatchEvent( event );
}
```

If at a later stage a custom Event Model (*and even window manager*) is preferred, and the existing (*external*) frameworks classes being used are deemed irrelevant, custom classes can be further designed and developed at a later stage, however for use wit the PS application, the currently selected and frameworks are believed to be more than efficient.

8.2.4 - Application Structure and Content Implementation

Code and content will now be added to the project library and thus to the two projects, and the PS application will start taking shape.
At this stage of development, the previous design section will be referred to including the created storyboards that will be used as a guide to help build the application structure, layout and content, and to keep the application on track and therefore meeting its set requirements.

The described frameworks (*FlexMDI Window Manager and Cairngorm MVC Event Model*) have been implemented along with a few other relevant classes required by general application instances and tested relevantly.

The inclusion of these classes has been a big step that has taken a lot of time to complete, although this step has not changed the applicant in terms of its functionalities or visual content (*it is still empty and lacking content*), however is a big and important step that is required (*for the framework*) for further implementation of not only the PS application, but for other relevant applications.

With framework foundation in, content can now be added (*implemented*) in relation (*specifically*) for the PS application, thus the application will be built upon the MODAQ framework and further (*helpfully*) meet the overall PS application requirements.

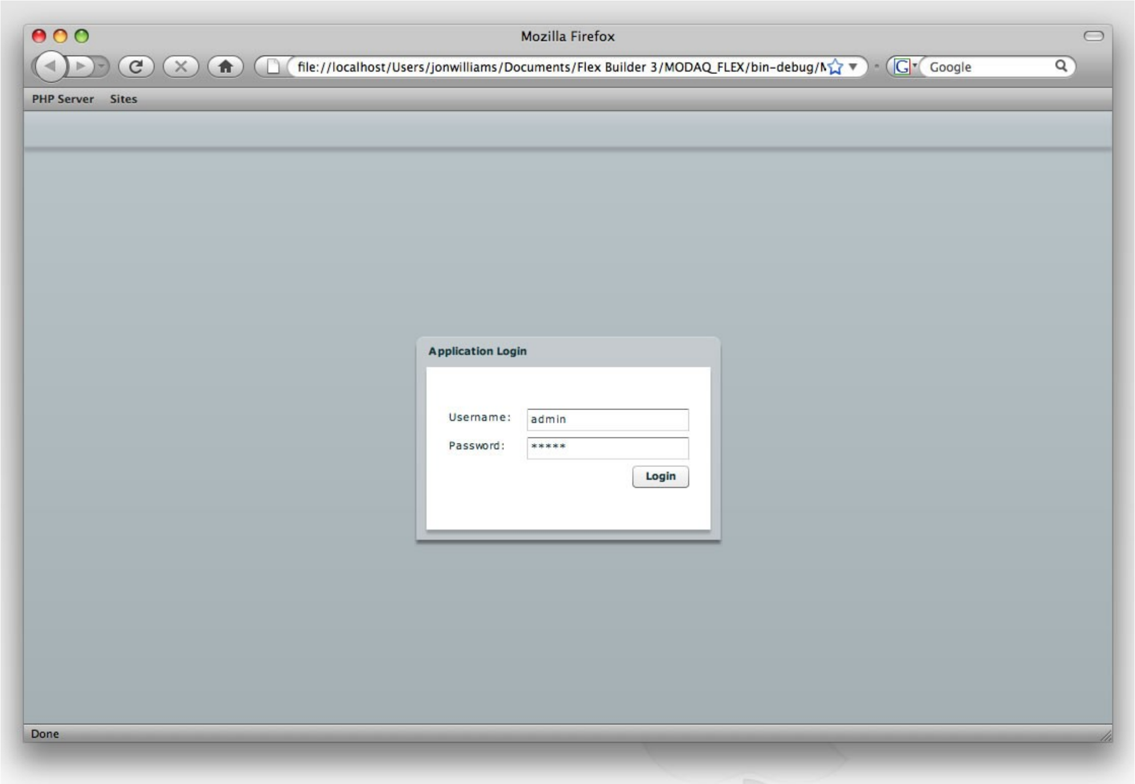
The advantage of having the application framework in place, and further justifying the need and advantage for such a framework, additionally making the time spent implementing and including all of the relevant/required classes worthwhile, is that the reaming application development "should" be a very fast and easy process, simply building components "upwards" from the framework foundations.

8.2.4.1 - Application Login

Referring back to application storyboard designs, the login consists of a simple dialogue containing username and password text inputs and a form submit button.

An interface called "**componentInterface**" has been created that extends the applications main canvas class and includes all the default methods that components within the application will require. All created components will extend this class, thus making for a consistent and organised (*easy to maintain and control)* framework based component construction process.

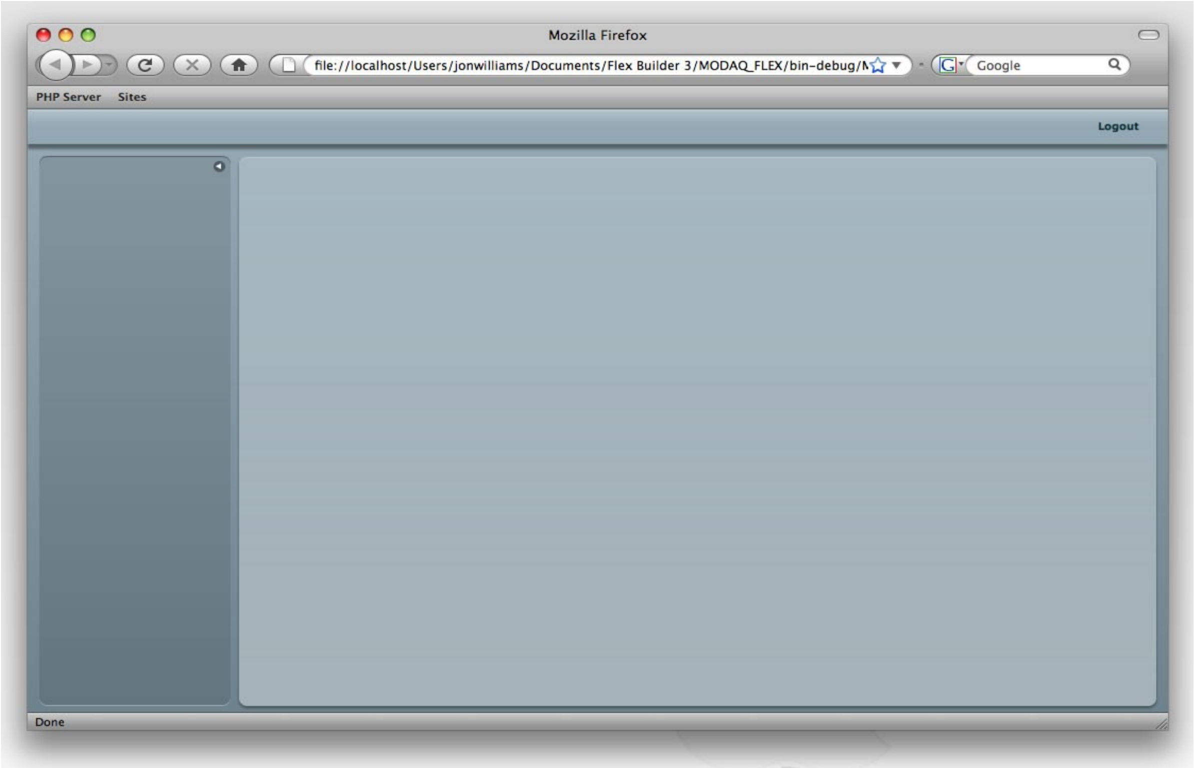
A login component (panel) has been created extending the "**componentInterface**", and is added to a new window instance, that is further added to the applications container canvas using the window manager. Therefore the window is displayed containing the login component to the application canvas where it can be further interacted with.



[Figure 60]-(MODAQ Application Login Panel)

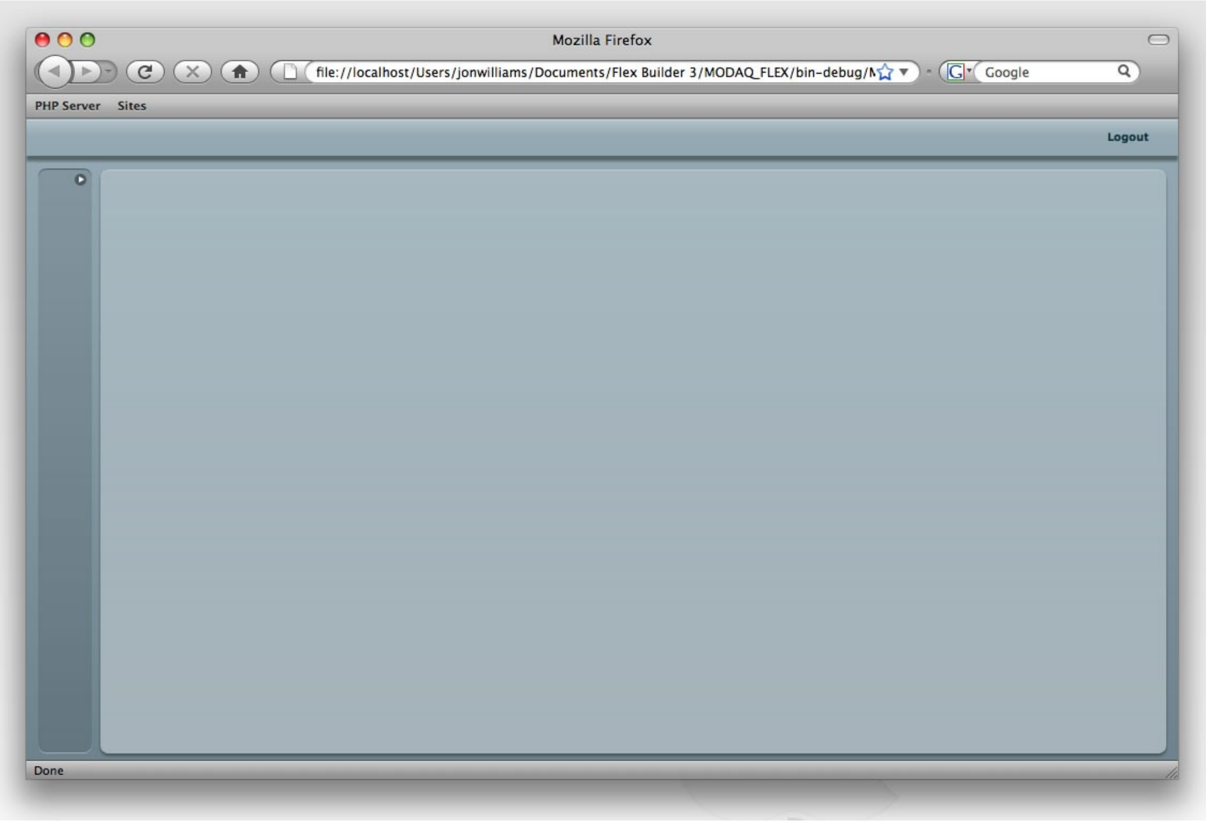
[Figure 60] Illustrates the application login window. When the "submit" button is "clicked" a new "login" event is created and dispatched. The associated event listener will catch the event and handle it appropriately within the relevant Command class.
If the login is successful, the login window is removed (*again using the window manager*) and the user is forwarded and logged into the application.
When a login is successful, a global login boolean is set to **true** until the application is either exited, or the user logs out, where it is reset to **false**, thus keeping track of the applications login status at all times.

8.2.4.2 - Application Layout



[Figure 61]-(MODAQ Application Layer)

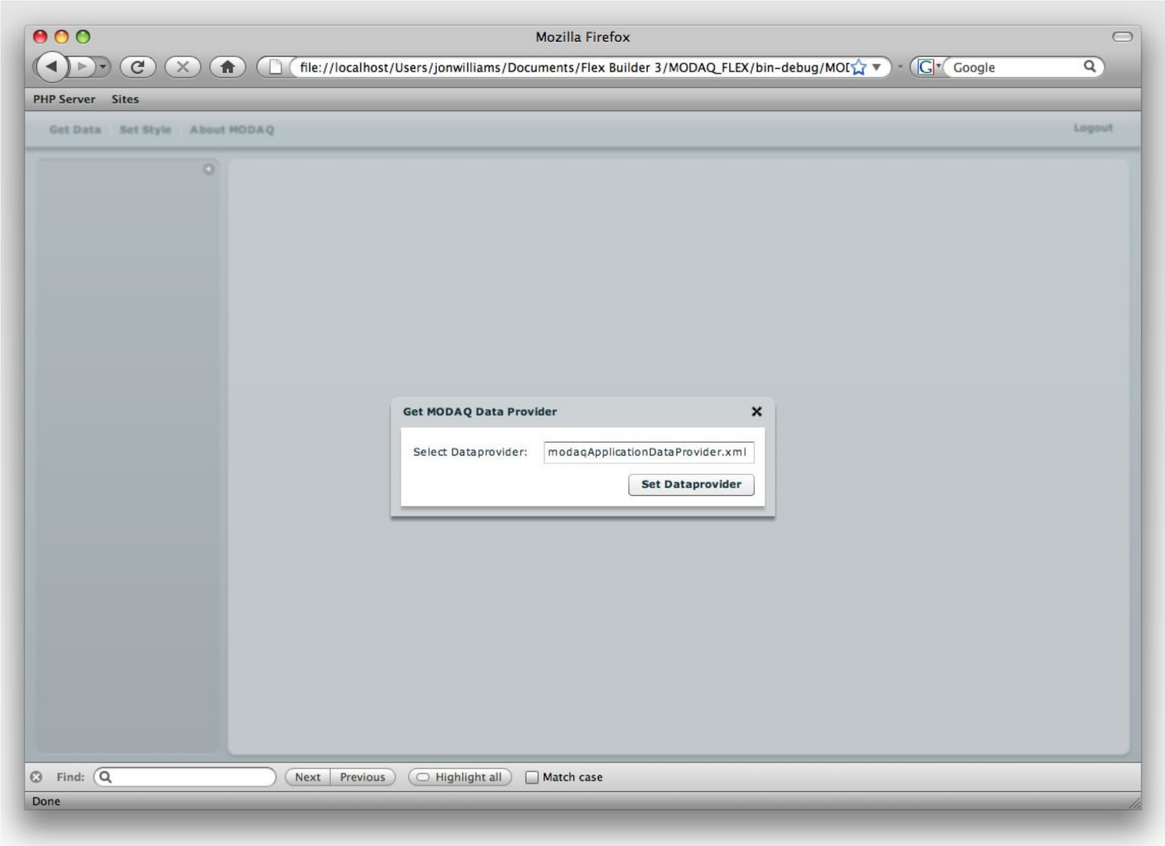
[Figure 61] Illustrates the default layout used within the PS application (*as previously designed*). The structure consists of a vertical layout containing a "docked" application control bar at the top of the container as its first child, in which relevant "higher-level" menu items and controls can be added (*with a control default being "logout"*).
The second child is a horizontal canvas containing two panels (*the menu panel and the perspective panel*). The first panel will parent the "lower-level" applicant perspective menu and the second panel will parent the different application perspectives.



[Figure 62]-(MODAQ Application Layout Optimisation)

During the layout implementation, an obvious problem was noticed (*especially when using smaller screens and resolutions*) with the perspective view real estate.
The perspectives will be the most rich and "busy" views within the application, therefore requiring the most amount of real estate. A solution to this was to allow the first panel (the perspective list container) to be hidden. A tween has been put in place so that when the panel is hidden, it minimises using a smooth animated effect. **[Figure 62]** Illustrates the "hidden" panel.

8.2.4.3 - Data Source Selection



[Figure 63]-(MODAQ Application Data Source Selection)

Once logged in, the user should be able to load data into the application from an external data source (*XML document*). **[Figure 63]** Illustrates the implemented data source selection window. The window is again built upon the window manager in the same way as the login panel, however the "**DataSourceSelection**" component is added to the window instance.
The "**DataSourceSelection**" component consists of a simple file select input box, that does not load the external data source (*at this stage*), but simply saves the path/directory or URL of the source as a variable within the application, that is further used by a data loader class.
The external source can be located anywhere, and as long as it has the relevant "read" rights applied, the document can be loaded into the applicant. When the "Set Data-provider" button is clicked, a "**getData**" event is fired.

8.2.4.4 - Data Provider Creation

An event listener is declared that listens for the "**getData**" event. When the event is dispatched, the event listeners command handles the event by loading the data from the path of the XML document that has been elected by the user and is wanted to be used as the applications DP.

The application checks the file type and content using a custom method to see if it is of the expected type (XML) and contains the expected and valid content, handling the file appropriately.
If the file is invalid, the user is notified appropriately, however if the file is valid and accepted the data from the file (*the XML content*) is loaded and used further as the application DP.

Within Flex, data can be loaded and DP's can be in a few different ways, with the most popular being through the use of either a **remoteObject** call, a **webService** call or a **HTTPService** call. For the PS application, a slightly different method is preferred due to performance issues.
The chosen method to load the XML data is to use a **URLLoader** object and a **URLRequest** object. Due to the large amounts of data that could be potentially loaded, the method needs to be as efficient as possible. Of the recommended methods, the **HTTPService** is most relevant to the PS application, and has been compared (*in terms of loading times*) against the discussed **URLLoader** and **URLRequest** objects, with the latter being noticeably faster.

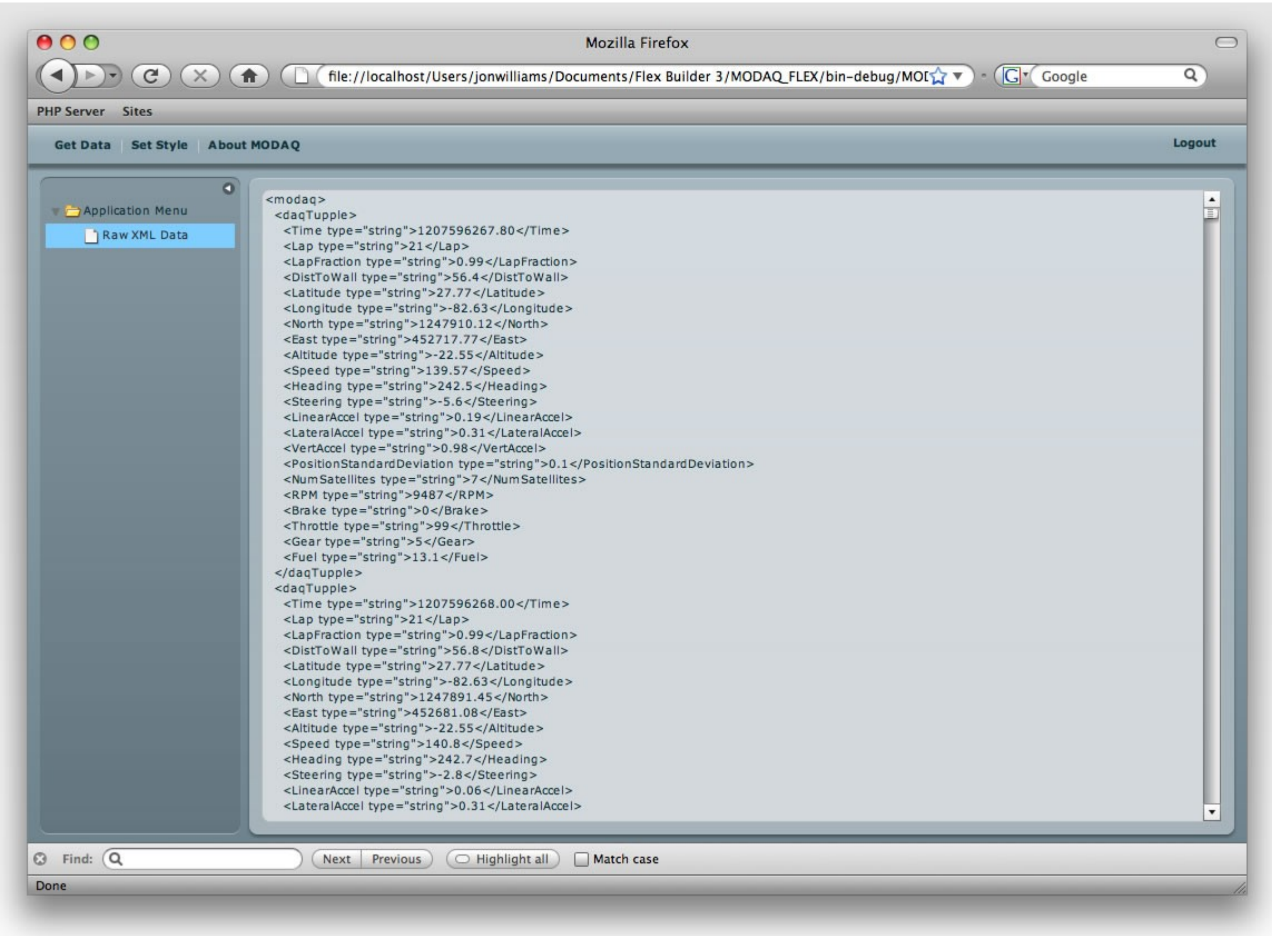
The data loading class creates new instances of both the **URLLoader** and **URLRequest** objects every time the data is loaded.
The **URLRequest** class expects a URL string to be passed, therefore the obtained XML document location is passed into the class, and the **URLLoader** class expects a **URLRequst** to be passed in, therefore the created **URLRequest** (*containing the XML document location*) is passed into the class (*justifying the need for the two objects*).
Relevant event listeners are added to the **URLLoader** that handle all possible results (*both success and failures*) that could occur during the loading process, including:

- A Complete Event (**Event.Complete**) handler (success)
- A IO Error Event (**IOError.IO_ERROR**) handler (failure)
- A Security Error Event (**SecurityErrorEvent.SECURITY_ERROR**) handler (failure)
- A HTTP Status Event (**HTTPStatusEvent.HTTP_STATUS**) handler (success and failiure)

The **URLLoader** is then loaded ("**URLLoader.load()**"), and depending on the result status form the loader determines the event handler that is called.
If the loader fails, the failure will be handled relevantly, depending on the error type that occurred, however, if the result is successful the XML is set as the application DP.

The application contains a global XML typed variable called "**modaqDP**" that is the DP object that the application will be driven from.
If the discussed **URLLoader** result event is successful, the loader will return a complete event (**Event.Complete**) containing the contents of the file as a data object.
The event handler command for the loaders complete event creates new instance of the XML object and sets it equal to the result of the event data object, therefore populating and formatting the object with valid XML data form the selected external XML document that is now used to drive the application, thus resulting in the creation of the application DP.

Once the application DP has been established, the relevant application controls and perspective list (*within the control (menu) panel*) can be enabled allowing the application perspectives to be selected and the data to be analysed.



[Figure 64]-(MODAQ Application Simple Perspective Example)

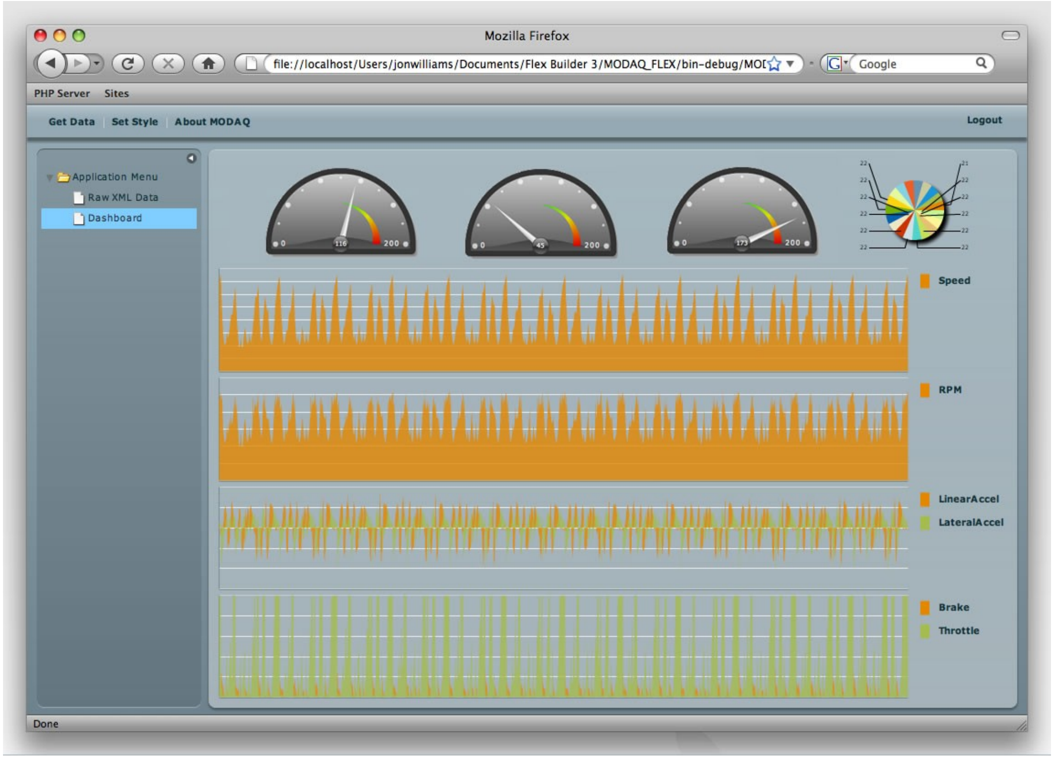
A perspective interface "**perspectiveInterface**" has been created that all application perspectives will extend containing the default methods required for a perspective.

[Figure 64] Illustrates a simple and test perspective "**rawDataPerspective**" that has been created extending the "**perspectiveInterface**" class as a test and to prove the applications concept.

An additional interface has been created "**componentInterface**" that is relevant to an individual perspective instance. A perspective can contain multiple components and each component must extend this interface thus keeping them in sync with one-another and the DP that is driving the perspective in relation to filter or data control changes.
The "**rawDataPerspective**" is a very simple perspective that contains a "**textArea**" component that extends the "**componentInterface**" interface and outputting the contents of the application DP (XML) object in a string (*text*) format.

The perspective name is listed within the perspective list tree menu control, which when "clicked" dispatches an event "**getRawDataPerspective**". The Command for this event creates a new instance of the "**rawDataPerspective**" class and adds it to the perspective panel where it is rendered as shown in **[Figure 64]**. Within the figure, the output of the DP (*the XML content*) can be seen within the perspective component "**textArea**".

The discussed process(es) represent the complete (*basic*) life-cycle of the PS system application and application framework.
The application can be logged into and a DP can be set using an external XML document. Once the DP has been set, it is used to drive custom created application perspectives and components.
Depending on the application, both different and additional higher and lower level menus and controls can be added as well as different perspectives and components.
As long as the perspectives extend the "**perspectiveInterface**" and the components within the perspectives extend the "**componentInterface**" class, they can be added, removed and used in a standardised and reliable manner.



[Figure 65]-(MODAQ Application Analytical Perspective)

[Figure 65] Illustrates a basic perspective that has been created as a "first-step" perspective to be used and expanded upon as a potential "analytical" perspective within the PS. What the eprspective does howver illustrate is the rendering of the acquired data in a visual format, thus highlighting the potential of the application in relation to further perspective improvements. The perspective has been created using the same creation process as previously described, however the perspective includes more components consisting of built in Flex charts and custom gauge components enabling a more "visual" tool for representing, analysing and visualising the "back-end" systems acquired data.

The benefit of the application framework is that perspective views can be created, added, removed and modified to and from an application at any time. Therefore custom and specialised perspectives containing custom visualisation components can be created and tailored towards specific application requirements, as well as existing components being modified or removed from perspectives if and when required.

8.2.5 - problematic implementation areas

The implementation process was very smooth with only a few problems occurring.
The main problem was that the standard Flex charts "struggled" to render the complete XML DP containing 4688 nodes (*especially when there are multiple graphs within a perspective all rendering the same amount of data points*). The solution to this problem was to simply reduce the size of the DP to an amount that the charts can handle reasonably (*around 1000 data points*).
This problem was expected to a certain extent, as rendering such a large amounts of data within the AFP is going to be a very heavy task, and was a worry when the "risk" was made to take the AFP route. There are solutions that could be implemented to solve these rendering problems (*including various caching and quantising techniques*) thus the problem is not considered to be a "show-stopper", however, in relation to the PS application in its prototype stage, it is deemed satisfactory to simply filter the data and document the problem area for further and future investigation.

An additional problem was that the applications performance slowed down dramatically after loading data a few times (without closing the application). It was noticed that the reason for this was that the application was using a lot of system memory, and when the appklication was shut down the memory was freed.
The code was proof read and there was no "obvious" areas within the code that would cause such memory leaks (*all variables, and objects were being re-used*), and the code was cleaning up after itself (*closing connections and removing event listeners as and where appropriate*).
Leak appeared to get worse every time data was loaded, so it was assumed that wither the data loader or the XML object itself was the culprit for the leak. Flex has a built in profiler that is not great, but does the job. The profiler confirmed that it was the XML object that was leaking. Evert time data was loaded into the object, the old data was not being disposed of, thus increasing the object size each time the data is loaded (*until the application or system os out of memory and crashes*).
The CML object was declared once within the application code, and was being re-initialised on each load, however there was something still referencing the object, thus not freeing the previous data loaded for **G**arbage **C**ollection (GC) when new data is loaded (*resulting in the memory increase*).
It is still not known what was referencing the object, and after finding a lot of bug reports within the Adobe Flex Builder "bug" ticket system in relation to XML objects and memory leaks, it was assumed irrelevant to peruse, and deemed as a framework bug.
A solution to the problem was however found. The solution consisted of checking if both the XML object and the XML object children are "**null**", and if not set them equal to "**null**" before any new data is loaded. This solution worked (*although a bit "flakey"*) removing all of the object references and feeing it for GC, and the application memory is now stable.

Additional (*smaller*) problems that arose included simple Flex setup issues that were quickly resolved, as well as a strange **URLLaoder** failure, being handled in the discussed **HTTPStatusEvent** handler, which was to do with the network settings of the development box being used, and which was again resolved quickly and easily.

9 - Testing and Validation Chapter

This aim of this chapter it to test and document the final system artifact. A suitable testing framework and plan will be identified and followed with areas and sub-areas of the PS being tested relevantly and the results from each stage being logged and evaluated accordingly.
This will determine the success of the overall system (*if it works, if it meets its requirements and overall what has been achieved during the systems development*).

As the PS result is separated into two separate sections, the testing process can also be separated into both "back-end" system and "front-end" application testing areas.

Testing is a very important phase of any project, as both the individual stages during a systems development process and the complete (*final*) system as a whole is tested thoroughly highlighting any problematic areas (*bugs*) that require fixing and showing the projects overall success.

This statement, and many other testing sources are contradicted to a certain extent in relation to the PS. The importance of testing is agreed with, however as the main PS solution is a "theoretical" based system solution, a "physical" artifact does not exist that can be tested (*at least not at a practical level*). As discussed, a prototype "front-end" application **has** been implemented that **can** be tested at a "practical" level, and for this reason this is where the content of this chapter will (*mainly*) focus, however the aim of the prototype application (*at this stage*) is as its name suggests, to be a prototype, and to simply act as a POC demonstrating what **could** be achieved, justify the need of such a system that can be taken and used as a foundation for further system development (as and if necessary).

Testing of the PS (*both "back-end" and "front-end"*)is not deemed unnecessary and will however be (*and has been*) included within the development process, although to a much more "laid back" manner in comparison to what would be needed if the requirements of the PS were for an actual "real" release rather than a POC.

The PS has been tested continuously throughout the development process, with problems that existed being found early and dealt with (*solutions were applied*) before the problems became more severe which could "potentially" prove both costly and time consuming to correct.

Before any test plans have been created, and any actual testing has been completed, testing based frameworks are required to be investigated in relation to what is the best method for the PS testing, thus enabling the creation of better, more solid test plans.

Testing is critical for the success of any project development. It is as discussed used to see and check if a system is doing and performing expected. The aim of this testing process is to find errors or misinterpretations within the PS solution (*artefact*) both "back-end" and "front-end", and fix them accordingly. A general testing cycle usually consists of three main goals:

- To asses the extent of a systems functionality
- To asses the effect of a system has on a user
- To identify any specific problems with an system.

As discussed within the design chapter, regarding HCI considerations, an applications usability plays an important role depending on its success therefore should be tested accordingly.
A system functionality must meet its specified requirements, providing appropriate functionality to complete a task in an expected manner. Testing at this level may involve measuring a user's performance to perform a task assessing the effectiveness of the application in supporting the user.
It is important to test an applications design (*the impact that it has on the user*), including the user's opinions, how easy it is to learn and understand, and its overall usability identifying any areas that may overload the user.
The testing for specific problems identifies any unexpected results or confusion amongst users concerning both the usability and functionality of the design.

[x]-(Dix et al, 1993)

For a complete and thorough test of a system both black box and white box testing is required. Black box and white box testing are two separate types of testing methods that give system testers a clearer view on the quality of an overall system.

Black box testing is also sometimes referred to as "functional testing". During black box testing the internal workings of a system are not visible or known to the tester therefore a novice user complete the tests.
The tester should only know the inputs that they are inputting into a system, and know the outcome that is expected from these inputs. The tester is additionally not concerned (*does not care*) how a system produces its outcomes.
For a black box test the tester does not need to have any knowledge regarding a system apart from some general and relevant specifications and its overall purpose.

White box testing is sometimes referred to as "glass box" or "clear box" testing. During white box testing the tester has access to, and understands the internal workings of system thus any existing system code must be readable, or any system foundations and functionality's must be understandable. The tester uses this knowledge to select the data being tested, and uses it to examine the outputs of the system, therefore this test is only valid if the tester knows what the expected output or result form the system should be.

More in depth investigation into testing methods that could be used to create a test plan for the PS (mainly aimed towards the testing of created application) can be found in **[Appendix 17]-(Possible Testing Methods)** including unit testing, validation testing, verification testing, security testing, usability testing, integration testing, and user acceptance testing.

9.1 - Selected Testing Method

A selection of testing methods relating to some of the above areas that are believed suitable for the testing of (*mainly*) the "front-end" application will be chosen. The selected methods will be used collectively to create a test plan that will be used for the overall system testing phase.

Unit testing **has** be carried out during the system implementation. As each individual element (unit) was implemented they have been tested and documented accordingly to see if it functions correctly.
The problem with unit testing can occur (*in some cases*) when the code is reused (*being common practice within OOP*). Classes can be tested and there functionality's assumed correct, however, when additional functionality from different components are added, although again tested, can cause problems within the components reusing the same code (*interface*) with slightly differing or additional functionality's.
Unit testing is a good method for error testing during implementation although cannot guarantee that all functions and methods are tested accordingly and working in one "conjunction".

Integration testing will also be used. It will combine and test all the functions and methods that were tested during the unit testing, however using the completed and fully functioning system focussing on the different areas and modules of the system working together as a whole.
Further validation testing will be incorporated into the integration testing looking at the relevant user feedback and error handling (*validation*) methods and properties in more detail. The integration testing will test all the functions within the system testing phase, and will therefore "potentially" be very detailed.

Requirements testing will be used to test if the system meets all of its requirements. A testable specification will be produced identifying the systems major requirements, and then an in depth "walkthrough" approach will be performed test test and check that each of the individual requirements are met, and if not, why not..

The compatibility of the system will be tested in relation to the "font-end" application using some of the most commonly used and supported desktop and web-browsers, and using the relevant screen resolutions to check if the application is functioning correctly within each scenario.

Usability testing will be completed using both selected "real" people and simulated people (*as it is faster to achieve simulated testing results, which at this "prototype" level of development is believed satisfactory and efficient*), further using and interacting with the system to complete set tasks. A user (*from the identified user class*) will be asked to take part in the usability testing. The users will be asked to perform relevant tasks to there pacific user class and they will be asked to fill out a usability schema and also observed on how easy and efficient the task was to perform.
The user will also be asked general questions regarding the overall navigation, enjoyment, easiness, and efficiency of the use of the system.

Before each test plan is undertaken, the area of the system being tested will be summarised, providing a better understanding of the requirements of the system area being tested and therefore additionally the goals within the test plan.
This summary can also be referred back to during the testing process as a reminder of what has been and is required to be tested, thus additionally keeping the testing phase on the right track (*heading in the correct direction*).

9.2 - Back-End System Test Plan

The main project goal of the project was to investigate and produce a theoretical system based on the advanced subject of tracking and acquiring (*both locational and statistical*) data from a high speed object (*motorcycle*) traveling within a specific environment (*a racetrack*).
The established "back-end" system solution and artifact is however theory based, as due to mainly cost and resource constraints the solution could not be implemented at a practical level (*at this prototype stage*), therefore further testing of the solution at a practical level is unfortunately not possible.

Due to the theoretical based solution of the "back-end" system, testing is limited, however a relevant test plan will be created and used to test the outcomes of the solution at a "theoretical" level in relation to system accuracy, effectiveness, learning outcomes and overall what has been achieved during the creation of the final solution, and if the solution meets the initial system requirements and if so, to what level.

A "back-end" application test plan has been created and can be found in **[Appendix X]-(Back-End System Test Plan)**

9.2.1 - Back-End System requirements summary

The "back-end" system is to acquire data from a fast moving mobile object (*a motorcycle traveling at speed*) as it travels around a set boundary and within a set environment (*a racetrack*).
The acquired data is to be transmitted to a CBS where it can be further handled and used for data analysis.
The system should allow both locational and statistical information to be acquired from the motorcycle using relevant and efficient methods. The locational information is the actual (exact as possible) position (location in relaxation to the racetrack) of the motorcycle at specific times during the lap. Such measurements could be in the form of coordinates (**-X**, **-Y** and even **-Z** coordinates).
additional statistical measurements should be additionally acquired. The amount of additional measurements required will be dynamic allowing a system to be application specific thus acquiring **x** number of measurements of **x** type. Such statistical data measurements could include **Speed**, **RPM**, **Acceleration**, **Braking**, **Throttle Position**, **Fuel Level** and **Current Gear...** An additional measurement that is required and believed beneficial to the data analysis process is the lean angle of a motorcycle during each section of a racetrack.
technologies and HW shroud be realistic in terms of price and general possibilities (*performance, maintainability and installation*). Additionally any hardware or devices required to be installed on the motorcycle must not affect its setup, handling or performance in ant noticeable way, thus should ideally be as small, light and compact as physically possible.

All of the acquired data should be transmitted (*together as one easy to mange file and format*) using a relevant mechanism efficiently and reliably in a relevant data format type to a CBS where is can be easily dealt with and further handled.

9.2.2 – Back-End System Test Plan Execution

The test plan has been executed (*at different times during the investigation cycle*) and due to the extent and repetition of some of the results, they have been simply summarised below highlighting the main results founds within each:

9.2.2.1 - Back-End System Unit testing

The unit testing was an informal stage and took place during the system implementation. As each object, component, and individual method was implemented it was tested and documented relevantly. Below is a summary of the documents taken during the Unit Testing cycle:

During the solution investigation process many area were deemed possible options to solutions and thus were further investigated and compared relevantly.
Different types of LBS and LBS architectures were investigated along with further tracking system solutions and a great understanding was obtained in relation to how the fundamentals of how such systems operate and are relevant to the context of the PS.
Two possible tracking methods existed (*that were deemed the most relevant*) including sensor based SBS and GPS. Theses systems were investigated and “theoretically” tested with many questions being asked and reasons justified regarding the systems in comparison to the PS, and its requirements. Existing systems within each domain were “theoretically” tested and compared in relation of strengths and weaknesses and primarily how they can be adopted and potentially used within the PS solution.
Of the two, the selected and most relevant solution for use was GPS, which was therefore taken and tested further to try to find any problematic areas of the technology within the set domain and thus possible solutions as and where appropriate.
It was established the the used of GPS alone was not accurate enough when being used to track such high speed moving objects. Another problematic area discovered was that of GPS satellite interference, blockage and reflection that could be (*and is likely to*) be caused within the PS environment (*a racetrack*) where there could (*potentially*) be many obstructions including buildings (grandstands), bridge and fences ... A solution found to both of these GPS problematic ares (*as well as other areas*) is the implementation and inclusion of an additional and advance DGPS system, consisting of additional GPS receivers located relevantly around a racetrack at fixed locations, calculating and applying any corrections (*offsets*) than may exist, thus making for more accurate location measurement acquisition, additionally supplying more satellite coverage so accurate corrections can be calculated and applied even when the mobile (*motorcycles*) receiver has poor visibility.
A further solution found and tested improving both system accuracy and efficiency was that of a telemetric system. A solution was required regarding the synchronisation and transmission of the acquired data to the CBS, is an accurate and reliable method. The solution to this was the used of an advance telemetric system installed and working alongside the discussed (D)GPS system.
Further “back-end” system fundamentals have been investigated with no real problematic areas (misunderstandings) arising, with a potential solution being established that can be taken and tested further.
To acquire data from the motorcycle, specialised sensors and systems (*mechanisms*) will be used. Outstanding elements at this stage in relation to the systems completeness is an actual GPS receiver and DAQ device that is to be installed on the motorcycle and used with the system. Both a physical DAQ device and GPS receiver is required to handle the acquired data from various sensor and mechanisms in place, thus relevant devices and receivers were further investigated. Due to the amount of devices available, an in-depth selection process was completed, resulting in a selected few (*believed most relevant for use*) with these few being further investigated and compared in relation to the PS requirements. Most DAQ devices have advanced GPS receivers built in, being a preferred solution as only one device has to therefore be location/positioned on the motorcycle (*having limited space*) opposed to two.
A device is required that can handle multiple data inputs (*channels*) from external sources (*sensors*) and transmit all inputs together. Of all the reviewed devices the best solution believed relevant for used was the Nology G-Dyno Plus device meeting all of the system requirements to a more than satisfactory level, and further being the most relevant in comparison to all the other devices investigated.

Overall the Unit test was a theoretical success with all requirements being met at the end of completion, although a long and exhaustive process was involved to establish a complete and final “back-end” solution. A lot of problematic areas and hurdles were identified during the process, and areas were dealt with and investigated further resulting in solutions being found (*as and when identified*) for all as an where required thus enabling a “complete” system solution for the “back-end” DAQ system.

9.2.2.2 - Back-End System Integration/Function Testing

The integration testing used the documentation produced within the Unit Testing and followed each step in a walkthrough manner, therefore testing the same elements but at the end of implementation when the application is complete and all functions are working together as a whole. This was an in-depth detailed stage looking into the individual aspects and functions contained within each component of the application.

The results form this stage are very good with no additional problematic areas being found. This stage is not as accurate at a "theoretical" level opposed to a "practical" testing level, however the process did benefit from, and provide a chance to re-iterate the chosen solution and reasons of the proposed solution in relation to (*comparison*) against other "possible" solutions, further justifying the selected solutions reasons and correctness from the Unit Tests and the complete solution investigation.

9.2.2.3 - Back-End System Requirements Testing

Further system requirements testing has been completed consisting of the testing initially set requirements. A rough "pencil" based specification draft containing and identifying the "major" system goals and requirements has been created, and a system walkthrough has been performed checking and testing that all of the documented requirements were met and covered within the theoretical system solution.

The results from the requirements testing show very positive with all of the initially set system requirements being met.
Again (*like the Integration/Function testing stage*) this stage is not as accurate at a theoretical level opposed to a practical testing level, although again helped to justify the chosen solution and further "double" check that all system requirements were covered.

An area that was identified as "missing" was the accusation of the motorcycles lean angles. It is still unsure as to why and how this important area was not covered (*overlooked*), and is good that it was found before any further investigation continued, otherwise resulting in the ares to "potentially" be forgotten about, and thus an important system requirement being missed. This measurement was further investigated and a solution was found.
Relevant sensors that could be used were investigated that could be used to measure such angle, however during this investigation it was understood that the actual (*similar*) measurements that such sensors use to calculate the angle are already acquired within the system solution, therefore it was deemed best practice to create an algorithm using these acquired measurements, that can be used client-side, thus saving on additional calculation and transmission times of the "back-end" system that are not required, and the lean angle being calculated appropriately for use during further data analysis.

9.3 - Front-End Application Test Plan

The testing of the "front-end" application is in comparison to the "back-end" a lot easier and more straightforward as an actual "physical" artifact has been created at a practical level (opposed to at a theoretical level) that can therefore be "physically" tested. Additionally, as the solution is a UI based artifact, testing can follow a standard "application" based approach (test plan) that is known and familiar with at a personal level

A "front-end" application test plan has been created and can be found in **[Appendix X]-(Front-end Application Test Plan).**

Not only the implemented flex application has been tested, but also the "intermediate" step between then "back-end" and "front-and" consisting of the PHP script that handles conversion and storage (within a DB) of the acquired ".csv" from the motorcycles DAQ device to a valid XML Document, will also be tested within this section.

9.3.1 - Front-End Application Requirements Summary

The requirements of the "front-end" application have both changed and grown dramatically during the project. These additional requirements will be added to the initial application requirements and included in the testing phase, allowing a more reliable test result.

The main requirements of the "front-end" application is to run in both a desktop and and web based runtime environment in an easy and accessible method. A user should be able to login to the application using a simple "username" and "password" login mechanism.

Once logged into the application, the user should be able to use the application to navigate to (*located*) and load and external XML document into the application that will further be used as the application DP.

Before the XML Document can be loaded into the application, it has to be available. the XML document is created using an “intermediate” process uses the acquired data from the “back-end” system, storing (*inserting*) it into a table within a (MySQL) DB and further creating the XML document from the DB table, all using an advance PHP script.
This script and process needs to be additionally tested within this process.

A simple interface supporting PHP (thus requiring a PHP server either local or web-based) should be provided a user can use to navigate to (locate) the data file (“.csv”) that was transferred from the motorcycles DAQ device containing the acquired measurements that a to be analysed. Once the file has been loaded, the PHP script should created a BD table and load the data into the created table, and further export the data into a valid XML document consisting of nodes constructed form the relevant table columns and rows.

Once the XML document has been loaded into the Flex application, the user should be able to “click” (*select*) a perspective form a list (*menu*) containing the available perspectives relevant to a particular application. When the perceptive list item is “clicked” the relevant perspective should be rendered within the application and the data should either be visible or at least accessible (*depending on the perspective component(s) in use*).

The user should additionally be able to logout of the application or exit the application at any time during use.

9.3.2 – Front-End Application Test Plan Execution

The test plan has been executed (*at different times during the development cycle*) and due to the extent and repetition of some of the results, they have been simply summarised below highlighting the main results founds within each:

9.3.2.1 – Front-End Application Unit Testing

The unit testing was an informal stage and took place during the application implementation. As each object, component, and individual method was implemented it was tested and documented relevantly. Below is a summary of the documents taken during the Unit Testing cycle:

During the creation of the PHP script initial problems were experienced related to the initial installation and setting up of the required environments and SW of both PHP and MySQL (proving a tedious task) additionally trying to get them to communicate with one another. However, once these hurdles were jumps, both worked and communicated error free throughout the rest of the development.
During the implementation of the PHP script, general errors and problems arose, however nothing “out of the ordinary”, and nothing that was not caused by wither a simple typo or of additional low-level cause that were easy and fast to put right.
MySQL provide an administrator that proved both very helpful but additionally confusing within areas. Due to the simplicity of both the DB interaction and SQL required during development it was decided (*and was proven more efficient*) to simply test all DB related tasks from the command line and eliminate the provided administrator.
The connection form the PHP script to the MySQL DB was initially proving problematic and unreliable as is was failing in an “unpredictable” on-and-off manner, with the cause of this error being assumed to be on the MySQL side. A solution to this error was to simply implement handler for the error into the script. This error disappeared after a while, and it is still not known what the cause was (*and its not deemed necessary to find out at this prototype stage*).
The generation of the XML documents was a time consuming task to complete each stage and node creation had to be tested, as the document needs to follow a strict format to be of any use, but over all was a very smooth process with no problems occurring.
All cases and handlers within the script were tested accordingly, with special cases and traces (*“echo” within PHP*) built into the code to “fake” each case and test accordingly.
No real problematic errors occurred dung the creation of the systems “intermediate” file handling and DP creation process, with all smaller problems that cropped up being solved immediately before further testing was continued.

The Flex application implementation was a complex development stage that due to the changed application requirement aimed towards a more "framework" based result added additional complexities to the system design and implementation approach. With this change in mind, the application implementation went very well worth only a few problems occurring.
One problem that did occur, (*and was not solved*) is that of performance within the AFP when handling and rendering large quantities of data. The problem was investigated to a certain extend, and was mainly narrowed down to the basic data handling of the ADF, and also the performance of the built in Flex components (*in this case the Areas Charts that are being used*). A solution at the time of testing was to reduce the sample data to one single lap (*the sample contains **x** amount of lap measurements thus is **x** times bigger*) which was deemed satisfactory at this stage of development.
Solutions to the problem (*at a later stage*) could be to implement advance caching and quantising techniques and classes that handle and render the data in a way the the AFP can handle, and/or create custom graph/charting components that are more efficient than the ones built in.
A memory leak was also found on the data load resulting in the application memory to increase dramatically (*with the amount being relevant to the DP size*) every time a DP was loaded thus in turn impacting on both application and system performance. The cause of this was that not all of the references to the XML Object were being removes therefore the object was not being disposed of. although being a "heavy" problem this fix was (*once understood*) fast and effective.
Another problem was a strange error on the **URLLoader** that is used to call load the DP. This problem took up a lot of time additionally causing a lot of stress to solve. the **URLLoader** was returning a failing **HTTPServiceEvent** when trying to load the external document. All standard areas were checked (*file and folder permissions and the correctness of the paths used*) with no obvious element appearing incorrect. Forums an other help sources were scanned, and the problem was eventually solved being related to the firewall settings on my test web-server where the test DP was situated.
Flex has a built in and very good debugger, that made the error identification and solution (*debugging*) process very efficient. A lot of smaller errors were found (*mainly due to programming mistakes and practices*) and resolved during the implementation process.

Overall the implementation process was very smooth with only a few (*a lot less than expected*) errors and problematic areas arising. Most errors were very simple and easy to resolve with others (*as discussed*) being slightly more in-depth. All errors identified were either solved or a simple solution (*work-around*) being justified for use at this prototype level was applied (*For example the workaround for the discussed large data handling and rendering*).

9.3.2.2 – Front-End Application Integration/Function testing

The integration testing used the documentation produced within the unit testing and followed each step in a walkthrough manner, therefore testing the same elements but at the end of implementation when the application is complete and all functions are working together as a whole. This is a detailed stage looking into the individual aspects and functions contained within each component of the application.

The results of the integration testing are overall good very with only a few additional errors being found additional to (*were not found within*) the Unit Test.
The additional problems found were mainly within the components that extend the perspective interface. This was in a way expected, during the implementation process it was noticed that some components additionally require access to objects within other components and interfaces that are not available to theses components while they are extending the perspective interface. Flex does not support the extension of multiple classes, therefore a "dirty-fix" was implemented, that seemed to solve the problem during the implementation testing, however during the integration/function testing process has re-occurred,thus highlighting the rule of not taking programming shortcuts and to do the job properly first time round. This issue has been fixed with "proper" fix being implemented and is now working well.

A custom timeline component was created but was obviously not tested correctly during Unit Testing. The component initialised correctly, however when it was moved or resized (*the data was filtered*) it caused the application to blow, with a lot of runtime errors occurring. The problem was debugged, but a "solid" not been found. This same issue was also noticed on the custom Gauge components that were included with one of the perspectives, therefore this issue was postponed for further investigation.
Additional smaller problems were found including cases were not typed correctly thus causing runtime errors during object conversations and castings, as well as a lot of "null" objects being used, additionally causing the application to blow (*mainly during data loading and interaction*). These errors were further debugged and fixed relatively easily and efficiently (*again suing the Flex debugger*).

All of the problems found during the integration testing were fixed apart from the discussed "custom" component support (*the time line and gauge components*) error, that has not been fixed due to time issues. The debugging process regarding this component has already soaked up too much time , and is therefore deemed irrelevant to fix at this prototype level as it "could" push the project over its deadline.
Overall the results gathered from the integration testing were good showing a strong functioning application successfully meeting its POC requirements even with the remaining (f*ew*) existing problems that can be investigated at a later date.

9.3.2.3 – Front-End Application Requirements Testing

Further application requirements testing has been completed consisting of the testing of all of the initially set, and additionally found/included application requirements. A rough "pencil" based specification draft containing and identifying all of the "major" application gaols and requirements (*as a whole*) was created, and a application walkthrough was performed checking and testing that all of the documented requirements were met.

The results from the requirements testing show very positive with all of the fundamental requirements being met.
Problems (*requirements that have not need "fully" me*t) were found, however they are problems that are already know and have been justified as to why they have not been implemented to a full level.
These areas include the application Log-in Component and the additional application specific components.
The Log-in panel have a default set log-in (*usn: "admin" pwd: "admin"*) opposed to having user and application specific log-in details stored in an external and secure (*encrypted*) source, therefore the application does not meet the requirement of supporting a "fully functioning" logging mechanism. This is however justified as a log-in template class has been implemented within the application supporting and handling all of the necessary events required for to log-in, and can be used as a "real" log-in mechanism would be, in the sense that a user-name and password can be entered and further accepted or rejected, thus demonstrating the component as it would be in a real situation, however not supporting different user accounts, thus not meeting its requirements, although being justified and accepted for use with the PS as it demonstrates what can be achieved, being the "fundamental" requirement of the application.
The specific "custom" component however has a stronger "worsening" effect on the application as a whole. the most important and beneficial elements of the application are the perspective and visualisation components. The problem restricts the support of "custom" visualisation components to used within perspectives thus limiting the application to a certain extent. default "standard" components can be included (*for example charts, graphs, grids ...*) and used problem free, however whenever a "custom" component is added that interacts with the data (*applies filters*) run-time errors occur. The problem is only believed to be something small, however it is a small problem that has so far not been resolved. Due to time constraints a solution has had to be postponed until after the project deadline due to other areas of the project requiring attention and completion. Although not meeting the application of being able to create interactive "custom" components, the fundamental application requirements have been met in relation that the perspectives can still be used (*with standard components*) to plot and render the acquired data relevantly and in an analytical way, however can just not be (*at this*) stage interacted with.

After a break from this problem, and some additional thought regarding what could be causing the problem, it was thought strange that custom components are not supported however standard components are, as both component types expected from the same interface, therefore having to be a problem within the custom component itself. A POC was created to justify this thinking and a simple custom component was created consisting of a canvas and graphics tool that drew lines to the canvas in a horizontal fashion based on the acquired "Speed" measurement form the XML document (*thus representing a simple line graph*). The component was wired up and successfully worked, therefore pointing the finger at the data filtering mechanisms. The data filtering mechanisms were further removed from the problematic components, and the problem was sovled, thus the error is caused by bad coding practices (and design work) within the specific custom components that were created.
The expansion of the application to the application framework, solves this problem (*when looked at from a different angle*) as the main application framework is to provide a platform that perspectives and components can be built from and provided a DP (*along with access to other classes and methods*)

which it does successfully, therefore new more reliable components can be created (*or the existing components could be repaired*) and simply added back into the framework.

9.3.2.4 – Front-End Application Compatibility Testing

As the application is both web and desktop based (depending on the sleeted runtime) it has been tested in both relevant we-browsers and on different OS's.
To run the Desktop based file, Adobe AIR needs to be installed on a system. The AIR client only (currently) supports **Windows**, **MocOS** and some other Linux based OS's. The PS application desktop runtime has been successfully tested on both Windows and MacOS boxes with no problems occurring. The application ran as expected on both desktops smoothly and efficiently.
The applications web-based runtime has been further tested in different browsers (*again on both OS's*) including **Internet Explorer**, **Mozilla Firefox**, **Safari** and **Google Chrome** again all producing good reliable and error free results.

The application worked well and proved to be very compatible on all tested desktop and web-based platforms. The layout remained consistent the components behaved as expected throughout.
Additionally the "intermediate" stage was tested on all platforms, placing and making accessible both the PHP script and DB online, again proving successful. All database interaction and dynamic user interaction worked well with no problems found with any of the PHP script, DB interactivity or UI.

The application was also tested in different screen resolutions. The main use of the application, and what the application was designed for are desktop PC's and laptops. With this in mind the following screen resolutions were tested **800 x 600**, **900 x 600**, **1024 x 768**, **1280 x 1024**, **1600 x 1200** and **1920 x 1200**.

The application was usable with all screen resolutions as all components and containers have been sized using percentage values allowing them to be resized accordingly adapting to the relevant screen resolutions.
The application is better for use and more visually pleasing using larger resolutions, especially when a complex dashboard perspective is in use containing many visualisation components (*in such application states and usage's, real estate is key!*). Using the smaller resolutions involved having to scroll through perspectives and making data visualisation and analysis (*especially when using a lot of data points*) difficult as the points created a "merging effect as the got closer and started to overlap".

Overall the application worked and performed very well on the desktop and web-based platforms that were tested, and additionally to a satisfactory level within the different screen resolutions tested.

9.3.2.5 – Front-End Application Usability testing

Usability testing was not exhausted to a high level due to the fact that user feedback and further usability testing has been planned for and expected to be completed within a system "Alpha" release phase (*if it is the system is decided to be picked up and continued/pushed further*). However, A few "selected" users were asked to use the application to perform some set tasks relevant to the applications requirement to asses its overall usability (*at the POC stage*).
As the users performed their tasks they were asked to complete a provided usability-testing sheet where they documented any relevant feedback as appropriate relating to the task and application usability.
The sheet consisted of positive statements that were in favour of the application that were structured using an improvised "Likert Scaling" format keeping the answers in a consistent style allowing for appropriate further evaluation of the study.

The statements were relevant to the distinct tasks. The aim of the tasks was to give the user a goal to achieve which allowed them to interact and use the application. The statements were consistent throughout and were general usability related aimed to get an overall opinion of each user regarding the general usability of the application.

Some example statements include simple tasks: "Log-in to the application and log back out again" thus requiring a simple two step process to be performed, where as other were more complicated : "Log-in to the application and open a perspective" as the perspective list is not enables until data has been loaded into the application thus the user must therefore wither know or work out that they have to get data before they can open a perspective.

All results form the testing process are positive with no user complaining or asking/suggestion any changes. All of the users mange to work out and complete all of the tasks with no additional help required to be provided.
These results are expected at this level, as the application is still very small consisting of only two perspectives during the time of testing.
The results from the usability testing show that the implemented application is very strong and usable. Overall the users liked the application and thought is was nice and easy to use (*useable*), with all users keen to see and use the application with both more general and powerful/capable perspectives that can be used to interact with the data in a “rich” way allowing for a valuable data analysis experience.

9.4 – Testing Summary

The overall results gathered during the “back-end” system are good, however the testing process was very limited in relation to the solution being a theoretical based solution and therefore not being possible to be tested at a “practical” level as a “physical” artefact, however, all test results proved positive suggesting that the solution has covered all required levels to a high standard and relevant standard.
It has been justified that the established solution has all of the fundamental elements (*and more*) to acquire all of the relevant data required from a motorcycle, in a very accurate and efficient manner, that can further be used to provide the data for further use, for example within an advance “front-end” application. Such an application has been further implemented and has also been relevantly tested.

The overall results gathered during the “front-end”application testing phase are of a high quality, especially when taking into consideration that the application has been designed and implemented as a prototype (POC) application.
The functionality testing completed during integration testing was very detailed and in depth and the results produced were of a high standard with not many changes being required, being resolved in a fast and efficient manner.
This also justifies the accuracy of the unit testing that was performed implementation was both strict and efficient. .
The application met almost all of its requirements (*apart from distinct few*), with the requirements not being met being justified accordingly.
The application was compatible within both runtimes on different OS's and different web-browsers, being executed s not only expected on each OS and browser, but additionally the same on all platforms (*with no noticeable differences even between runtimes*). All the possible OS's and popular browsers were tested with a successful outcome being compatible, reliable and performing efficiently on each, as well as in different resolutions, with the only problem being of “strange” (although expected) effects when the amount of data points are greater than the amount of real estate (*in pixels*) causing the data to overlap.
The usability-testing phase proved to be an important stage of the test plan. All participating users had positive feedback and a positive feeling towards the application agreeing that it was a usable and enjoyable application to use.

Overall and looking at the test results collected, it can can be deemed that successful design, analysis and implementation phases have been completed and applied resulting is the creation of a successful and usable application.
Errors have been both identified and resolved during the testing process, as well as a few problematic areas being found and still remain within the application. These areas have been justified and documented and will be (*possibly*) take and built upon if the project is taken further in the future.

10 - Evaluation Chapter

This is the final chapter of the report and will be used to evaluate and discuss the complete project, identifying key contributions, strengths and weakness of the project, as well as its its personal and commercial values, the overall project achievements and further directions and project possibilities

10.1 - Project contributions

Many useful outcomes have been achieved during the project completion. Some of the major contributions include the establishment of a detailed and advanced system solution regarding the acquisition of data from a motorcycle (*traveling at speeds up to and exceeding potentially 200mph in places*) as it travels around a racetrack.
This system solution offers (*potentially*) very accurate, effective and justified results if implemented correctly thus able to be used to aquaria very interesting locational and statistical based information and measurements regarding both a motorcycles and riders "state"
during a lap(s) of a racetrack.
The System consists of an advanced DAQ device that is used to acquire (*all relevant and required*) statistical data using both internal functions and also from additional external sensors. The device additionally contains a built in and advance GPS receiver, thus providing the ability to acquire locational measurements, which is further aided by a documented DGPS system extension that is built into the solution providing better accuracy and reliability to the acquisition of locational measurements (*being a fundamental requirement of the system*).
An additional major system contribution was the design and development of the "front-app" application that runs off the proposed (*and discussed*) "back-end" foundation and data.
The application has been designed and implemented enabling a user to load the data that was acquired from the back-end system into the application and use it for further analysis. The application provides standard and "custom" perspectives consisting of visualisation components that data is fed into (*the same data acquired by the back-end system*) further plotting and rendering the data, allowing further data interaction and filtering possibilities (*using of custom control components*) enabling a veery "rich" and valuable analytical experience.

The two major contributions when working together (*as a complete system*) offer very impressive and advantageous results in relation to DAQ and data analysis possibilities within the motorcycle racing domain.

Many more additional minor contributions have been established thought the project, including smaller individual elements and components that are required and created to build up (step-*by-step*) the complete system model.
The main minor contributor that stands out is the created "intermediate" step between the two system stages (*major contributions*), consisting of an advance PHP script and UI that takes the data file that has been generated and transmitted from the "back-end" system (*the motorcycles DAQ device*) containing all of the acquired data in the format of a ".csv" file. The script converts the file into a series of SQL statements, creating a DB table and inserting the data accordingly. The script continues further to extract the data again from the DB and convert (*construct*) it as a valid XML document that can be used as a DP for the "front-end" application.
Therefore this intermediate step is a very advance step (***and even product in its own right***) that converts a dynamic (*any size or structure*) ".csv" into a valid XML Document constructing the XML nodes (*both parent and child*) relevantly and dynamically from the relevant headers and tupples within the ".csv" file, with the additional (*and optional*) step of creating and saving the data (*as a backup*) in a DB.
This additional step allows the data to be used for many other reasons and used, and within many other applications. A DB is the most commonly used data storage that data-driven applications and website use, therefore once the data is in the DB, many additional possibilities and usages of the acquired data from the motorcycle have been created in one single step.

10.2 - Project Achievements

A very large amount has been achieved not only at a development/project level during the project completion, with a lot more being achieved in comparison to what was initially planned and expected.
To help summarise the main achievement areas relevant to the project itself, the initial project aims and objectives will be referred to provide a better understanding of the achievements to a less "in-depth" technical level (*a more summarised and overview of the project as a whole*):

- A detailed project plan outlining the project schedule and development stages/life-cycle.
- In depth research and investigation into all necessary areas concerned.
- The identification of possible problematic and challenging areas.
- Solutions to these identified problematic and challenging areas.
- A detailed design specification.
- A successful implementation stage and guideline creation.
- A relevant and necessary testing and evaluation process.
- The completion of the project in both an ethical and professional manner.
- A great and valuable all round learning experience.

When referring to the initially documented project aims and objectives (*set at the beginning of the project*) it is clear that the project has been a great success achieving almost of what it set out to do and additionally a lot more!

Before the project was started, a detailed project plan was created. The plan helped to break the project down and understand the individual steps required to not only for completion, but complete it on time. however, the project plan was not strictly followed, with almost all of the deadlines either being missed and in some cases being completed early, thus highlighting bad project management.
At a personal level, it is believed that no more or less would have been achieved if the plan was strictly followed as a more flexible approach is always preferred and is better adopted to, however if the plan was followed strictly, it is believed that the project would have been completed in a more organised, structured and less stressful manner, without having to rush and have long nights trying to get everything completed on time (*as being the resulting case for this project*).

An in-depth and detailed literature search and review was performed identifying relevant sources, systems and more importantly areas that are necessary (*and even just beneficial*) for a successful project completion. Theses areas (*along with others that were discovered "along the way"*) were taken and investigated further to a high standard and level of detail within the R&I chapter.
The investigation process was by far the most in depth and long winded process throughout the whole project, having covered what is believed to be all required areas (*and more*) to a high level of both detail and understandings, resulting in a lot (*far more than expected*) possibilities and results, further leading to solid system solutions.
During the investigation a lot of problematic and challenging areas were discovered and solutions were identified with either solutions "work-arounds" to these areas being established accordingly, therefore (*and hopefully*) leaving no potential "holes" or areas of dramatic failures within the system solution.

Once the "back-end" system solution had been established, the "front-end" application design and implementation could commence. The design of the application (*even due to its prototype nature)* was taken very seriously, as is believed at a personal level very beneficial to get right first time round, at least at a foundation level, as the rest of the application will build up (*and theoretically work and fall into place*) from the initial system foundations (*similar to building a house*).
Detailed task and navigation models were created and documented accordingly, and form these models further detailed storyboards were created, which at a personal level, story-boarding is a very helpful and beneficial preferred "visual" approach to application design that has in the past, and again within this project proved very valuable.
After the design process was complete the implantation could be started. The application implementation was again a very long-winded and in depth process being both tricky and difficult within areas, although very interesting at the same time. The complete process was very smooth will no real "major" problems arising. Some problems did occur (*as documented*) that could not be resolved due to mainly time constraints being disappointing in relation to the overall application effectiveness, however these problems do not affected the initially set application requirements so much (*as they are additional features*), thus are not regarded (*in this sense*) as failed areas.

After the implementation process, a testing process was completed. The testing process was very laid back in relation to other more vigorous testing processes, as such a (vigorous) test plan, although interesting, was deemed irrelevant for the application in its proposed POC state.
The testing of the "back-end" system was a little restricted due to its "theoretical" solution nature opposed to a "physical" solution thus could not be tested at a "practical" level.
The testing process did however find and highlight some valuable areas and problems that required further investigation and fixing, which was completed in an efficient and practical manner.

The project was additionally completed in an ethical manner with all object(s) and person(s) being approach and handling in a relevant and professional manner.

The main values and achievement of the project completion is the leaning experience that has been gained as a result. A massive amount has been learned during the completion of the project not only in relation to motorcycle data acquisition domains but within the complete mobile and wireless domain as a whole, and additionally in other relevant areas including development, planning and computing in general. The learning outcome, curve and experience that has been established during the project completion is priceless.

10.3 Project Deliverables

Before the project was started the following project deliverables were set:

- A successful research program gaining a good and knowledgeable understanding of relevant research areas.
- A detailed project report documenting well each individual part and stage of the project.
- A detailed design specification of the PS.
- A set of detailed guidelines for guidance concerning hardware and technological elements regarding the collection of data from a mobile object (*motorcycle*) in relation to a specific environment (*racetrack*).
- A successful implementation producing a functional prototype (*front-end*) application, using (*test*) data from a relevant database.
- A detailed test plan and evaluation of the test outcome and results.
- a detailed critical evaluation of the overall project as a whole.
- A great learning enhancement within systems development.
- A great learning enhancement within the mobile and wireless computing domain.

In relation to the the set deliverables, after the completion of the project it is again clear and can be justified that the project has been a great success with all deliverable being completed and met to a high standard (*higher than initially thought possible when the deliverables were initially set*). It is additionally apparent that the project may have got "carried away" in areas and therefore going slightly off track due to the amount of additional work completed in different (*although still relevant*) areas. It is believed (*now looking back*) that time could possibly have been spent better concentrating on areas specific to the initially set requirements and deliverables further hilighting another problematic project management issues that will need to be addressed for future projects. Its fine to get side-tracked, however being side tracked can be very cost effective (*in relation to project completion goals*) as time (*and even money can be spent*) in areas that are not required. Luckily in the case of this project the sidetracked areas were will within relevance and interest of the project additionally proving beneficial, however highlighting the ease of going off course during larger scale projects.

10.4 - Research Field Comparisons

In comparison to other work completed within the similar domain to the completed system solution is (taking into account the systems POC nature) very powerful and advance. It is difficult to compare the system "completely" with other systems, as other systems are aimed at catering for other solutions and within other system domains (*with the most relevant being within car racing*). There are very few existing applications (*that have been found*) that are aimed specifically toward DAQ within motorcycle racing. Of the few systems that do exist, the PS offers a lot more (*wider range*) than all of the systems therefore offering more to a potential users and customers. Motorcycle systems exist that are more advance and accurate within specific areas, for example the lean angle measurement used within the MotoGP racing series, however such systems are specific to a particular measurements (*or area of measurement*) opposed to offering a more "complete" and general solution to DAQ analysis within the motorcycle racing domain which is what this system offers.

The most advanced system discovered, is the highly valued (*at a personal level*) and discussed RaceFX system that is used within the American NasCar and IndyCar racing series.
In comparison to this system, the PS is very basic, however, the RaceFX system is very large with a lot of time, experience money and a strong development team behind it. It is also very different having not only different goals and requirements, but also different target domain.
The similarities of the two systems is however the mobile and wireless foundations of the tow that following a similar route and setup.

10.5 - Commercial Usage and Value

The system a great commercial values and potential. An example use of the system is what is was originally deigned for, to be used as a learning and teaching aid for riders, visually highlighting at what points they are fast and slow, with additional possibilities of comparing the data against "favorable" measurements highlighting where improvements can be made, and how they can be achieved.
Another potential the system use could be within racing teams to analyse both rider and machine performance identifying weak areas. For example highlighting a riders "riding" style (inputs and technique) or lines taken, as well as motorcycle setup issues, including when power delivery or speed is low, or even where noticeable and frequent drifts occur during cornering. Overall highlighting very valuable areas that could potentially "make or break" race results.

Example uses for the application are almost endless within the subject domain, with the overall system goal being to reduce lap times performed by both rider and machine, in the fastest and safest method possible.

10.6 - General Conclusions

This section of the evaluation will highlight the projects projects strength and weaknesses as well as any additional areas deemed relevant for discussion.

Overall the project is believed to be successful meeting all of its set requirements, deliverables, aims and objectives to a very high standard.
The project contains many strengths and weaknesses that will be discussed further within this chapter, with strengths consisting of the obvious (*and already discussed*) areas including the overall project success, the finalised solution and application, the addition to the mobile and wireless research domain and the overall and learning outcome of the project, and some of the weaknesses including the poor project management and the failure to complete all of the application perspective classes in time.

Reading back through the project it is clear that it has been attacked at a very "practical" level. This is not deemed as a weakness but even a strength in relation to personal approaches and methods of achieving goals. I am very practical person, and therefore this is identifiable within the project.

Reading thorough the project, the practical element is further noticed in the texts as all items are described and explained in depth at a very practical level, which is deemed as in a way disadvantage for such a report (*especially for readers*) who are interested in the project at such a "practical" level, but require and more overviewed (*easier to read*) insight into and about the project. This is another noticeable weakness of the report, as if the reader does not know, understand or is aware of what is being discussed within certain areas of the report, it can become very confusing.
This "report writing" issue has noticed and taken on board and will be used as a learning experience that will be improved upon within future projects.

The project started to follow set guidelines and structure, but as the project became more in depth, it started to become (*in an exaggerated way "uncontrollable"*) growing into its own "animal" further taking its own directions, that I simply took the easy route, and pursued, again proving bad project management, and highlighting an additional area that needs to be improved upon for further work.

The project was (and had to be) completed in a very flexible manner (*mainly due to time constraints*). The project was completed on top of working full time working on other projects of similar (and larger) scales, also being very intensive and time consuming. Therefore the project was was not completed in a flowing or smooth synchronised process, but in fact more asynchronously without any particular order or structure, with different areas were attacked at different times and stages throughout the project which is noticeable in the in the body and structure of the report.
Although a less structured, strict and flexible approach is (*and was*) preferred, this approach was believed to be inadequate and too unstructured to enable the creation of a perfectly flowing and "smooth" result, however, due to other constraints such an approach was difficult to avoid.
For similar future project under similar conditions, such a process is believed to benefit from a more detailed project plan that focusses on the constraints and caters for them accordingly.

Plenty of time was available to successfully compete the project, but due to project management failures, the report still had a lot of loose ends that required "tying" up towards the project deadline, which was therefore a completed and "last minute" panic, resulting in many late nights and not much sleep.
This "last minute" approach is however not personally believed as a weaknesses, all of my previous projects have been completed in a "last minute" fashion. This is not through laziness or not wanting to complete a project. This project has been a very interesting and rewarding process and I have looked forward to, and enjoyed working in and completing the project. I believe that I just work a lot better under pressure, and it is the pressure that is required to build up motivational levels, thus get the job done.

One of the most difficult areas of the project was (*as discussed*) my full-time job. I work every day on computer projects with strict deadlines, therefore in many cases having to work long and late hours. Motivation was a real issue during the completion of the project as it was very difficult to motivate myself continue to sit in front of my computer working on another project as soon as I am home from work.
I am additionally very much an outdoors person, so again it was very difficult for me to sit inside during sunny weekends working on my computer, as I wanted to be out enjoying the weather and the outdoors in general, however looking back, and considering what I have achieved and learned during this project, all the additional hard work and restrictions experienced has been more than worthwhile.

An additional note regarding the application implementation. The project application was created using Adobe Flex. I use flex every day at work and this one of the reasons that the technology was chosen. The already obtained experience within the framework was believed to (*and did*) speed up and improve the application implementation process.
During the implementation project a lot of new areas were covered using the technology in which a lot was learned and gained from, and can be further taken and used within and benefit future application and project development processes.

10.7 - Future directions and recommendations

The area in which the project excels the most is the expansion possibilities that have been opened and created during the project completion. Just as many additional system possibilities in relation to progression and expansion into new and valuable areas have been found as areas that have been solved. almost every section of each chapter highlights at least one are for further expansion and future direction, thus being a foundation for a very powerful system offering a lot of "potential" possibilities.

The documented change of the proposed system application to a framework based foundation, was a massive step in the direction of creating further system directions an possibilities.
The original and planned application was to be a stand alone application that met the requirements of this "specific" system. Changing it form a single system into a complete framework allows the creation of multiple DAQ applications that can be run from the same (*created*) application framework offering obvious and beneficial areas and solutions to unlimited (*yet relevant*) application and component possibilities, expansions and capabilities.

Another area worthy of inclusion within this chapter (as believed very worthwhil at a personal level) is that of the system upgrade to support veideo overlays durina analysism through the use of onboard cameras installed on the motorcycle
"... *The obtained positioning information acquired from the GPS system could be further interpolated to correspond with measurements acquired by the on board cameras. The camera could additionally measures factors such as pan, tilt and zoom which could be further combined "frame-to-frame" with the acquired location information relevant to a particular frame, allowing an accurate video overlay to be defined* ..." overall being believed to offer a lot in terms of results (*implementation time vs achieved results*) aiding the system dramatically.

The overall project has been veery successful in relation to meeting its set goals and requirements. All that was planned to be achieved and wanted to be achieved has been, to a very satisfactory level, and a level that was not expected from the beginning of the project.
The main most rewarding outcome of the project it the amount of learning that has been achieved. The knowledge that has been acquired within the mobile and wireless domain, and especially within the

data acquisition and GPS domain within wireless systems is amazing. I did not think when I was initially planning this project, how much additional knowledge and confidence I would acquire during the project development stages. and for this reason, every ounces of stress, time and worry spent on the project is worthwhile.

I hope to take the project further within the future, and pick up and resolve any loosed ends that exist, further exploring into some of the many identified directional and expansion areas that have been identified, so watch this space ...

11 - References

- [1]-(J Chen, C Adams), (2004), (Short-range wireless technologies with mobile payments systems, ACM International Conference Proceeding Series), (ISBN:1-58113-930-6).
- [2]-(Norwich Union), (2007), (UK Motorcycling Interest Growth), (From: http://www.norwichunion.com).
- [3]-(AmaPro racing), (2006), (AMA Pro Racing, Series Guide 06), (From: http://www.amaproracing.com).
- [4]-(MotorbikeNews), (2006), (British Development Corporation, Loughborough Innovation Center based Intelligent Energy and Suzuki), (From http://www.motorbikenews.co.uk).
- [5]-(J Oliver), (1997), (Teach yourself research, NTC publishing group), (ISBN: NA).
- [6]-(J McGivern), (2003), (The practice of market and social research, Personal education limited), (ISBN: 0-273-65506-X).
- [7]-(C Spencer), (1990), (Digital Design for Computer DAQ. Cambridge University Press), (ISBN: 0-521-37199-6).
- [8]-(S McBeath), (2002), (Competition Car Data Logging: A Practical Handbook), (ISBN: 1-85960-653-9).
- [9]-(National Instruments), (2006), (GPS synchronisation Architecture for Data Acquisition Devices, National Instruments Corporation).
- [10]-(P Mann), (2004), (Timing synchronization for 3G wireless, EE Times, Symmetricom Inc).
- [11]-(National Instruments), (2007), (Data Acquisition Fundamentals, Application Note 007, National Instruments Corporation).
- [12]-(N Gerein, A Brown), (2006), (Modular GPS Software Radion Architecture, NAVSYS Corporation).
- [13]-(E Strassberg, A Deshpande, C Guestrin, S Madden, J Hellerstein, W Hong), (2006), (Model-Driven Data Acquisition in Sensor Networks).
- [14]-(J Curtis), (2002), (Good DAQ driver software: small considerations to avoid big problems. (DAQ), Scientific Computing & Instrumentation).
- [15]-(S Bhattacharyya, U Maulik, P Dutta), (2009), (High-speed target tracking by fuzzy hostility-induced segmentation of optical flow field), (ISSN:1568-4946).
- [16]-(Nology Enginieering Inc), (2007), (G-Dyno Plus GPS Performance Data Acquisition System System Nology Enginieering Inc), (From: http://www.nology.com).
- [17]-(Traqmate), (2008), (Traqmate.com GPS Data Acquisition), (From: http://www.traqmate.com).
- [18]-(Doble Engineering), (2008), (F6050, Universal Time Stchronizer, Doble Engineering Company).
- [19]-(P Jueang, H Oki, Y Wang, M Martonosi, L Peh, D Rubenstein), (2002), (Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet), (ISBN:1-58113-574-2).
- [20]-(T Hi, P Sikka, P Corke, L Overs), (2004), (Wireless sensor devices for animal tracking and control), (DOI: 10.1109/LCN.2004.141).
- [21]-(S Krishnamurthy, T He, L Luo, R Storeo, T Yan, J Hui, B Krogh), (2006), (Energy-efficient surveillance system using wireless sensor networks), (ISBN:1-58113-793-1).
- [22]-(M Dohler), (2008), (Wireless Sensor Netoworks, The Biggest Cross-Community Design Exercise To-Date), (PN:1,9-25).
- [23]-(H Goh, Moh Lim Sim, Hong Tat Ewe), (2006), (Energy Efficient Routing for Wireless Sensor Networks with Grid Topology), (ISBN: 978-3-540-36679-9).
- [24]-(M Friedermann, N Jeon, C Leem, M Kim, 2004), (Wireless Commincations, A taxonomy of ubiquitous computing applications), (DOI 10.1007/s11277-007-9297-9).
- [25]-(J Tavares, F Velez, J Ferro), (2007), (Application of Wireless Sensor Networks to the Automobile, RTCM).
- [26]-(J Stankovic, T He, S Krishnamurthy, T Abdelzaher, L Luo, R Stoleru, T Yan, L Gu, B Krogh), (2007), (An Integrated Sensor Network System for Energy Efficient Surveillance System Using Wireless Sensor Networks), (ISBN:1-58113-793-1).
- [27]-(M Ruggeiri, D Deepak Ayyagari, A Ephremides, 2008), (Wireless Networks, A satellite-augmented cellular network concept), (ISSN:1022-0038).
- [28]-(A Smith, H Balakrishnan, M Goraczko, N Priyantha), (2004), (Support for location, Tracking moving devices with the cricket location system), (ISBN:1-58113-793-1).
- [29]-(F Dugast, J Sikora), (2007), (The Cricket Indoor Location System, Facualty Informatik, Institute of System Architecture, University Dresden).
- [30]-(J Hightower, j Astrain, J Villadangos, R Garitagoitia, J González de Mendívil, V Cholvi), (2006), (Fuzzy location and tracking on wireless networks), (ISBN:1-59593-488-X).
- [31]-(H Balakrishnan, T Iwamoto, A Kobayash, S Nishiyama), (2007), (Preserving Anonymity in Indoor Location System by Context Sensing and Camera-Based Tracking), (ISSN 0302-9743).
- [32]-(A Smith), (2004), (A Three Dimensional Choke Ring Ground Plane Anntenna, NovAtel inc).
- [33]-(H Bienser, J Rudd, R Benner), (1998), (Real-time APL prototype of a GPS system), (ISSN:0163-6006).
- [34]-(H Wu, C Wang, N Tzeng), (2005), (Self-configurable positioning technique for multihop wireless networks), (ISSN:1063-66923).
- [35]-(V Zeimpekis, G Giaglis, G Lekakos), (2002), (A taxonomy of indoor and outdoor positioning techniques for mobile location services), (ISSN:1551-9031).
- [36]-(Garmin), (2008), (Potential Garmin GPS Useages), (Garmin International Corpoation), (From: http://www.garmin.com).
- [37]-(G Blewitt), (1997), (Basics of the GPS Technique: Observation Equations, Department of Geomatics, University of Newcastle).
- [38]-(J Zogg), (2002), (GPS Basics, Introduction to the system application overview).
- [39]-(W Kunysz), (2004), (High Performance GPS Pinwheel Antenna, NovAtel Inc).
- [40]-(G Lachapelle, C Liu, G Lu, M Cannon, B Townsend), (1995), (Precise Marine DGPS Positioning Using P Code and High Performance C/A Cade Technologies, canadian Hydrographic Service (pacific Region)).
- [41]-(J Mikulski, 2006), (Advances in Transport Systems Telematics), (ISBN 83-917156-4-7).
- [42]-(N Nixon), (2004), (Keep on track: pioneered by the Formula 1 teams, telematics can offer significant benefits to the van fleet operator – telematics), (From: www.findarticles.com/p/articles/mi_m0MXJ/is_1_10/ai_105769563).
- [43]-(E Belgeonne), (2006), (Thale Telematic Systems), (From: http://www.thalestelematics.com).
- [44]-(L Armstrong), (2000), (Dedicated Short Range Communications (DSRC)), (From:

http://www.leearmstrong.com/DSRCHome/GeneralInfo/DSRCGeneral).

- [45]-(J Yin, T ElBatt, G Yeung, B Ryu, S Habermas, H Krishnan, T Talty), (2004), (VANET channel characteristics and safety applications: Performance evaluation of safety applications over DSRC vehicular ad hoc networks), (ISBN:1-58113-922-5).
- [46]-(W Randall, Y Kim, J Gratch), (2002), (mobile embodied agents: Anticipating where to look: predicting the movements of mobile agents in complex terrain), (ISBN:1-58113-480-0).
- [47]-(J Busack), (1997), (Auto Race Monitoring System), (pn:6020851).
- [48]-(T Melgrad, D Last, B Thomas), (1999), (Precise GPS time Transfer to a moving vehicle), (DCN Brest (DGA), France).
- [49]-(J Skaloud, Q Ladetto, B Merminod, M Vetterli, MGyr, A Marcacci, P Luthi), (2001), (GPS World - With Racing Heart).
- [50]-(K Milnes, T Ford), (2002), (Applying GPS Tracking Devices: Real-time RaceFX), (NovAtel, RaceFX).
- [51]-(J Magee), (2007), (Multi Sensor Location Tracking), (ISBN:1-58113-035-X).
- [52]-(H Kung), (2004), (Efficient location tracking using sensor networks), (ISBN: 0-7803-7700-1), (ISSN: 1525-3511).
- [53]-(R Schack), (1994), (The length of a typical Huffman codeword), (DOI: 10.1109/18.335944).
- [54]-(D.A. Huffman), (1952), (The Huffman Method).
- [55]-(L Turbak), (2001), (Huffman Tree Example), (CS231, Algorithms).
- [56]-(Abatec Electronics AG), (2008), (From: http://www.abatec-ag.com).
- [57]-(S Kennedy, D Cosandier, J Hamilton), (2008), (GPS Integration in Real-time and Post-processing with Novatel's SPAN System), International Global Navigation Satellite Systems Society IGNSS Symposium 2007.
- [58]-(NovAtel Inc), (2009), (From: http://www.novatel.com).
- [59]-(K Milnes, T Ford), (2002), (Real-Time GPS FX, On Screen Positioning of Race-Cars), (Sport Vision Inc).
- [60]-(ESRI), (2003), (GIS and Mapping Software), (From: http://www.esri.com).
- [61]-(L-3), (2003), (L-3 Communications Telemetry West), (Excellence you can measure).
- [62]-(The European Society of Telemetering), (2009), (From: http://www.aktm.org).
- [63]-(J Neumann, T Ford), (1994), (NovAtel’s RT20 - A Real Time Floating Ambiguity Positioning System), (Proceedings of ION GPS ‘94), (The Institute of Navigation,Washington, D.C), (pp. 1067–1076)
- [64]-(A Manz, O Mulyk), (1996), (Test Results from a New 2 cm Real Time Kinematic GPS Positioning System), (Proceedings of ION GPS ‘96), (The Institute of Navigation,Washington, D.C), (pp. 873-882)
- [65]-(T Ford, K Milnes), (2002), (Track Model Constraint Enhancement for NovAtel’s OEM4), (Proceedings of KIS Conference)
- [66]-(Nology Systems), (2008), (G-Dyno Plus), (from: http://www.nology.com)
- [67]-(G-Dyno Plus with GPS, 2008), (Automotive Performance Computers and Data Acquisition System with Accelerometer and GPS), (Nology Systems)
- [68]-(Sirf, 2009), (from: http://www.sirf.com)
- [69]-(RacePak Data Systems), (2009), (Professional motor-sports data acquisition systems), (G2X and G2X-pro), (from: http://www.racepak.com)
- [70]-(Veracity Racing Data, 2008), (Data Aquisisiot Specialists), (DL1 data logger), (from: http://www.veracitydata.com)
- [71]-(Race Technology), (2008), (DL1 Data Logger), (from: http://www.race-technology.com)
- [72]-(TraqMate), (2008), (GPS Data Acquisition), (from: http://www.traqmate.com)
- [73]-(PI Research describe), (2008), (PiIMU), (PI Research Inc)
- [74]-(Xens Motion Technologies), (2009), (MTi, Mtx), (from: http://www.xsens.com)
- [75]-(Team Domination), (2006), (Team Domination Data Acquisition System Paper)
- [76]-(AWI TechNet), (2001), (Object-Oriented Programming Overview), (from: http://www.archwing.com/technet/technet_OO.html)
- [77]-(W Sutherland), (1966), (The On-Line Graphical Specification of Computer Prcedures), (MIT)
- [78]-(Mike Niel), (2005), (Labview Product manger), (National Instruments)
- [79]-(National Instruments), (2009), (LabVIEW Product Description), (from: http://www.ni.com/labview)
- [80]-(The MathWorks), (2009), (MATLAB and Simulink for Technical Computing), (form: http://www.mathworks.com)
- [81]-(Adobe Flash Player), (2009), (Adobe Products), (from: http://www.adobe.com/products/flashplayer)
- [82]-(Adobe Flash Professional), (2009), (Adobe Flash CS4 Professional), (Adobe Produscts), (from: http://www.adobe.com/products/flash)
- [83]-(Ted Patrick), (2008), (Adobe Systems), (Adobe Flex), (from: http://www.onflex.org)
- [84]-(E Walters), (2000), (Essential Guide to Computing), (Prentice Hall PTR), (0-13-019469-7)
- [85]-(Edward W Tennant), (2005), (GPS in the field), (A Practical Guide to the Theory and Application of GPS Technologies)
- [86]-(J Marchewka), (2003), (Information technology project managemt), (John wiley and sons inc), (0-471-39203-0)
- [87]-(SearchVB), (2004), (SLDC), (from: http://searchvb.techtarget.com/sDefinition)
- [88]-(S Standings), (2000), (Internet Commerce Development), (Artech House INC), (1-58053-051-6)
- [89]-(R Campion, G Crum, K Liu), (1995), (TRIuMPh), (A Method for Multimedia Interface Design)
- [90]-(D Benyon), (1990), (A guide to usability), (0-7492-4344 9)
- [91]-(Dix, Finlay, Abowd, Beale), (1993), (Human Computer Interaction), (0-13-437211-5)
- [92]-(The Usability Company), (2005), (from: www.theusabilitycompany.com)
- [93]-(Ben Holmes, Brendan Meutzner), (2009), (FlexMDI Framework), (Robust, Extensible MDI Framework for Adobe Flex)
- [94]-(Chris Giametta), (2007), (Flex, Spring, iBATIS, Cairngorm: Bringing It All Together), (from: http://www.appfoundation.com/blogs/giametta/2007/)
- [95]-(D Stottlemyer), (2003), (Automated Tssting Toolkit), (978-0471414353)

12 - Bibliography

12.1 - Online Resources and E-Journals

- (A theoretical study of optimization techniques used in registration area based location management: models and online algorithms), (Sandeep K. S. Gupta, Goran Konjevod, Georgios Varsamopoulos), (2002), (ISBN:1-58113-587-4)
- (A System based on Mobile Agents for Tracking Objects in a Location-dependent Query Processing Environment), (Sergio Ilarri, Eduardo Mena, Arantza Illarramendi), (2001), (ISBN:0-7695-1230-5)
- (Efficient In-Network Moving Object Tracking in Wireless Sensor Networks), (Wen-Chih Peng, Yu-Chee Tseng), (2006), (ISSN:1536-1233)
- (TRAX: real-world tracking of moving objects), (Christian S. Jensen ,Stardas Pakalnis), (2007), (ISBN:978-1-59593-649-3)
- (Real-Time Feature Matching in Image Sequences for Non-Structured Environments. Applications to Vehicle Guidance), (Joaquín Ferruz, Aníbal Ollero), (2000), (ISSN:0921-0296)
- (Tracking Multiple Moving Objects for Real-Time Robot Navigation), (Erwin Prassler, Jens Scholz, Alberto Elfes), (2000), (ISSN:0929-5593)
- (Modeling dynamic scenarios for local sensor-based motion planning), (Luis Montesano, Javier Minguez, Luis Montano), (2008), (ISSN:0929-5593)
- (BlueTrak-A Wireless Six Degrees of Freedom Motion Tracking System), (Hans Krüger, Lasse Klingbeil, Edgar Kraft, Rene Hamburger), (2003), (ISBN:0-7695-2006-5)
- (Data acquisition and cost-effective predictive modeling: targeting offers for electronic commerce), (Foster Provost, Prem Melville, Maytal Saar-Tsechansky), (2007), (ISBN:978-1-59593-700-1)
- (Adaptive training of video sets for image recognition on mobile phones), (Erich Bruns, Oliver Bimber), (2009), (ISSN:1617-4909)
- (Wireless technologies for data acquisition systems), (Sakari Junnila, Jarkko Niittylahti), (2003)
- (A simple practical approach to a wireless data acquisition board), (Smilen Dimitrov, Stefania Serafin), (2006), (ISBN:2-84426-314-3)
- (Wireless Sensor Network and Remote Data Acquisition System for Mobile Applications), (Jean Paul Talledo Vilela, Jose Carlos Miranda Valenzuela), (2006), (ISBN:0-7695-2505-9)
- (An Adaptive and Autonomous Sensor Sampling Frequency Control Scheme for Energy-Efficient Data Acquisition in Wireless Sensor Networks), (Supriyo Chatterjea, Paul Havinga), (2008), (ISBN:978-3-540-69169-3)
- (Global Positioning System (GPS) Time Dissemination for Real-Time Applications), (Peter H. Dana), (1997), (ISSN:0922-6443)
- (International Journal of Mobile Communications archive), (Sameer Kumar, Joel Stokkeland), (2003), (ISSN:1470-949X)
- (Tracking multiple mobile objects using IEEE 802.15.4-based ultrasonic sensor devices), (Shinyoung Yi, Jiyoung Yi, Hojung Cha), (2007), (ISBN:1-59593-480-4)
- (Mobile object tracking in wireless sensor networks), (Hua-Wen Tsai, Chih-Ping Chu, Tzung-Shi Chen), (2007), (ISSN:0140-3664)
- (Service composition for mobile environments), (Dipanjan Chakraborty, Anupam Joshi, Tim Finin, Yelena Yesha), (2005), (ISSN:1383-469X)
- (A practical approach for accurate positioning with L1 GPS receivers using neural networks), (M. R. Mosavi), (2006), (ISSN:1064-1246)
- (A Road-Matching Method for Precise Vehicle Localization Using Belief Theory and Kalman Filtering), (Maan E. El Najjar, Philippe Bonnifait), (2005), (ISSN:0929-5593)
- (Information fusion for wireless sensor networks: Methods, models, and classifications), (Eduardo F. Nakamura, Antonio A. F. Loureiro, Alejandro C. Frery), (2007), (ISSN:0360-0300)
- (Wireless sensor network survey), (Jennifer Yick, Biswanath Mukherjee, Dipak Ghosal), (2008), (ISSN:1389-1286)
- (On the lifetime of wireless sensor networks), (Isabel Dietrich, Falko Dressler), (2009), (ISSN:1550-4859)
- (A Parallel Architecture for Network Control and Mobility Tracking in Wireless Systems), (Abhaya Asthana, Paul Krzyzanowski), (1997), (ISSN:0929-6212)
- (Hierarchical architecture for managed wireless networks), (Balázs Kovács, Rolland Vida, Gergely Biczók), (2005), (ISBN:1-59593-197-X)
- (The empirical study of automotive telematics acceptance in Taiwan: comparing three Technology Acceptance Models), (Huei-Huang Chen, Shih-Chih Chen), (2009), (ISSN:1470-949X)
- (A Rapid Prototyping Approach For The Design Of Extensible In-Vehicle Telematics Systems), (D. Reilly, A. Taleb Bendiab, John Moores), (2002), (ISSN:1092-0617)
- (Cooperative collision warning using dedicated short range wireless communications), (Tamer ElBatt, Siddhartha K. Goel, Gavin Holland, Hariharan Krishnan, Jayendra Parikh), (2006), (ISBN:1-59593-540-1)
- (Vehicle-to-vehicle safety messaging in DSRC), (Qing Xu, Tony Mak, Jeff Ko, Raja Sengupta), (2004), (ISBN:1-58113-922-5)
- (Location-Based System for Mobile Devices Using RFID), (Seyed Hossein Siadat, Ali Selamat), (2008), (ISBN:978-0-7695-3136-6)
- (Calibration-free WLAN location system based on dynamic mapping of signal strength), (Luís Felipe M. de Moraes, Bruno Astuto A. Nunes), (2006), (ISBN:1-59593-488-X)
- (Design of indoor positioning systems based on location fingerprinting technique), (Kamol Kaemarungsi), (2005), (ISBN:0-542-31142-9)
- (Structured data flow programming), (M. D. de Jong, C. L. Hankin), (1982), (ISSN:0362-1340)
- (Mathematical semantics and data flow programming), (Paul R. Kosinski), (1976)
- (Graph based data flow programming of parallel machines), (Stephan V. Bechtolsheim), (1985)
- (Supporting Design Patterns in a Visual Parallel Data-flow Programming Environment), (Masashi Toyoda, Buntarou Shizuki, Shin Takahashi, Satoshi Matsuoka, Etsuya Shibayama), (1997), (ISBN:0-8186-8144-6)
- (Applications of Dynamic Data Flow Programming to Real-Time Interactive Simulations), (Stephen A. Morrison), (2001), (ISSN:1080-241X)
- (Visual programming in data flow environments), (Craig Upson), (1992)

- (Road survey for GIS by means of low cost DGPS/DR), (R. Cefalo, G. Manzoni, G. Skerl), (1997), (ISBN:3-211-82806-0)
- (Dgps-aided ins land vehicle navigation systems), (Faroog Abdel-Kareem Ibrahim), (2000), (ISBN:0-599-90370-8)
- (Automatic Guidance of a Farm Tractor Relying on a Single CP-DGPS), (B. Thuilot, C. Cariou, P. Martinet, M. Berducat), (2002), (ISSN:0929-5593)

12.2 - Books

- (Protocols and Architectures for Wireless Sensor Networks), (Holger Karl, Andreas Willig), (2007), (ISBN: 0470519231)
- (Wireless Sensor Networks: Architectures and Protocols), (Edgar H. Callaway Jr.), (2003), (ISBN: 0849318238)
- (GPS For Dummies), (Joel McNamara), (2008), (ISBN: 0470156236)
- (Understanding GPS: Principles and Applications), (Elliott D. Kaplan, Christopher Hegarty), (2005), (ISBN: 1580538940)
- (Aided Navigation: GPS with High Rate Sensors), (Jay A. Farrell), (2008), (ISBN: 0071493298)
  (Development and Testing of a Multiple Filter Approach for Precise DGPS Positioning and Carrier-Phase Ambiguity Resolution), (Paul E. Henderson), (2001), (ISBN: 1423528670)
- (Data Acquisition for Sensor Systems), (H.R. Taylor), (1997), (ISBN: 0412785609)
- (PC Interfacing and Data Acquisition: Techniques for Measurement, Instrumentation and Control), (Kevin James), (2000), (ISBN: 0750646241)
- (Smart Sensor Systems), (Gerard Meijer), (2008), (ISBN: 0470866918)
- (PC Interfacing and Data Acquisition: Techniques for Measurement, Instrumentation and Control), (Kevin James), (2000), (ISBN: 0750646241)
- (Taking Telematics into the 21st Century), (Sala Mandik, Michael Rigby, Ruth Roberts, Michael Thick), (2000), (ISBN: 1857753445)
- (LabVIEW for Everyone: Graphical Programming Made Easy and Fun), (Jeffrey Travis, Jim Kring), (2006), (ISBN: 0131856723)
- (Research Methods for Human-Computer Interaction), (Paul Cairns, Anna L. Cox), (2008), (ISBN: 0521690315)
- (Interaction Design: Beyond Human-Computer Interaction), (Helen Sharp, Yvonne Rogers, Jenny Preece) , (2007), (ISBN: 0470018666)

12.3 - Websites

- (Discrete Wireless GPS Fleet Management - Dispatching - Vehicle Tracking), (2009), (from: http://www.discretewireless.com/)
- (Wireless Tracking System uses RFID/GPS to locate containers), (2009), (from: http://news.thomasnet.com/fullstory/522366)
- (Wireless tracking system and method utilizing tags with variable power level transmissions), (2008), (from: http://www.freepatentsonline.com/7504928.html)
- (Open source personal tracking system gets first test), (2009), (from: http://www.infoworld.com/article/07/01/02/HNopensourcetracking_1.html)
- (GPS and Racing Applications), (1995), (from: http://www.biggerhammer.net/offshorepage/features/gps/raceapps.html)
- (Electronics in auto racing), (2007), (from: http://www.edn.com/index.asp?layout=article&articleid=CA6479489&industryid=47041)
- (Advantage Motorsports Race Car Computer Systems and Racing Software), (2008), (from: http://www.advantagemotorsports.com/)
- (Race car design and software), (2008), (from: http://www.roadsters.com/software/)
- (DAQs track race-car moves), (2004), (from: http://www.racecartesting.com/images/Machine_Design.pdf)
- (A multi-sensor system for positioning in urban environments), (2001), (from: http://cat.inist.fr/?aModele=afficheN&cpsidt=14785272)
- (MATLAB), (2008), (from: http://www.mathtools.net/MATLAB/)
- (InterAction), (2009), (from: http://www.bcs-hci.org.uk/)
- (Centre for HCI Design), (2009), (from: http://www-hcid.soi.city.ac.uk/)
- (Adobe Flex Product Center), (2009), (from: http://labs.adobe.com/technologies/flex/)
- (What Is Adobe/Macromedia Flex?), (2008), (from: http://www.cflex.net/about_adobe_flex.cfm)
- (MATLAB), (2009), (from: http://www.mathworks.co.uk/)
- (LabVIEW Tutorials), (2007), (from: http://www.upscale.utoronto.ca/GeneralInterest/LabView.html)
- (NI LabVIEW), (2009), (from: http://www.ni.com/labview/)
- (Telemetry System Equipment), (2008), (from: http://www.epgco.com/wireless-switches.html)
- (What is Telemetry?), (2008), (from: http://www.tech-faq.com/telemetry.shtml)
- (Telemetry Systems), (2007), (from: http://www.l-3com.com/tw/tutorial/telemetry_system_overview.html)
- (Telemetry System Overview), (2008), (from: http://www.comtechm2m.com/m2m-telemetry-system/m2m-telemetry-system.htm)
- (Dynamic non-DGPS positional accuracy performance between recreational and professional GPS receivers), (2007), (from: http://www.informaworld.com/smpp/content~content=a779468037~db=all~jumptype=rss)
- (DGPS corrections over the Internet), (2008), (from: http://www.wsrcc.com/wolfgang/gps/dgps-ip.html)
- (Differential GPS), (2005), (from: http://www.gpsinformation.org/dale/dgps.htm)
- (Talking about GPS - What is Differential GPS ?), (2007), (from: http://www.effective-solutions.co.uk/dgps1.html)
- (DGPS, GPS and Tracking Solution Specialists), (2009), (from: http://www.dgps4u.com/)
- (GPS Warehouse), (2009), (from: http://www.gpsw.co.uk/)

13 - Acknowledgments

This project has benefitted from valuable and generous contributions from a selection of people. I would like to acknowledge these people as a sign of thanks and appreciation within this section of the report.

In relation to the project Research and Investigation would like to thank:

- **Ken Milnes** (*SportVision, Sr. VP Engineering*) from the RaceFX team. Ken is one of the developers of the RaceFX system, and the author of a very valuable literature regarding the system that has additionally benefitted the project greatly. A contact was established with Ken during the Literature review stages of the project regarding system possibilities and technicalities. Ken has offered valuable device and literature sources that have greatly improved the development of the system solution pushed the project in the correct directions. Ken additionally provided the sample data that was used to develop and test the Front-End application.

- **Juergen Loeberbaur** (*Abatec Electronic AG*). A contact with Juergen was established during the project investigation phase requiring some information regarding Abatec Location Based Sensor System. Jeurgen provided me with the information that I required as well as additional presentations regarding similar technologies and areas relevant to, and to help with my research progress. Jeurgen also “offered” his support and advice regarding the system domain as and when I required / wanted it.

- **Edyson Pavilcu** (*BitRaptor*). Edyson was another contact established during the project investigation regarding information and advice regarding a technology developed by BitRaptor. Edyson was pleased with my interested within the domain and offered advice regarding the selected technology along with further general advice and support regarding the investigation process and project as a whole.

I would like to thank some people at a more personal level relative to the project:

- **Dave Thomas** (*Project Advisor*). Dave is the project supervisor and has provide both support and guidance throughout the entire project development. Dave have been a great support in relation to the completion of the project. He has understood the conditions in which the project has been completed and adapted to them relevantly allowing (*and trusting*) me to complete the project in my own (*very diverse and flexible*) way which is appreciated. Dave has been available for advice night and day, seven days of the week (*including the weekends!*). Dave has not only been available for mentoring reasons, but has also been a friend offering general support and advice, further being available for a beer and a chat during his visits to Germany.

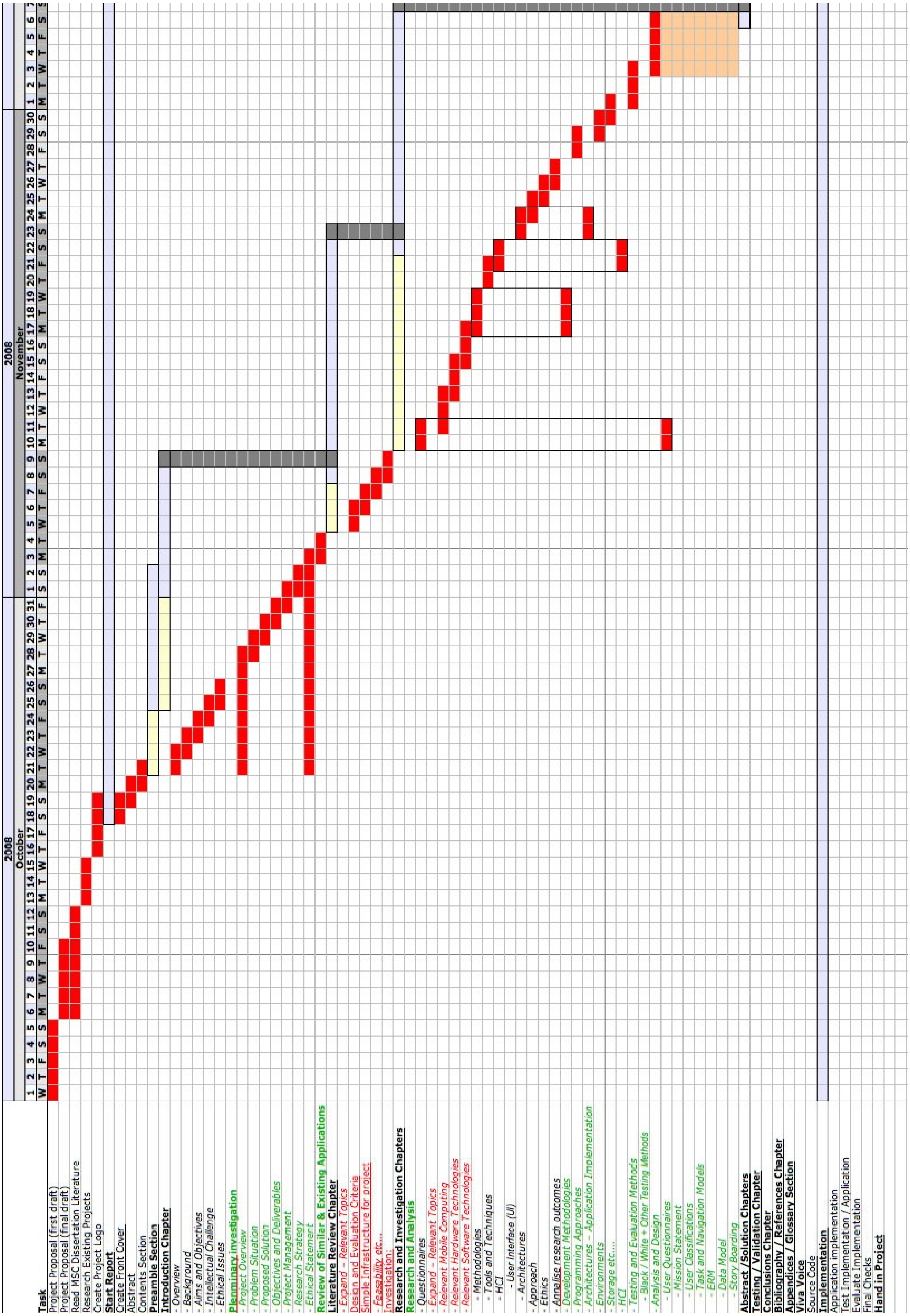
- **Elena Emminger**. Elena has provided “general” support, advice, motivation (*and a lot of patience!*) throughout project. She has helped me through the “difficult” and stressful times and areas during the project, and has provided both excellent food and drink (*which is always appreciated*) providing energy and further motivation to get the job done.

- **Susan and David Jones**. Susan and David have helped greatly throughout the project in terms of both support and advice. They have provided a lot of time and have been available throughout the whole project to read and listen to my ideas, understanding the concepts providing a “second person” perspective, and proof reading “selected” areas of the report checking general spelling and grammar, which has proved very valuable and is greatly appreciated.

- **June Williams**. June has provided support throughout the complete project. She has been available for general advice and support or even just for a “chat” helping build up motivation levels during stressful times. June has provided financial support, throughout the whole masters degree. It is believed that without this support (*both general and financial*) the project could not have been completed. Therefore I would like to say a final Thank You to June for her help and support throughout both the project and the complete Masters course - it is very much appreciated!

14 - Appendices

<u>1 - summarised Project Plan (Gantt Chart)</u>



2 - Local Position Measurement Components Investigation

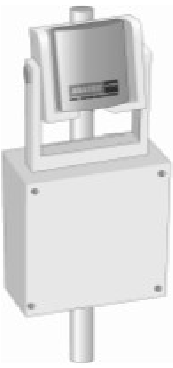
To enable an LPM system to function various components are required to be in place within the first principle process working both independently and with one another, acquiring and processing data from a mobile object. To get a better idea of the requirements of such system and the hardware that is needed to achieve such results, an example LPM system setup (*based upon the standard choice of components used within an Abatec LPM system*) will be investigated:

2.1 - Transponders

A transponder is a small and lightweight battery powered wireless device that is installed on the mobile object (motorcycle) within an LPM system thats location position is being tracked. The transponder has an antenna and communicates with the base stations surrounding the area of where the mobile object is traveling, transmitting position information (such as current coordinates) in real time.
Example specifications of a typical transponder used within such an LPM system include:

- *Dimensions: 92 x 57 x 15 mm*
- *Weight: 60 g*
- *Battery: lithium-ion*
- *Antenna: 4 data links*
- *Data Channel: 32 bit (RS-232 / 38,4k)*
- *Frequency: 5,725 Ghz – 5,875 Ghz*

2.2 - Base Station (BS)



BS's are positioned at statically at particular points surrounding an environment/region in which a mobile object can move (for example at particular points around and within a racet**Stottlemyer**rack).

Example specifications of a typical BS used within such an LPM system include:

- *Electrical Connection: 110/230 Volt / 20 Watt*
- *Frequency/Rating: 5,8 Ghz / 25 mW EIRP*
- *Measured Distance: up to 1.000 meters*
- *Weight: 4 kg (approx)*
- *Height: 138 x 180 x 70 mm*

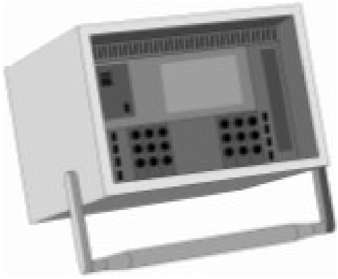
2.3 - Reference Transponder (RT)

The RT within a system has a busy role communicating and collecting the data acquired by each BS within a system, synchronising this data before it is sent to a central hub. The RT behaves as an type of filtering system, collecting and organising (filtering) the data from each BS into a relevant format for both readability and transaction processing.

Example specifications of a typical RT used within such an LPM system include:

- *Rainproof transponder with 360 degree radiator*
- *Synchronisation with all base stations*
- *Weight: 500 g*
- *Height: 190 x 75 x 75 mm*
- *Electricity supply: main supply circuit 230 V to 12 V*
- *Battery: rechargeable Nickel Cadmium (if required)*

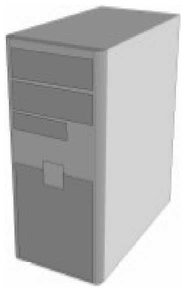
2.4 - Central Hub



The central hub acts as a gateway between the acquired data and the second system principle. The hub is connected to the BS's from which the data is sent, and is additionally connected to a (real-time) positioning calculator.
Example specifications of a typical Hub used within such an LPM system include:

- *Connected with the positioning calculator (real time)*
- *connection with base stations*
- *15 slots per two base stations*
- *height: at the installation of 19" rack (6 HU)*
- *Case: optional (for mobile appliance)*
- *Including: board slots, power supply, HUB-Main*

2.5 - Positioning Calculator



The positioning calculator is the final stage of the fist principle of a LPM system, calculating and preparing the data for its final push into the second principle stage of the system.
Example specifications of a typical positioning calculator used within such an LPM system include:

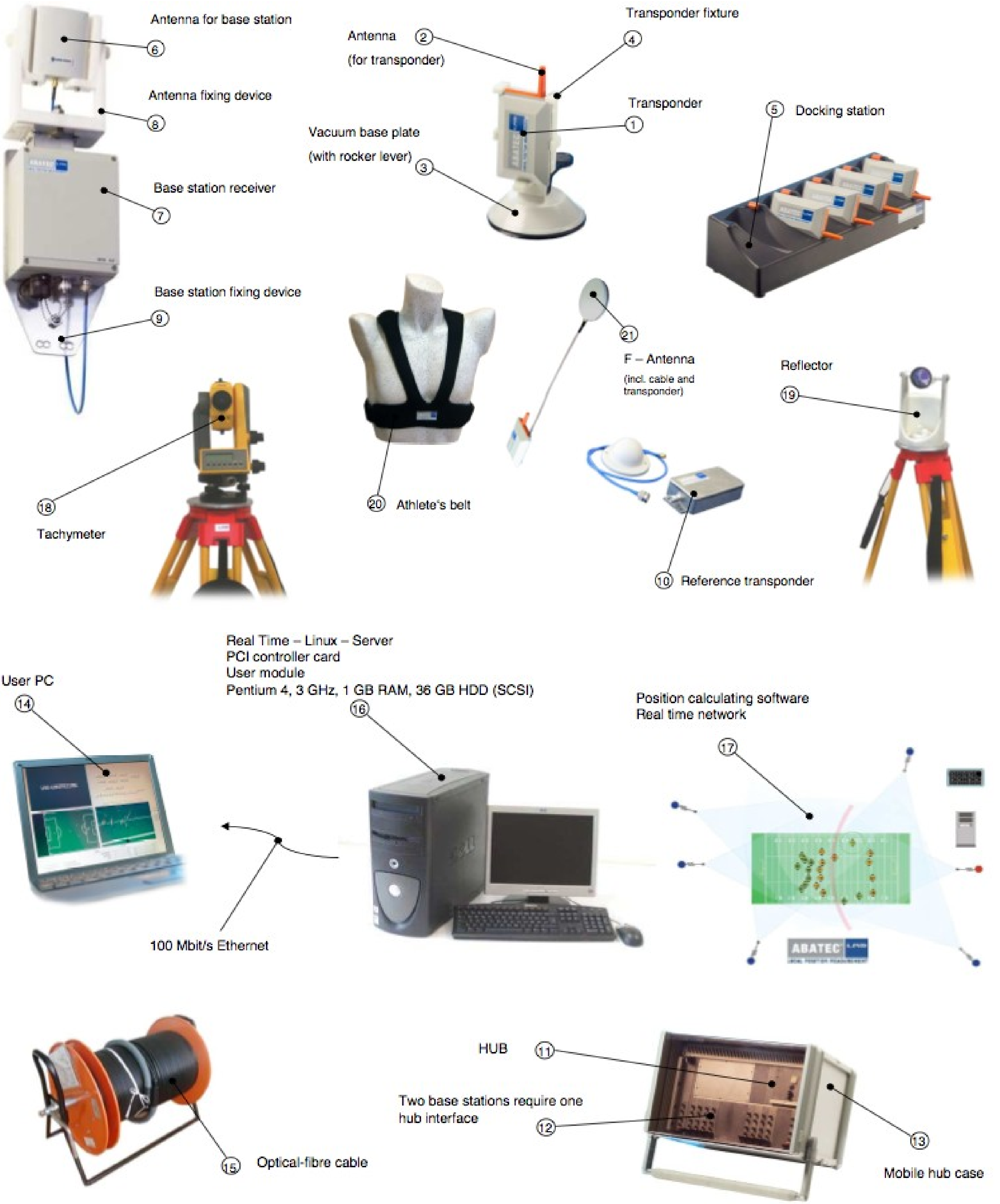
- *Dell Precision 370 (or any PC chassis of a similar specification)*
- *Pentium 4 Hyper Threading; 3 Ghz*
- *1 GB DDR 2 533*
- *PCI LPM Interface-card*
- *Gbit LAN*

[56]-(Abatec Electronics AG, 2008)

3 - Local Position Measurement Components Specifications

This chapter of the appendices includes pictures and specifications of the components that can be and are used within **L**ocal **P**osition **M**easurement (LPM) **S**ensor **B**ased **S**ystems (SBS). The content of this chapter is extracted from a component specification sheet released by Abatec Electronics AG regarding one of there "typical" LPM setups.

This chapter continues from the discussion regarding LPM systems and the individual components within the research and investigation chapter of the report, adding more depth and detail (*as well as additional components and tools*) that are and can be used within such LPM systems:



Item Number	ID Number	Item	Specifications
1	AB 500 5010 103	Transponder with4 antenna terminals	Battery: Rechargeable Li-Ion Performance: max. 5 hours 1 Antennae: not included Weight: 61g / Dimensions: 92 x 57 x 15 mm Data channel: 32 bit/ Frequency: 5,8 GHz
2	AB 501 0010 201 AB 501 0010 301	Standard LPM Antenna straight version Standard LPM Antenna 90 degree version	There are various versions of antenna available. Individual adaptions are available on request
21	AB 501 0010 103	F – antenna incl. 30 cm cable	
3 & 4	AB 501 1010 101	Car mounting system	The mounting system is usually adapted to suit individual customer requirements.
3	5 0900 023	Vacuum base plate with rocker lever	
4	6 0000 128	Milled fixture	
5	AB 500 7010101	Docking station	5 slots: 4 slots for charging 1 slot for charging and programming via RS232 serial interface to PC Power supply: 110/230 V / 30 W Charging time: 2 hours/Dim: 355x125x95mm
6	5 0100 006 5 0100 008	Antenna for base station 36 deg Antenna for base station 65 deg	Antenna36°: 14dBi 3dB beam – width of 36°horizontal / 32°vertical Antenna65°: 11dBi 3dB beam – width of 65°horizontal / 35°vertical Weight: 100gram/ Dimensions: 101 x 95 x 32 mm
7	AB 500 3000 201	Base station receiver	Power supply: 110/230 V / 20 W Frequency/ output: 5,8 GHz / 25mW EIRP Detection range: 0 –500 m Weight: 1,6 kg / Dimensions: 138x180x70mm
8 & 9	AB 500 9000 101	Base station fixing device	Dimensions: 210x 670x145mm Material: POM/stainless steel (electronic unit)
20	5 0900 153	Athlete‘s belt	Material: Neoprene/ Sizes: One Size Weight: 181g (transponder+ 2 antennae+ belt) The belt can accommodate one transponder, as well as 2 antennae on the shoulders.
10	AB 5015 000 101	Reference transponder incl. nondirectional antenna and fixture	Weight: 1500g / Dimensions: 190x75x75mm Power supply: 9V/3W, or battery: rechargeable NiCd fixture: aluminum
11	AB 500 1000 101	HUB	Connection to Linux PC Connection to base stations via optical-fibre cable, 15 slotsfor2 base stations
12	AB 500 1040 141	Hub interface	
13	AB 500 1070 101	Mobile Hub Case	Dimensions: For installation in 19"rack Housing: Optional Including: PCB plug-in unit, power supply
	AB 500 1080 101	Network cabinet(for indoor systems)	2 base stations require one hub interface
14		User laptop	Can be adapted to suit individual customer requirements.
15	2 6150 038	Double break out cable (with PUR coating)	Various optical fibre cable types are available (opened –pre-converted/indoor–outdoor) for individual and object-specific applications
	2 6150 005	Master line standard cable	
	2 6150 013	Mobile field cable	
16 & 17	AB 501 2010 101	Real-time PC Dell Precision 3,0 GHz –Pentium 4 Linux–station including monitor, keypad, position measurement system software and real time networkfor1 base station and 1 r reference transponder	Real-time PC Dell Precision 3,0 GHz –Pentium 4 PCI –Interface Real Time –Linux–Server HP 1502 Flat Screen 1 GB RAM, 36 GB HDD (SCSI)
18	AB 501 3010 101	Tachymeter	Measuring length: With reflector: 3000 m Without reflector: 100 m Standard accuracy: ±2 mm + 2 ppm Dimensions: 173 x 268 x 193 mm Weight incl. battery: 3,500 g without tripod The reflector is used to measure the exact positions of the base stations.
19	AB 501 3020 101	Reflector	

4 - RaceFX System Track Model Generation

The following chapter is extracted from a paper titled "*Real Time GPS FX*" by the author "*Ken Milnes*". The paper was found during the literature search of this project and is considered to be one of the best sources in terms of GPS related tracking systems in relation to the PS.

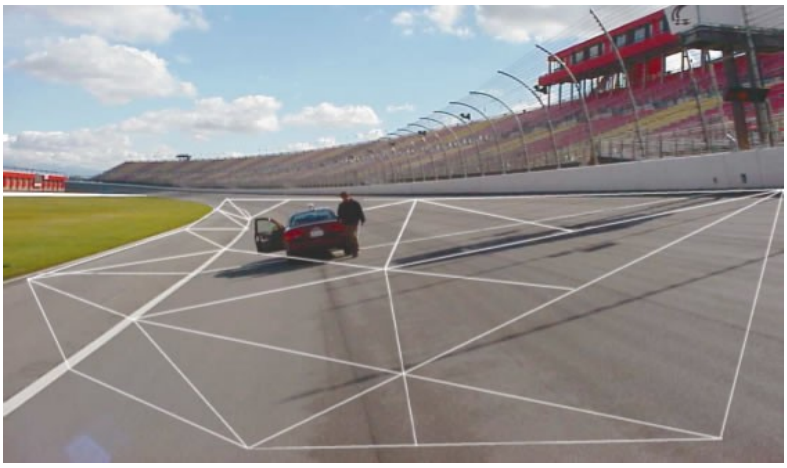
The section of the paper is believed very interesting in regards to future project expansions and possibilities, and for this reason is deemed relevant for extraction and inclusion.

A section describing the creation of track models used within the system has been taken fomr the paper continuing form the discussion relating to the RaceFX system within Research and Investigation section of the report body:

4.1 - Building Track Models

"A track model is a virtual set of planar surfaces that approximate the contiguous surface on which the race takes place. Typically, a NASCAR track can be accurately represented by a series of fewer than 3,000 contiguous triangles defining the planar surfaces of the track. (Each triangle is typically between 5 and 10 meters on a side). Once the GPS reference station-to-track model reconciliation takes place, every point or vertex on the triangles has a position accuracy of 10 centimeters or better (1 sigma), relative to the GPS base station".

"To obtain this, a digital representation of the track surface is created using aerial photogrammetric techniques. High-resolution overlapping aerial photographs taken from approximately 300 meters above the track surface are processed by analytical photogrammetric workstations to create a digital terrain model of the track surface in the local state plane coordinate system. GPS survey receivers are used to obtain ground and camera control for the aerial photography process. Observable features such as paint on the asphalt, fences, walls, buildings, and pavement boundaries are captured as well. The relative accuracy of the track model itself is also within 10 centimeters".



[Figure 66]-(conceptual Track Model Representation)

[Figure 65] "*shows a conceptual representation of the planar sections defining the track model. The fence on the right side of the track plus the grandstand behind it contribute to the signal masking and signal degradation associated with this environment. We will address this problem in more detail later".*

4.2 - Track Model Translation

"Proper operation of the track model GPS software requires that the GPS reference station be located so as to enable removal of any offsets between the track model or aerial survey (based on state plane coordinates) and the GPS reference frame (based on WGS84). The GPS base station coordinates are initially determined from a time-averaged set of single point positions".

"Once the GPS base station is installed, GPS positions for the features captured by the photogrammetry and represented in the track model are measured using real-time kinematic, carrier phase differential techniques. The longitude, latitude, and altitude position offsets between positions of features in the GPS base station frame and in the track model frame are computed for at least three locations. Once a consistent offset is established, the position the GPS base station uses to generate a

differential reference is shifted so that the track model and the differentially corrected GPS positions are in agreement to within 10 centimeters".

"A New Observable in that the accurate three-dimensional model of the track with its contiguous triangles defining the planar surfaces provides an additional observable to the GPS position computations. RaceFX accomplishes this by determining the triangle "within" which a vehicle's GPS antenna is at any moment. As mentioned earlier, each point on the triangle has an accuracy of 10 centimeters. These triangles comprise a planar position grid in which each grid covers an equal area of the track".

"Based on the known, fixed distance between the GPS antenna and the ground in each vehicle, a planar constraint can be defined for each vehicle's antenna with respect to the local planar section. As long as the vehicle operates on the track, the distance to the GPS antenna in a direction perpendicular to each triangle representing the track surface is known. This is quite similar to a height constraint. Once a vehicle is identified with an appropriate triangle, this information provides an observation equivalent to a very accurate additional satellite/receiver range".

"RaceFX determines the correct triangle using the best available vehicle position calculation. This is derived from either the previous pseudorange position or the current epoch's best pseudorange/carrier position, depending on which of two software filters the system is using at the time. Once a position is supplied, the GPS receiver transforms the position into the planar frame of the track model grid. The computer then conducts a sequential search through a linked list of triangles until it finds either the triangle that contains the supplied position, or it determines that none of the triangles contains the given position. When the computer finds a triangle, then it can use the information to generate an enhanced position".

The track model is incorporated into either a least squares pseudorange filter or a Kalman filter that generates RTK carrier ambiguities to provide a constraint for more precise position determination. See the sidebar, "Two Filters and a Track Model."

"The effects of the track model constraints are twofold: they significantly improve position accuracy prior to ambiguity resolution and substantially shorten the time to fixed ambiguity resolution. Taken together, these constraints typically allow the GPS receiver in the vehicle to generate positions on the racetrack with a resultant error of less than 0.5 meters".

[59]-(*K Milnes, 2002*)

4.3 - Two Filters and a Track Model

*"The method used to incorporate the track model into the RaceFX filtering process treats the model as an approximate starting position in a least squares case and as a position observation in a Kalman filter case. Mathematically, the key to this implementation is to generate a variance covariance matrix of the appropriate planar-section position. This matrix is known in the "planar section frame" and is defined with the **x** and **y** axis parallel to the triangle and with the **z** axis orthogonal to it.*
*In the planar section frame, the variance covariance matrix is just a diagonal matrix (**Cp**) with large entries for the **x** and **y** elements and 0.01 meters squared for the **z** element. This is transformed into the WGS84 earth-centered earth-fixed (**ECEF**) coordinate frame by means of the formula".*

***"Ce Rpe Cp Rep** where **Ce** is the variance covariance matrix of the triangle position, **Rpe** is a 3-by-3 rotation matrix that will rotate a vector from the planar section frame into the ECEF frame (Repis its transpose). Then **Ce** can be used either directly in the Kalman filter update equation in the gain computation or its inverse can be used as a weight matrix for an approximate position in the least squares case, as follows:*
Pseudorange Observations - Least Squares Implementation:
***x (A PA (Ce)-1)-1A P** Where **A** is the design matrix appropriate for GPS ranging (direction cosines to satellites in ECEF frame and 1 for range bias) **P** is the weight matrix describing the pseudorange observations **Ce** is the variance covariance matrix of the triangle position is the misclosure vector (difference between measured and computed pseudoranges based on the approximate position and range bias used). In this application, the approximate position is the mean position of the three triangle points offset in the direction of the normal to the triangle by the antenna height. **x** is the correction to the approximate position and range bias Pseudorange Carrier Observations Kalman Filter Implementation **K = PH T (HPH T + C e) -1 x(+) = x(-) + K(z-Hx(-))** Where **z** is the triangle constraint position **x** is the state (position and carrier ambiguities) **H** is defined such that **Hx** is the vector of position entries of the state **P** is the state variance covariance matrix **Ce** is the variance covariance matrix of the triangle position".*

[59]-(*K Milnes, 2002*)

5 - GPS Receiver Fundamentals

Like with the discussed LPM tracking systems, the GPS receivers installed on the mobile objects must be small, compact and lightweight, especially in the case of the PS with the mobile objects (*motorcycles*) travelling at very high speeds, additionally (*within a GPS based system*) requiring the device to be aerodynamic, all with being visible (*having a direct signal*) to the sky and thus the GPS satellites.

A GPS receiver consists of and requires different stages that allow it to operate with the main (*minimum stages*) including:

- **Antenna:** The antenna receives extremely weak satellite signals on a frequency of 1572.42MHz. Signal output is around 163dBW. Some (Passive) antennae have 3 3dB gain.
- **LNA 1:** This low noise amplifier (LNA) amplifies the signal by approximately 15 - 20dB.
- **HF Filter:** The GPS signal bandwidth is approximately 2MHZ. The HF Filter reduces the affects of signal interference. The HF stage and signal processor actually represent the special circuits in a GPS receiver and are adjusted to each other.
- **IF Filter:** The intermediate frequency is filtered out using a bandwidth of 2MHz. The image frequencies arising at the maximum stage are reduced to a permissible level
- **Signal Processor:** Up to 16 different satellite signals can be correlated and decoded at the same time. Correlation takes place by constant comparison with the C/A circuit code. The HF stage and signal processor are simultaneously switched to synchronise with the signal. The signal processor has its own time base (**R**eal **T**ime **C**lock (RTC)). All the data ascertained is broadcasted (*especially the signal transit time to the relevant satellites determined by the corrector*), and this is referred to as source data. The signal processor can be offset by the controller via the control line to the function in various operating modes.
- **Controller:** Using the source data, the controller calculates position, time speed and course. It controls the signal processor and relays the calculated values to the display. Important information are decoded and saved to the RAM. The program and the calculation algorithms are further saved in ROM.
- **Input:** Most (although not all) GPS receivers offer some sort of user input allowing a user to select the coordinate system that is to be used and its parameters (*for example the number of visible satellites*) should be displayed.
- **Power Supply:** the power supply delivers the necessary operational voltage to all levels of electronic components.

[38]-(J Zogg, GPS Basics)

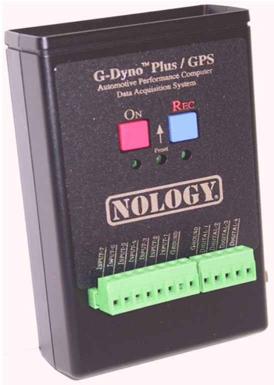
As discussed GPS receivers (*in most cases*) have to receiver and evaluate (sometimes) very weak signals form at least four satellites to enable the determination accurate 3-Dimensional location information. Time signals are (*sometimes*) further emitted along with the sent longitude, latitude and height values, and is further synchronised using **U**niversal **T**ime **C**oordinated (UTC).
GPS receivers issue information on the basis of the constellation, satellite and signal health and the number of visible satellites at any one time. The location and time value variables that the receiver has calculated can be further used to calculate and determine additional and relevant variables such as object speed distance and acceleration values.
A GPS receiver pre-amplifies and transfers the signals that is received into a lower and intermediate frequency, and the necessary carriers wave to enable the frequency converter as well as the relevant clock frequency required by both the processor and corrector are provided by the reference oscillator. The intermediate frequency is further converted from an analogue signal into a digital signal using a 2-bit ACD.

PRN pulse sequences are used to ascertain the signal transit times from satellite to a GPS receiver, and the satellite PRN sequence must therefore be used to determine this time or or else no maximum correlation will exist. Data loss can be recovered using RPN sequences by mixing the data with the correct sequence, and at the same time the signal determined as useful is applied to a greater level than the level of interfering existing. Up to 16 GPS satellite signals can be processed at any one time (simultaneously), with the ".. control and generation of the PRN sequences and the recovery of data carried out by a signal processor. Calculating and saving the position, including the variable derived from this, is carried out by a processor with a memory facility" **[38]-(J Zogg, GPS Basics)**.

Further relevant information regarding GPS receivers can be found at **[38]-(J Zogg, GPS Basics)**, **[57]-(S Kennedy et al, 2008)** and **[85]-(E Tennant, 2005)**.

6 - Investigated Data Acquisition Device photos

6.1 - Nology G-Dyno Plus



6.2 - RacePak G2X Data Logger

6.3 - Velocity Racing Data DL1

6.4 - TraqMate GPS Data Acquisition

7 - Programming Approaches

There are several directions computer programming has taken over the years, two of the newer (*more commonly used*) and popular approaches relevant to the PS include:

- Structured programming
- Object oriented (OO) programming

7.1 - Structured Programming

Structured programming follows a methodology that programming should pursue a well thought and documented structure. Its aim is to simplify, tidy and make code understandable improving both conceptual and organisational programming aspects. Structured programming includes concepts to provide a framework for organising programs and thinking about the way that they flow.

[84]-(E Walters, 2000)

The "real" meaning of a structured language is to control the flow of a program. Flow control often falls into the following categories:

- **Sequential flow control** - Once a statement has been executed the next is executed (*top down approach*).
- **Selection flow control** - When a program reaches a particular point where alternative options can be selected. Generally conditional statements are used to control the route through the selections.
- **Repetition flow control** - Operates a set of data in which each individual element of the set must be processed. Loops are generally used.

Structured programming languages perform specific tasks (*printing text and performing calculations*), allowing program structure to be divided into a number of tasks, thus making programming more "straightforward", however, it is believed that although more straightforward, for larger applications the use of a structured based language can get both very large and complex in areas, therefore being not so "straightforward" in comparison to a more OOP controlled language/environment).

Depending on the language used, the functions can be provided either by the manufacture (*pre-defined functions*) or by the programmer (*user-defined functions*). The programming language best known for its structured approach is "C".

Strengths	Weaknesses
• Organised and documented structure • Easy to understand and follow code • Framework allowing consistent Application flow and control • Division program structure into tasks	• Difficult to model problems • Less code re-use (*less efficient in comparison to OO*) • Redundant use of code • Can get complicated for complex applications

7.2 - Object Orientated Programming (OOP)

OOP is neither a method nor procedure thus (*sometimes*) creating complications in understanding its concept. There is no set rule to define how it works, it relies on having a conceptual framework created by thinking in a certain manner.

OOP consists of the following modules / concepts:

- **Object** - a software bundle of related variables and methods. Often used to model real-world objects.
- **Message** - software objects use messages to interact and communicate with one-another.
- **Class** - defines the different variables and methods that are common to objects of a certain kind – (*Objects are instances of classes*)
- **Inheritance** - provides a powerful and natural method for organising and structuring software programs

OOP is a powerful technique that can manage complexity. It provides a smooth progress of decompositions allowing more complicated applications to be created. OOP tools and techniques are used to break down large (*complex*) problems into smaller more manageable ones.

An advantage that OOP has over structured programming techniques is that it is popular among programmers allowing modules to be created that don't have to be changed or adapted when a new object is added. New objects can be created using features from existing objects making OOP easier to modify.

[84]-(E Walters, 2000)

Although, there are a lot OOP languages, and many existing and developing (*structured based*) programming languages are being upgraded to cater for OOP techniques due to its power and ever increasing popularity for application development.

Strengths	Weaknesses
• Efficient code re-use • Easy to find and fix problems • Very powerful • Great / efficient use for complex applications • Organised break down of problems • Easy / shared object modification • Easy maintenance and extendibility possibilities	• Complex structure to initially understand • Follows a conceptual framework

8 - Simple LabVIEW Application Development Walkthrough

This section contains a walkthrough of the development process when creating a simple LabVIEW application, allowing a “general” feel of the process (steps) required when creating an application using a DFP language approach. The development of a simple application has been completed, taking a basic DP and using it to provide data to, and render a selected visual component (a bar chart).

LabVIIEW is not just a language but a complete application that controls a complete development cycle of an application, creating both back-end “foundation” to front-end data rendering as a complete project.
LabVIEW applications are created using a “visual” approach consisting the generation of block diagrams used to represent data flow and control (input, output and the execution order of an application) with each block representing code that can be interacted with and controlled through various built in “interaction dialogues”. For example a DAQ block would need to know where and how to get and receive data, what format or type this data is and how to processes and handle such data when it is received (for example, limitations and filters can be set such as the minimum and maximum values the block will accept). Such information would be configured within the described dialogues.
Once the example DAQ block has been configured, LabVIEW can connect to the set data source and thus retrieve the relevant data. Different blocks can be built, configured and added to an application, additionally being linked (*connected*) together as and where relevant, all being created and configured within the block diagram.
As described, further (*additional*) visualisation blocks can be easily added and configured to the diagram, thus making application extensions an easy process.
The added a visualisation block(s) could be a **D**ata **P**rovider (DP) for an existing visualisation component (such as bar chart) requiring a (*new*) DP, and can be further configured to use a specific measurement from the previously configured DAQ block (*acting as a DP*) being configured (*again through a built in dialogue*) to show the averages of this data with the configured averages being rendered in the bar chart (*visual component*).
Within LabVIEW, data flow is represented by the flow of execution, thus in the case of the discussed block diagram, the visualisation block is required to be linked (*wired*) to the DAQ block to tell the data to be pushed from the DAQ block to the visualisation block (*thus the DAQ block acting as the visualisation blocks DP*).

Within the above example, A block chart has been created consisting of a DAQ block (acquiring *data from a specified source*) and a visualisation block that is connected to the DAQ block and has been configured to select a specific data measurement the DAQ block and to calculate the average of this measurement(s).

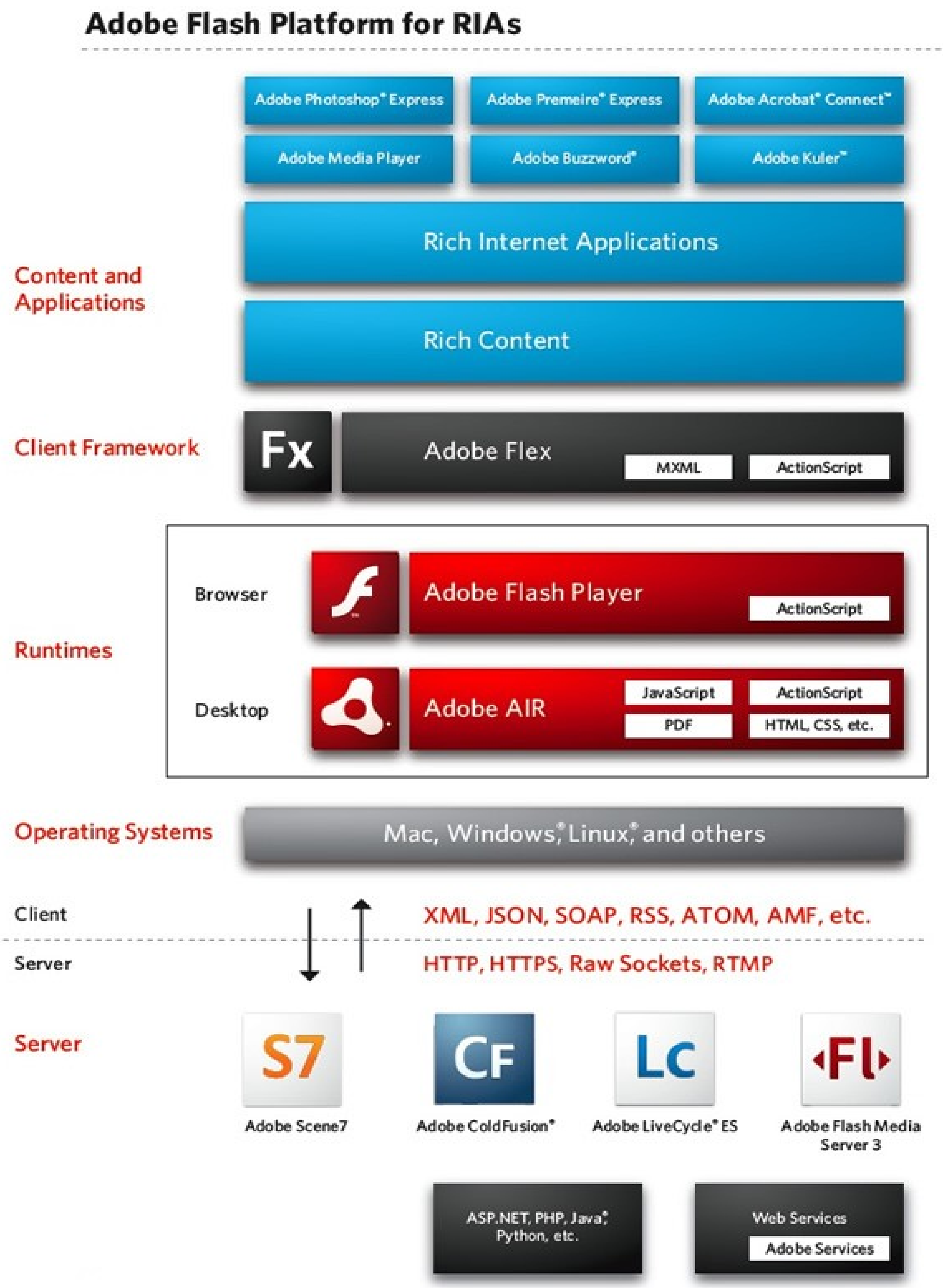
Labview has, as well as the block chart (*back-end*) development view, and front end view consisting of a canvas, where specific LabVIEW components can be added to the canvas (*in a drag and drop manner*) and further positioned, scaled and skinned accordingly thus creating an advance **U**ser **I**nterface (UI). Such a component could be that of a specific chart (*Area, Bar, Pie, Line*) and can then be linked/connected to a specific back-end block to use as a DP. For example, an Area chart could be added to the canvas, and linked to the discussed DAQ block using rendering the data from this blocjk within and Area chart visualisation, additionally adding another (*output*) block to the block chart (*from the visualisation block to a new block representing the actual chart*).

LabVIEW also contains relevant programming functionalities such as loops, including for loops and while loops that can be simply wrapped around blocks so they are continuously ran until the end of the loop is reached, or (*in the case of an endless loop*) a stop button is clicked.
Such programming functionalities are however also controlled by a UI (dialogue) and added to the objects within the block diagram (*a for loop is represented as a box, and all of the items within the box are therefore contained within the loop*).

Once all data has been acquired and the visualisations (*and other relevant user interface controls have been set up*) the application can be ran with the included visualisations rendering the data from the DAQ block (*live or static*) relevantly.

Further functionalities and controls can be added to such an application by simply extending the block diagram and adding additional components to the application and the application DP's and renderers.

9 - Adobe Technology Platform for RIA's

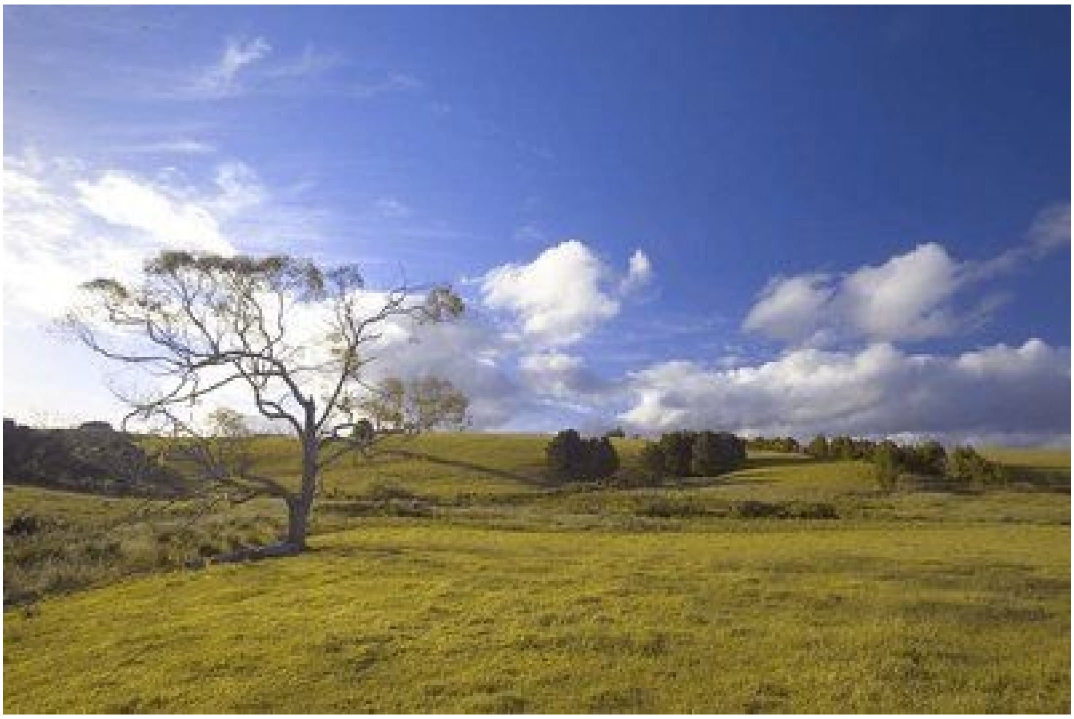


(from: http://www.adobe.com/devnet/actionscript)

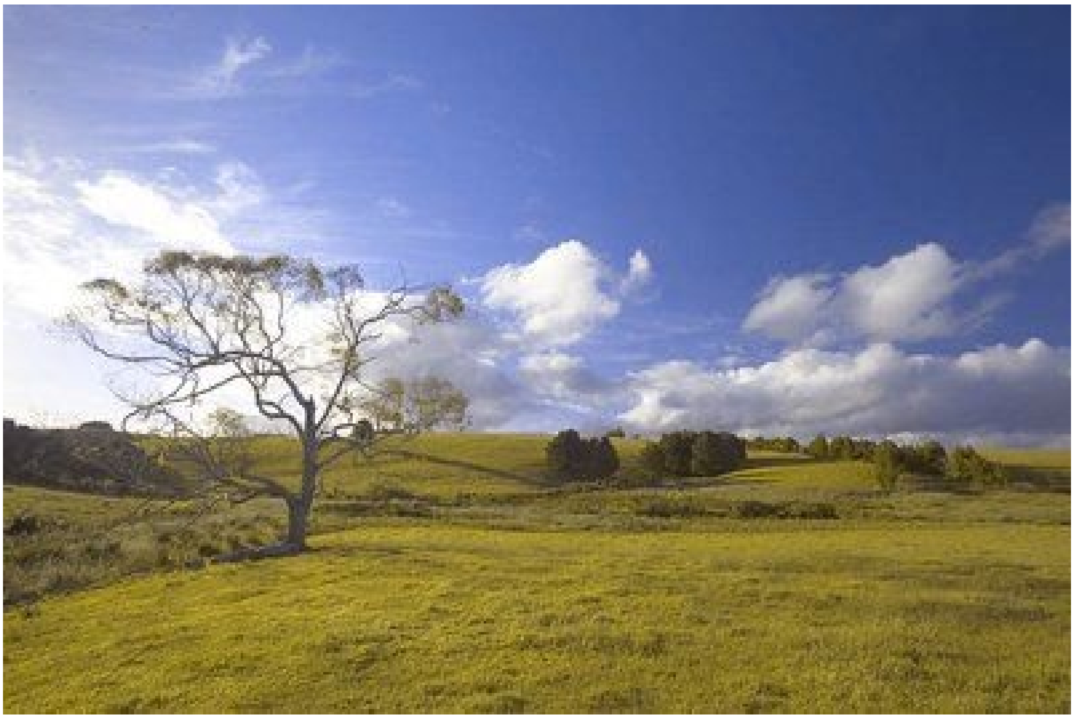
10 - Colour Blindness Image Examples

10.1 - Normal Image

10.2 - Deuteranope Image



10.3 - Protanope Image



10.4 - Tritanope Image

11 - Further Human Computer Interface Considerations

For the development of a successful and usable application certain HCI issues need to be taken into consideration.

"*Human-Computer Interaction (sometimes also referred to as Computer - Human Interaction, CHI) is the study of how people interact with computers and to what extent computers are or are not developed for successful interaction with human beings*" **[92]-(The Usability Company, 2005)**.

HCI is concerns lie with the design, evaluation and implementation of interactive applications for human use. Its goals are to develop or improve the safety, utility, and effectiveness of computer based applications.

To create a good usable application the context and setting needs to be understood in terms of:

- Who will the application be used by (*target audience*)?
- What tasks will the user be doing?
- What is the users environment like?
- What tools and techniques are available to develop the application?
- What technology is feasible?

The challenge for designers is the knowing what can functionally be done, and the manner that it should be carried out to meet the users needs (*usability*).
The appropriate input and out put tools should be chosen, an the best interaction style should be chosen (*form based, graphical user interface (GUI), multimedia, or virtual reality (VR)*).
The user fit with the application should be looked at, - how the users improvise the application, how the application should adapt to the user (*customisation*) and how the user adapts to the application (*training / frequent use, ease of learning*). The application should include the relevant guidance options (*help, error handling, documentation*) all-aiding in the overall usability of the application.

The use of colour used within an application should be appropriate to the and easy to view (*not causing any strain on the eye*). When used appropriately colour can be effective and aid in the usability of an application (*for example highlighting or giving certain feelings*). No more than 7 colours should be used within an interface and the use of colour should be consistent throughout. To enhance accessibility colour could be put under users control / preferences.

The navigation of an application should be simplistic and easy to use / understand. Links and menus should be positioned and grouped accordingly and should be clear and easy to understand. Titles and locations statuses (*highlighting*) may help users know where they are within an application.

General HCI issues that can effect / improve a systems usability include:

- Accessibility
- Interaction

General methods used for creating more usable systems include:

- Task Analysis and User Profiling
- User centred design
- Contextual design (*Ethnography*)

Other areas for further considerations include:

- Globalisation (for applications on a larger scale)
- Human Factors (*Ergonomics*)

11.1 - Accessibility

Systems should be made to be as accessible as possible. In the case of HCI, accessibility involves making a system as easy to use as possible taking into account different users skills and abilities.
Systems should be as user adaptable as possible to have a good accessibility level. For example allowing the user to choose a colour scheme or other options to suite their use.
System users can have disabilities and van therefore registered under the **D**isability **D**iscrimination **A**ct (DDA). It is an unlawful activity for a service provider to discriminate (*within reason*) by failing to provide a serve, not making reasonable adjustments, providing and worse service or providing the service under worse terms (*extra costs*). The Internet is a service so this also applies to the web.
Interface designs need to consider various visual impairments that users may have, for example colour-blindness and cater for these accordingly using relevant colours and shapes that will not be

distracting or confusing. Aging has effects on people that are not defined within the DDA including:

- Sensory effects
- Cognitive effects
- Motor effects

Aging affects users focus, acuity, depth perception and has greater sensitivity to glare. Relevant fonts and sizes should be used within an interface, moving text should be avoided, glare and brightness changes should be avoided, and shifts in focal distance should be avoided.
Aging affects users attention and memory. Users find it harder to maintain memory over large periods of time and page scanning is more difficult. Over detailed pages and extra 'noise' should be avoided, more learning tome should be allowed, and demands on **S**hort **T**erm **M**emory (STM) should be reduced (*the use of menus/labels*).
Aging affects co-ordination. Users have longer response times and things like target tracking / mouse manipulation is more difficult. Advanced mouse control (*scrolling, dragging and dropping*) should be avoided and single clicking rather than double clicking should be enabled.

11.2 - Interaction

Interaction is a very important aspect of an application. It allows users to control it by manipulating screen objects. All aspects of applications interaction methods need to be considered and chosen upon carefully.

Common screen objects used within applications include **Menus**, **Forms** and **Dialogue** windows.
There are many different types of menus available (*single, scrolling, iconic and hierarchical*) each having there own advantages and disadvantages within varying applications. Menus should not be overcomplicated (*especially when using hierarchical menus*). They should by logically grouped and use relevant terminology.
Form filling is a good data entry method. It is easy to use with a tabbing effect moving between the different fields therefore giving user confidence. The movement around the form should be logical, neat and the instructions understandable. Relevant titles and field names should be used, relevant error checking / prevention should be implemented and compulsory fields should be highlighted.
Dialogue boxes are pop-up boxes displaying messages to the user. They should be meaningful and consistent throughout an application. They should use standard buttons (*ok, cancel*) and should be clear how to complete.

There are two types of manipulation, direct and indirect manipulation. An example of indirect manipulation is command languages or using a text editor to manipulate a HTML file, and an example of direct manipulation is clicking directly on an object to carry out an operation. Most applications follow a direct manipulation route. Operations and functionaries are easily picked up, even for a novice user with the intermittent user being reminded of the functionality, and visual feedback is available.

I/O devices should also be considered and chosen appropriately. They should be appropriate for the tasks that need to be performed, and they should be relevant/suitable for the environment they aimed for. Input devices are separated into two areas:

1. Discrete entry devices (*keyboards*)
2. Continuous entry devices (*mouse, joysticks, tracker balls*)

Output devices include technologies such as monitors, speakers and printers all providing relevant output from an application to its user(s).

11.3 - Task Analysis and User Profiling

The purpose of task analysis is to describe what users do to complete their tasks, it focuses on the user views rather that on the data views.
Users have goals and tasks when using an application. A goal is there overall aim, and task is steps taken for them to achieve their goal. The users tasks need to be analysed to improve the usability on an application therefore allowing it to be as the user expects it to be.

There are many different task analysis methods available some of the more general ones being:
- User Task analysis -focuses on observing users performing there tasks. Considered better than

questionnaires as 'users don't always do what they say they do'

- GOMS Modeling - (*Goals, Operators, Methods, Selection Rules*). Defines what the user wants to achieve, there actions taken to achieve there goals, there procedures for accomplishing there goal, and method chosen to complete there goal (*if more than one available*).
- Hierarchical Task Analysis (HTA) -decomposes takes into sub-tasks and provides a simple description of the sequence of sub-tasks needed to complete a task.

HTA is a well-established approach and the proposed application would benefit from. It is visually easy to understand, can capture some cognitive aspects of tasks and is flexible.

The gulf of execution is the difference between a users expectations of haw an action should take place (*be completed*) and the action(s) that are actually allowed by the application. A good interface design should keep the gulf of execution to a minimum.
Card sorting can be used where users can sort cards (*containing various concepts*) into similar piles then naming these piles. This allows designers to view a rough outline / structure from a users perspective.
Users can be surveyed either by the use of questionnaires or interviewing which can uncover certain concerns. When using Questionnaires design is important. The questions should be kept short, be relevant, 'closed' ease of analysis. Further interviewing can explore in more detail responses to questions.

Scenarios can be used to capture certain requirements and communicate with designers. They describe available interactions within an application and can be written by either users or usability experts.
Ethnographic observation is observing real users in their real working environment performing real work tasks.

11.4 - Contextual Design

The focus of contextual design is to address the customer needs, decide what a system, should / needs to do, and how a system should be structured.
Contextual design is divided into seven parts known as **INCREMP**:

- Contextual Enquiry – talking to the users in there own working environment
- Work Modeling – provide diagrammatical models of individuals and organisations.
- Consolidation – the bringing together of data captured from interviews looking for relevant patterns and structures.
- Work Re-design – storyboards / frame sketched showing how users will perform tasks in a new context of a system.
- User Environment designs – showing the structure, function and flow of a floor plan for a new system.
- Mock-up and Testing – user interface design produces on paper (paper mock-up) and used for testing with users.
- Putting into practice – implementation

11.5 - User Centered Design

To be able to design for the target users of an application you need to know them and there requirements. If users are satisfied when using a system then this shows a good design.
User centered design involves the user in the design from the beginning (*participatory design*) and they are in constant communication with the designer(s). User inputs / changing concerns are adapted into the design.

The advantage of including users within a design process is that they can react, give there opinion to suggested system design, and they are going to be the final users of the system so it can be designed how they want it to be, thus using there inputs.
The disadvantage of user participation within the design process is that it can sometimes be costly (*pay them for there tim*e) and they may be reluctant to take part. End-users are not expert designers so might not understand certain design concepts needed, and the user 'customer' is not always right and don't always know what they want.

11.6 - Globalisation

Globalisation is to do with the internationalisation and localisation of a system. It generally applies to larger systems that are produced for use on an international level. They are designed to be deployed at the first step internationally (*internationalisation*) and then modified for each relevant locale / country (*localisation*).

Globalisation takes into account various cultural aspects such as:

- Colours – colours have different meanings in different countries. Red may have no meaning in one country yet mean death or even be offensive if used inappropriately in another
- Flags – flags are generally used within applications for language selections. This should be avoided as some flags contain more than one language (Canada – English or French or Brazil – Spanish or Portuguese), which do you use?
- Icons - icons my have different meanings in different countries or may not be recognisable in others therefore be meaningless
- Formats – different countries use different formats / variables for things such as date, time, addresses, phone numbers, clothes sizes, measurements to name a dew.

Language also needs to be considered during globalisation as different countries use different languages, and it is not always as simple as just translating the words.
During translation word expansion and contraction occurs i.e. words are sometimes longer (*contain more letters*) in one country or a lot smaller in the case of China/Japan representing one symbol per word. This expansion and contraction can change the format of text, buttons and general page layout causing potential problems.

The users in the different countries need to be designed for. During globalisation it is not appropriate to advertise holiday breaks or celebrations if they are not taken or celebrated in a specific country, as it would not be understood.

11.7 - Human Factors

Issues concerning human factors include technologies used and working environments. There are millions of users affected by human factor issues each year, main causes resulting from fatigue and stress due to inappropriately designed working areas and environments:

Bad **V**isual **D**isplay **U**nits (VDU) can cause both physiological (*task control*) and physical (*eye-strain*) problems, and keyboards can cause muscular problems such as frozen shoulder, **R**epetitive **S**train **I**njury (RSI) tendonitis, spine and neck problems and circulation problems if used incorrectly or too much.
RSI is a common strain caused by computer interaction. It is caused by constant repetition of the same activity and can cause long-term damage. The keyboard, as well as other input devices such as mouse and joysticks is the main causes of RSI.

Poor lighting can lead to fatigue, stress, productivity decrease, eyestrain and headaches. The lighting within an environment should be sufficient and controllable. Glare should be kept to a minimum with "up-lights" being preferred as they produce fewer glares than "down-lights". Other environmental areas that can affect human factors include:

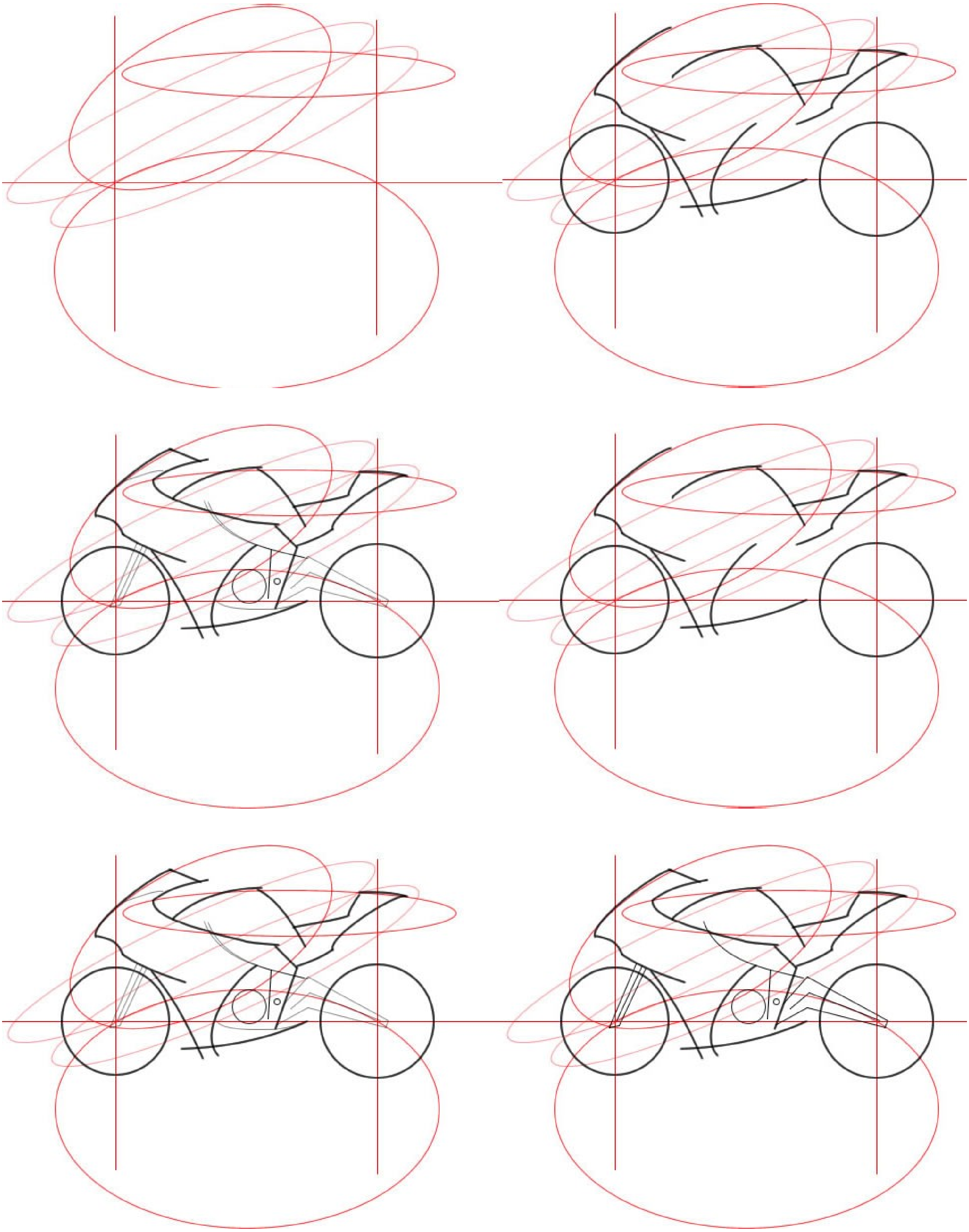
- Static electricity
- Heat and humidity
- Noise
- Personal space

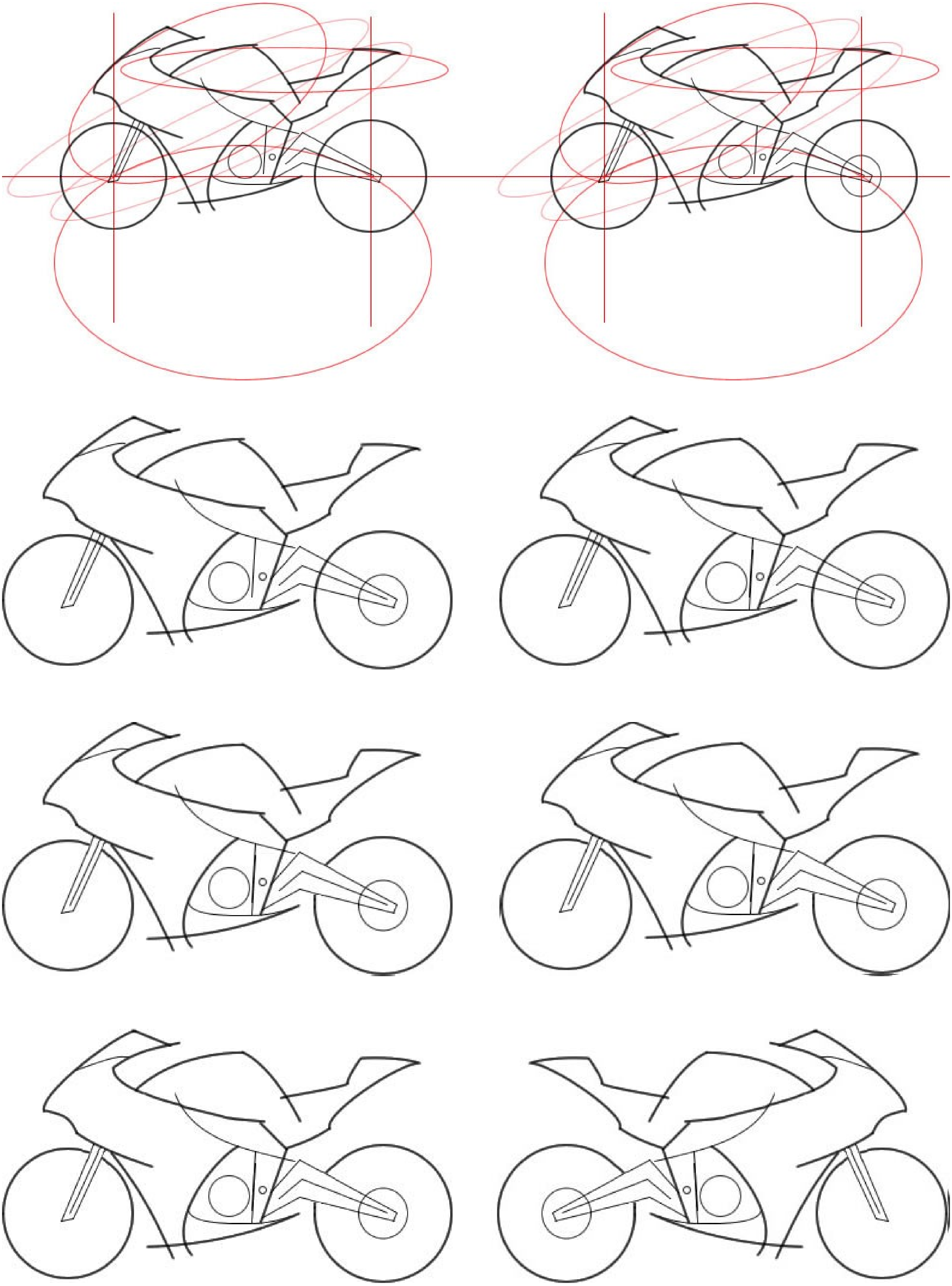
Applications should not put extra stress / strain on users preventing stressors:

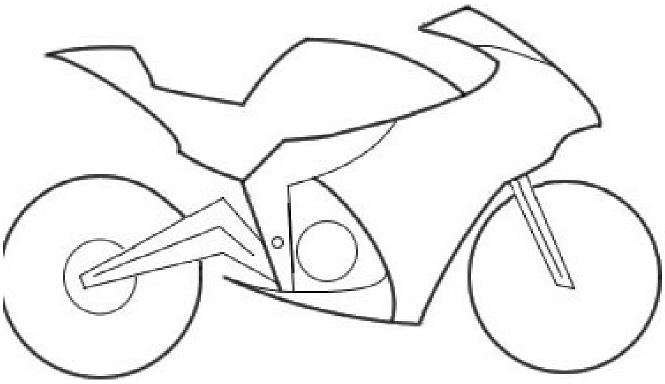
- Physical – lifting heavy objects
- Cognitive – sensory overload / under load
- External – environmental

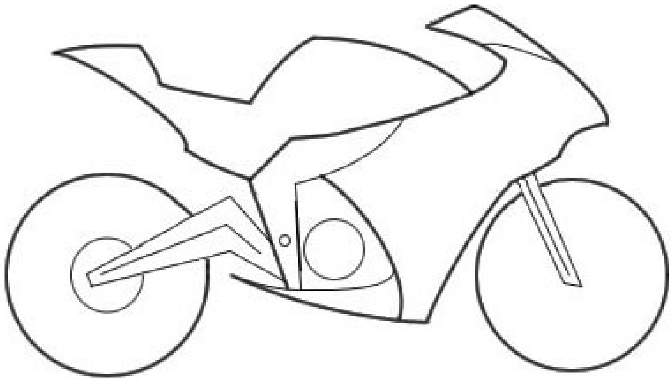
Things such of bad use of colour is a good example of an application interface that can cause a stressor to occur.

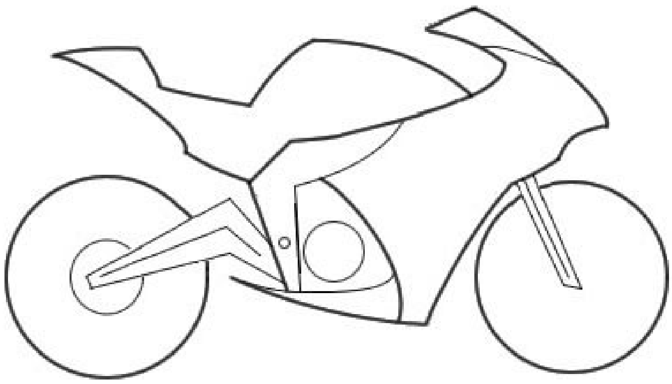
12 - Application Logo Design Process

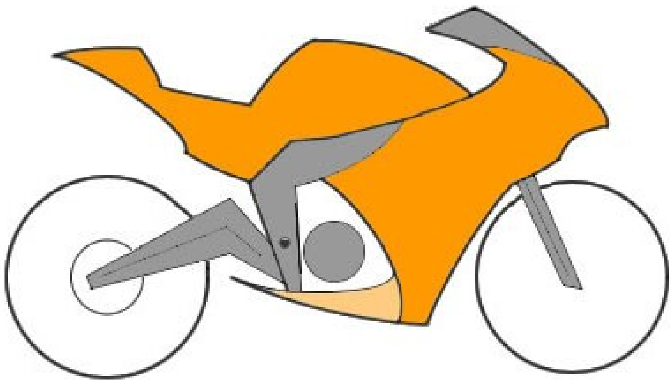


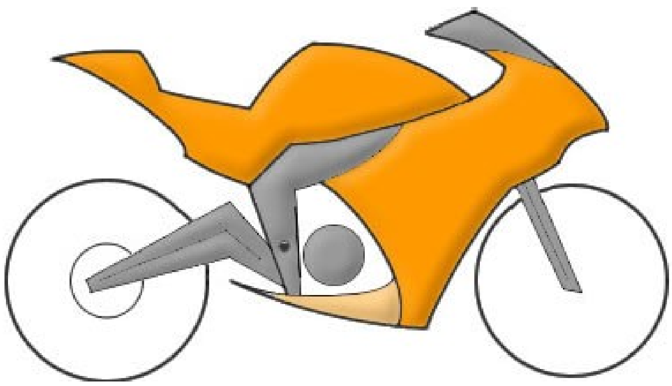












13 - Sample DAQ Application Data

Time	Lap	Lap Fraction	Dist to Wall	Lat	Lon	North	East	Alt	Speed	Heading	Steering	Linear Accel	Lateral Accel	Vert Accel	Position Sta	Num Sate	RPM	Brake	Throttle	Gear	Fuel
1207596267.80	21	0.99	56.4	27.77	-82.63	1247910.12	452717.77	-22.55	139.57	242.5	-5.6	0.19	0.31	0.98	0.1	7	9487	0	99	5	13.1
1207596268.00	21	0.99	56.8	27.77	-82.63	1247891.45	452681.08	-22.55	140.8	242.7	-2.8	0.06	0.31	1.05	0.1	7	9627	0	99	5	13.1
1207596268.20	21	1	56.8	27.77	-82.63	1247872.74	452643.98	-22.52	142.08	243	-5.6	0.06	0.31	1.05	0.1	7	9759	0	100	5	13.1
1207596268.40	22	0	57.4	27.77	-82.63	1247854.1	452606.42	-22.48	143.33	243.5	-8.4	0.19	0.25	0.99	0.1	7	9689	0	100	5	13.1
1207596268.60	22	0	58.7	27.77	-82.63	1247835.71	452568.35	-22.46	144.49	244.1	-8.4	0.38	0.31	1.03	0.1	7	9874	0	99	5	13.1
1207596268.80	22	0.01	60.5	27.77	-82.63	1247817.56	452529.79	-22.4	145.64	244.5	-5.6	0.25	0.31	0.98	0.1	7	9940	0	99	5	13.1
1207596269.00	22	0.01	62.9	27.77	-82.63	1247799.47	452490.83	-22.4	146.71	244.7	-1.4	0.06	0.25	0.98	0.1	7	10046	0	100	5	13.1
1207596269.20	22	0.02	65.4	27.76	-82.63	1247781.24	452451.62	-22.36	147.54	244.5	1.4	-0.13	0.25	1.05	0.1	7	10040	0	99	5	13.1
1207596269.40	22	0.02	67.5	27.76	-82.63	1247762.76	452412.3	-22.32	148.34	244.1	1.4	-0.31	0.19	0.96	0.1	7	10094	0	99	5	13.1
1207596269.60	22	0.03	69	27.76	-82.63	1247743.67	452372.98	-22.31	149.06	243.4	5.6	-0.44	0.19	0.99	0.1	7	10204	0	100	5	13.1
1207596269.80	22	0.03	69.6	27.76	-82.63	1247724.13	452333.71	-22.27	149.78	242.9	1.4	-0.38	0.19	0.97	0.1	7	10197	0	100	5	13.1
1207596270.00	22	0.04	69.3	27.76	-82.63	1247704.15	452294.35	-22.28	150.56	242.5	1.4	-0.25	0.19	0.99	0.1	7	10309	0	99	5	13.1
1207596270.20	22	0.04	68.8	27.76	-82.63	1247683.88	452254.94	-22.24	150.42	242.1	1.4	-0.25	0.19	1.03	0.1	7	10288	0	99	5	13.1
1207596270.40	22	0.05	67.7	27.76	-82.63	1247663.42	452215.94	-22.22	146.33	241.6	1.4	-0.38	-0.88	0.94	0.1	7	9920	11	19	5	13.1
1207596270.60	22	0.05	66.9	27.76	-82.63	1247643.55	452179.02	-22.22	138.69	241.1	-1.4	-0.38	-1.88	0.96	0.1	7	9469	23	3	5	13.1
1207596270.80	22	0.05	66.2	27.76	-82.63	1247624.59	452144.31	-22.23	130.71	240.8	-1.4	-0.25	-1.81	1.03	0.1	7	8870	24	2	5	13.1
1207596271.00	22	0.06	65.3	27.76	-82.63	1247606.57	452111.74	-22.2	122.86	240.5	-2.8	-0.13	-1.69	1.04	0.1	7	8902	23	25	4	13.1
1207596271.20	22	0.06	65.6	27.76	-82.63	1247589.57	452081.21	-22.18	115.15	240.3	-4.2	-0.19	-1.81	0.97	0.1	7	10295	22	13	3	13.1
1207596271.40	22	0.07	65.9	27.76	-82.63	1247573.53	452052.72	-22.17	107.95	240.2	-5.6	-0.13	-1.63	0.97	0.1	7	9505	21	3	3	13.1
1207596271.60	22	0.07	66.3	27.76	-82.63	1247558.49	452025.94	-22.17	101.15	240.4	-5.6	0	-1.56	1.03	0.1	7	9118	17	2	3	13
1207596271.80	22	0.07	66.9	27.76	-82.63	1247544.6	452000.82	-22.14	95.02	240.8	-5.6	0.13	-1.44	1	0.1	7	10080	16	17	3	13
1207596272.00	22	0.07	67.4	27.76	-82.63	1247531.59	451977.04	-22.14	90.14	240.8	-1.4	0.06	-1.06	0.98	0.1	7	9063	9	2	2	13
1207596272.20	22	0.08	67.9	27.76	-82.63	1247519.15	451954.39	-22.13	86.05	240.8	0	-0.06	-0.94	1	0.1	7	8965	8	2	2	13
1207596272.40	22	0.08	69.8	27.76	-82.63	1247507.23	451932.75	-22.13	81.72	240.7	0	-0.13	-0.88	1.03	0.1	7	8645	8	2	2	13
1207596272.60	22	0.08	72.4	27.76	-82.63	1247495.96	451912.36	-22.1	76.49	240.8	-5.6	0	-1.19	1	0.1	7	8038	14	2	2	13
1207596272.80	22	0.09	74.9	27.76	-82.63	1247485.61	451893.36	-22.1	71.32	241	-7	0	-1.25	0.98	0.1	7	7317	13	2	2	13
1207596273.00	22	0.09	77.4	27.76	-82.63	1247475.95	451875.56	-22.09	67.06	241.1	-2.8	0.06	-0.94	1.03	0.1	7	6890	10	2	2	13
1207596273.20	22	0.09	78.6	27.76	-82.63	1247466.83	451858.73	-22.07	62.81	241.6	-5.6	0	-0.94	0.98	0.1	7	6637	9	2	2	13
1207596273.40	22	0.09	79.3	27.76	-82.63	1247458.66	451842.97	-22.08	58.25	243	-22.5	0.25	-1.13	0.99	0.1	7	6075	14	2	2	13
1207596273.60	22	0.1	80.5	27.76	-82.63	1247451.51	451828.13	-22.07	54.69	244.6	-30.9	0.38	-0.81	1.02	0.1	7	5747	8	2	2	13
1207596273.80	22	0.1	79.6	27.76	-82.63	1247445.09	451813.84	-22.07	52.46	245.9	-15.5	0.25	-0.44	1.02	0.1	7	5482	4	2	2	13
1207596274.00	22	0.1	75.1	27.76	-82.63	1247439.13	451799.9	-22.05	50.95	247.5	-29.5	0.25	-0.31	0.98	0.1	7	5368	2	2	2	13
1207596274.20	22	0.1	70.5	27.76	-82.63	1247433.81	451786.1	-22.07	49.39	250.2	-43.6	0.38	-0.38	0.97	0.1	7	5315	2	2	2	13
1207596274.40	22	0.1	66.2	27.76	-82.63	1247429.46	451772.53	-22.08	47.49	253.9	-60.5	0.63	-0.44	1.01	0.1	7	5083	4	2	2	13
1207596274.60	22	0.1	62.6	27.76	-82.63	1247426.23	451759.23	-22.1	45.44	260.4	-81.6	0.75	-0.38	1.09	0.1	7	4926	4	2	2	13
1207596275.00	22	0.11	56.2	27.76	-82.63	1247422.99	451732.97	-21.94	44.29	265	-95.6	0.69	-0.19	0.9	0.25	7	4582	2	3	2	13
1207596275.20	22	0.11	53.8	27.76	-82.63	1247423.05	451720.28	-22.1	42.61	272.9	-105.5	0.88	-0.38	0.93	0.25	7	4623	2	3	2	13
1207596275.40	22	0.11	52.1	27.76	-82.63	1247424.4	451707.96	-22.09	41.45	278.7	-98.4	0.94	-0.25	1.16	0.4	7	4365	3	2	2	13
1207596275.60	22	0.11	51	27.76	-82.63	1247426.83	451696.2	-22.14	40.6	284.9	-112.5	0.94	-0.25	1.01	0.4	7	4378	1	3	2	13
1207596275.80	22	0.11	50.6	27.76	-82.63	1247430.64	451684.92	-22.13	40.77	291.5	-115.3	0.88	0.06	1.05	0.55	8	4377	0	7	2	13
1207596276.00	22	0.11	50.7	27.76	-82.63	1247435.69	451673.94	-22.15	41.46	298	-129.4	0.88	0.13	0.94	0.55	8	4519	0	14	2	13
1207596276.20	22	0.11	50.4	27.76	-82.63	1247442.15	451663.45	-22.19	42.52	304.2	-129.4	1	0.25	1.03	0.25	8	4612	0	13	2	13
1207596276.40	22	0.12	50.4	27.76	-82.63	1247449.79	451653.33	-22.19	43.71	310	-128	1.06	0.25	1.05	0.4	8	4863	0	13	2	13
1207596276.60	22	0.12	51.1	27.76	-82.63	1247458.7	451643.86	-22.2	44.44	316.2	-126.6	1.06	0.19	0.99	0.4	8	4849	1	10	2	13
1207596276.80	22	0.12	52.3	27.76	-82.63	1247468.65	451635.34	-22.21	44.13	322.3	-111.1	1.06	-0.13	1	0.55	8	4456	2	5	2	13
1207596277.00	22	0.12	54	27.76	-82.63	1247479.2	451628.08	-22.22	43.22	329.5	-109.7	1	-0.25	0.97	0.55	8	4612	1	2	2	13
1207596277.20	22	0.12	57.4	27.76	-82.63	1247490.51	451622.55	-22.26	43.1	335.7	-109.7	1.13	-0.06	1.03	0.7	8	4733	1	7	2	13
1207596277.40	22	0.12	61.1	27.76	-82.63	1247502.23	451617.76	-22.25	44.13	339.4	-84.4	0.75	0.38	1.05	0.7	8	4743	0	17	2	13

1207596310.80	22	0.47	18	27.77	-82.63	1250053.49	451652.51	-22.42	48.73	118.1	-113.9	1.06	-0.25	0.99	1.15	5	5177	0	2	2	12.9
1207596311.00	22	0.47	24.2	27.77	-82.63	1250046.17	451664.67	-22.42	48.76	124.2	-133.6	0.94	-0.13	0.85	1.3	5	5062	0	2	2	12.9
1207596311.20	22	0.47	27.4	27.77	-82.63	1250037.34	451676.15	-22.55	47.93	129.9	-130.8	1.06	-0.25	1	1	5	4838	0	2	2	12.9
1207596311.40	22	0.47	29.7	27.77	-82.63	1250028.08	451686.21	-22.57	46.2	135.7	-129.4	1.13	-0.19	1.21	1.15	5	4988	0	3	2	12.9
1207596311.60	22	0.47	30.1	27.77	-82.63	1250017.89	451695.02	-22.55	45.67	142.2	-122.3	1.06	-0.13	1.08	1.15	4	4884	0	3	2	12.9
1207596311.80	22	0.48	28.6	27.77	-82.63	1250006.88	451702.56	-22.53	45.76	148.5	-120.9	1	0	1.04	1.15	4	4955	0	7	2	12.9
1207596312.00	22	0.48	25.6	27.77	-82.63	1249994.99	451708.97	-22.5	46.05	156.7	-118.1	1.06	0.25	1.16	1.15	4	4995	0	13	2	12.9
1207596312.20	22	0.48	21.9	27.77	-82.63	1249982.08	451713.14	-22.4	47.31	162.5	-115.3	1	0.25	1.04	1.15	4	5096	0	14	2	12.9
1207596312.40	22	0.48	17.9	27.77	-82.63	1249968.56	451717.21	-22.35	48.82	165.2	-92.8	0.88	0.31	0.9	1.15	5	5242	0	18	2	12.8
1207596312.60	22	0.48	12.5	27.77	-82.63	1249954.43	451720.34	-22.33	50.07	169.4	-81.6	0.75	0.25	0.91	1.15	5	5599	0	18	2	12.8
1207596312.80	22	0.48	8.1	27.77	-82.63	1249939.73	451722.46	-22.32	51.49	173.4	-67.5	0.88	0.44	0.92	1.3	5	5588	0	18	2	12.8
1207596313.00	22	0.49	4.6	27.77	-82.63	1249924.49	451723.66	-22.33	54.14	177.1	-66.1	0.75	0.31	0.81	1.3	5	5898	0	29	2	12.8
1207596313.20	22	0.49	2	27.77	-82.63	1249908.09	451723.92	-22.43	56.31	180.3	-57.7	0.75	0.56	0.97	1	5	6085	0	33	2	12.8
1207596313.40	22	0.49	0.2	27.77	-82.63	1249891.55	451723.3	-22.43	57.85	183.1	-42.2	0.63	0.63	1.11	1	5	6355	0	42	2	12.8
1207596313.60	22	0.49	-1	27.77	-82.63	1249874.31	451721.93	-22.41	60.53	184.8	-14.1	0.5	0.69	1.01	1.15	4	6578	0	45	2	12.8
1207596313.80	22	0.49	-1.8	27.77	-82.63	1249856.28	451720.16	-22.4	63.46	185.8	-15.5	0.25	0.63	1.01	1.15	4	6864	0	52	2	12.8
1207596314.00	22	0.5	-2.6	27.77	-82.63	1249837.4	451717.95	-22.37	66.36	186.8	-15.5	0.19	0.69	1.05	1.15	4	7295	0	64	2	12.8
1207596314.20	22	0.5	-3.1	27.77	-82.63	1249817.75	451715.35	-22.32	69.08	187.4	1.4	0.25	0.69	1.06	1	4	9003	0	67	2	12.8
1207596314.40	22	0.5	-2.9	27.77	-82.63	1249797.35	451712.48	-22.24	71.71	187.5	53.4	0	0.75	1.03	1	4	10409	0	63	2	12.8
1207596314.60	22	0.5	-2.1	27.77	-82.63	1249776.19	451709.66	-22.16	74.18	187.7	46.4	0.13	0.56	1.02	1	4	10638	0	54	2	12.8
1207596314.80	22	0.5	-0.1	27.77	-82.63	1249754.38	451706.38	-22.06	76.36	188.4	15.5	0.25	0.63	0.93	1	4	8551	0	54	2	12.8
1207596315.00	22	0.51	2.2	27.77	-82.63	1249732.03	451702.85	-22.03	78.66	188.6	5.6	0	0.5	0.79	1.15	4	7485	0	53	3	12.8
1207596315.20	22	0.51	4.6	27.77	-82.63	1249708.92	451699.21	-22.09	80.93	188.4	1.4	-0.06	0.5	0.99	0.85	4	7466	0	51	3	12.8
1207596315.40	22	0.51	6.8	27.77	-82.63	1249685.23	451695.64	-22.07	83.2	188	2.8	-0.19	0.5	1	1	4	7594	0	53	3	12.8
1207596315.60	22	0.51	8.8	27.77	-82.63	1249660.75	451692.18	-22.12	85.74	187.5	5.6	-0.19	0.63	0.88	1	4	8187	0	63	3	12.8
1207596315.80	22	0.52	10.7	27.77	-82.63	1249635.53	451688.8	-22.2	88.31	186.9	7	-0.19	0.63	0.97	1	4	8017	0	60	3	12.8
1207596316.00	22	0.52	12.3	27.77	-82.63	1249609.49	451685.66	-22.28	90.88	186.1	12.7	-0.38	0.69	1.09	1	4	8417	0	78	3	12.8
1207596316.20	22	0.52	13.5	27.77	-82.63	1249582.69	451682.86	-22.3	93.69	185.3	4.2	-0.38	0.63	1	1	4	8517	0	82	3	12.8
1207596316.40	22	0.52	14.6	27.77	-82.63	1249554.94	451680.28	-22.39	96.71	184.8	1.4	-0.25	0.69	0.95	1	4	8736	0	99	3	12.8
1207596316.60	22	0.53	15.3	27.77	-82.63	1249526.34	451677.78	-22.45	99.65	184.6	1.4	-0.13	0.69	1.06	1	4	9085	0	99	3	12.8
1207596316.80	22	0.53	16.3	27.77	-82.63	1249496.85	451675.29	-22.49	104.18	184.2	1.4	-0.13	0.69	1.03	1.15	4	9316	0	99	3	12.8
1207596317.20	22	0.54	18.3	27.77	-82.63	1249435.2	451670.59	-22.55	106.74	183.8	2.8	-0.31	0.56	1	1.15	4	9784	0	99	3	12.8
1207596317.40	22	0.54	18.7	27.77	-82.63	1249403.43	451668.58	-22.59	109.81	183.2	2.8	-0.31	0.38	1	1.15	4	8282	0	99	4	12.8
1207596317.60	22	0.54	19.3	27.77	-82.63	1249371.07	451666.68	-22.62	111.91	182.8	-2.8	-0.19	0.56	1.02	1.15	4	8566	0	99	4	12.8
1207596317.80	22	0.55	19.7	27.77	-82.63	1249338.05	451665.02	-22.64	114.1	182.2	11.3	-0.38	0.5	1.01	1.15	4	8833	0	100	4	12.8
1207596318.00	22	0.55	19.8	27.77	-82.63	1249304.38	451663.81	-22.66	116.34	182.1	12.7	-0.5	0.5	0.96	1.15	4	8720	0	50	4	12.8
1207596318.20	22	0.55	20.3	27.77	-82.63	1249270.04	451662.15	-22.71	116.94	180.7	15.5	-0.69	0.13	1.01	1.15	4	8782	0	32	4	12.8
1207596318.40	22	0.56	19.2	27.77	-82.63	1249235.97	451662.59	-22.72	116.19	177.8	18.3	-1	-0.06	1.09	1.15	5	8690	0	22	4	12.8
1207596318.60	22	0.56	16.7	27.77	-82.63	1249202.11	451664.4	-22.7	115.71	175.3	28.1	-1.25	-0.06	1.02	1.3	5	8761	0	21	4	12.8
1207596318.80	22	0.56	12.6	27.77	-82.63	1249168.47	451667.86	-22.69	115.32	172.3	26.7	-1.5	0	0.98	1.3	5	8710	0	21	4	12.8
1207596319.00	22	0.57	6.6	27.77	-82.63	1249135.2	451673.16	-22.71	115.7	169.6	18.3	-1.5	-0.06	0.84	1.3	5	8655	0	22	4	12.8
1207596319.20	22	0.57	-0.6	27.77	-82.63	1249101.83	451679.77	-22.83	115.56	166.5	35.2	-1.38	0.13	1	1.3	5	8620	0	23	4	12.8
1207596319.40	22	0.58	-2.6	27.77	-82.63	1249069.39	451688.71	-22.86	115.14	162.7	29.5	-1.69	0.06	1.21	1.3	5	8670	0	33	4	12.8
1207596319.60	22	0.58	1.3	27.77	-82.63	1249037.43	451699.63	-22.81	115.83	159.5	28.1	-1.5	0.19	1.13	1.3	5	8710	0	49	4	12.8
1207596319.80	22	0.58	8.5	27.77	-82.63	1249005.79	451712.24	-22.76	116.98	156.7	29.5	-1.31	0.38	1.01	1.3	5	9030	0	62	4	12.8
1207596320.00	22	0.59	14.1	27.77	-82.63	1248974.42	451726.5	-22.72	117.32	155.6	22.5	-1.31	0.38	0.92	1.45	5	8987	0	97	4	12.8
1207596320.20	22	0.59	19.8	27.77	-82.63	1248943.14	451740.43	-22.74	118.89	153.1	22.5	-1.19	0.5	0.97	1.15	5	9174	0	99	4	12.8
1207596320.40	22	0.59	22.5	27.77	-82.63	1248912.21	451757.78	-22.72	122.09	149.6	15.5	-1.06	0.44	0.99	1.3	8	9185	0	99	4	12.8
1207596320.60	22	0.6	24.1	27.77	-82.63	1248881.34	451776.44	-22.75	124.13	147.9	12.7	-0.88	0.5	0.94	1.3	8	9457	0	100	4	12.8

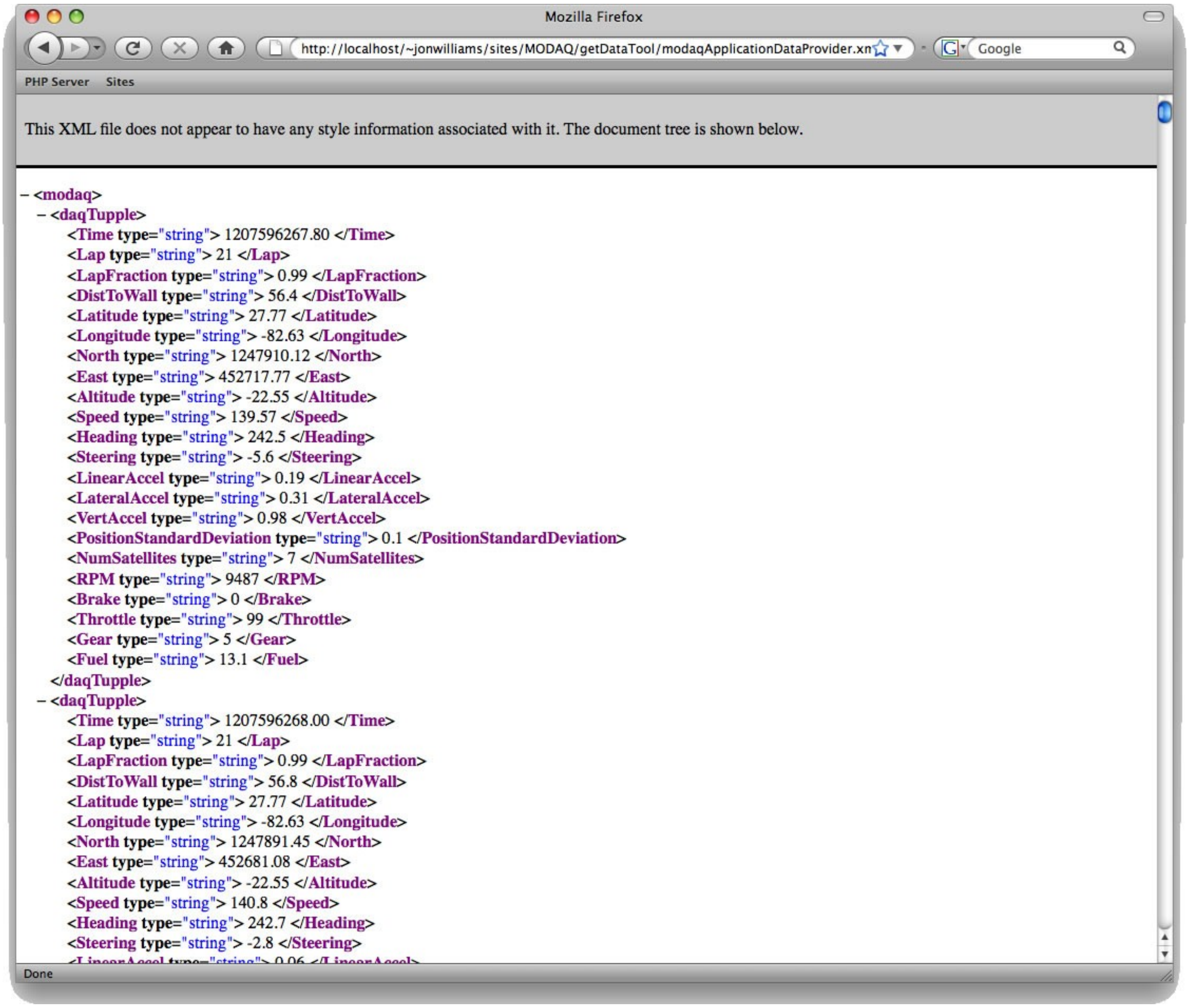
207596861.60	29	0.43	22.5	27.77	-82.63	1249986.1	451359.1	-22.12	45.44	25.5	-84.4	0.94	-0.13	0.97	0.85	4	5122	0	4	2	8.9
207596861.80	29	0.43	29.9	27.77	-82.63	1249997.77	451365.42	-22.17	45.05	30.8	-125.2	0.94	-0.13	0.98	0.85	4	4806	0	4	2	8.9
207596862.00	29	0.43	35.2	27.77	-82.63	1250008.67	451372.76	-22.25	44.37	36.7	-133.6	0.88	-0.19	1	1	4	4737	0	2	2	8.9
207596862.20	29	0.44	39.2	27.77	-82.63	1250018.5	451381.11	-22.32	44.01	42.9	-133.6	1	-0.06	1.07	0.85	4	4755	0	5	2	8.9
207596862.40	29	0.44	40	27.77	-82.63	1250027.44	451390.48	-22.36	44.32	49.2	-133.6	1	0.13	1.15	0.85	4	4840	0	7	2	8.9
207596862.60	29	0.44	38.6	27.77	-82.63	1250035.34	451400.93	-22.34	44.89	55.7	-133.6	1.19	0.13	1.13	1	4	4851	0	7	2	8.9
207596862.80	29	0.44	32.4	27.77	-82.63	1250042.11	451412.38	-22.3	45.64	62	-133.6	1.06	0.19	1.11	1	4	4943	0	9	2	8.9
207596863.00	29	0.44	24.2	27.77	-82.63	1250047.77	451424.68	-22.25	46.54	67.4	-112.5	0.94	0.25	0.97	1	4	5192	0	13	2	8.9
207596863.20	29	0.45	18.6	27.77	-82.63	1250052.46	451437.69	-22.27	47.61	71.9	-91.4	0.88	0.31	0.94	1	4	5252	0	14	2	8.9
207596863.40	29	0.45	14.6	27.77	-82.63	1250056.29	451451.35	-22.28	48.97	75.8	-50.6	0.75	0.38	0.98	1	4	5339	0	21	2	8.9
207596863.60	29	0.45	11.6	27.77	-82.63	1250059.36	451465.65	-22.31	50.85	79.6	-57.7	0.69	0.5	0.94	1	4	5653	0	30	2	8.9
207596863.80	29	0.45	9.4	27.77	-82.63	1250061.53	451480.78	-22.37	52.95	83.3	-53.4	0.69	0.38	0.97	1	4	6342	0	37	2	8.9
207596864.00	29	0.45	8	27.77	-82.63	1250062.82	451496.59	-22.42	55.49	85.4	111.1	0.31	0.63	1.02	1	4	6278	0	12	2	8.9
207596864.20	29	0.45	6.8	27.77	-82.63	1250063.99	451513.32	-22.47	57.06	86.7	32.3	0.56	0.44	1.01	1	4	6053	0	12	2	8.9
207596864.40	29	0.46	6.2	27.77	-82.63	1250064.6	451530.11	-22.51	58	88.5	-22.5	0.38	0.31	1.01	1	4	6247	0	20	2	8.9
207596864.60	29	0.46	6.1	27.77	-82.63	1250064.69	451547.43	-22.55	59.59	90	-22.5	0.25	0.56	0.98	1.15	4	6541	0	33	2	8.9
207596864.80	29	0.46	6.5	27.77	-82.63	1250064.42	451565.15	-22.61	58.79	91.7	-25.3	0.38	-0.31	0.97	1.15	4	5764	9	2	2	8.9
207596865.00	29	0.46	7.5	27.77	-82.63	1250063.51	451581.98	-22.67	55.3	93.9	-39.4	0.5	-0.69	1.13	1.15	4	5651	8	2	2	8.9
207596865.20	29	0.46	9.1	27.77	-82.63	1250062.03	451597.59	-22.63	52.65	98.1	-60.5	0.88	-0.63	1.07	1	4	5611	8	2	2	8.9
207596865.40	29	0.46	12.2	27.77	-82.63	1250059	451612.62	-22.62	50.98	103.7	-80.2	0.94	-0.56	0.96	1	4	5345	7	2	2	8.9
207596865.60	29	0.47	16.4	27.77	-82.63	1250054.84	451626.69	-22.61	48.99	108.9	-87.2	1	-0.44	1.04	1.15	4	5072	4	2	2	8.9
207596865.80	29	0.47	21.9	27.77	-82.63	1250049.59	451639.84	-22.59	47.52	114.4	-98.4	0.94	-0.25	1	1.15	4	5074	1	2	2	8.9
207596866.00	29	0.47	28.5	27.77	-82.63	1250043.22	451652.08	-22.6	46.95	120.2	-112.5	1	-0.19	0.98	1.15	4	4930	1	2	2	8.9
207596866.20	29	0.47	34.5	27.77	-82.63	1250035.63	451663.62	-22.61	46.51	125.9	-115.3	0.88	-0.13	1.01	1	4	4865	0	4	2	8.9
207596866.40	29	0.47	37.2	27.77	-82.63	1250027.14	451674.14	-22.63	45.77	131.8	-122.3	1.06	0.06	1.01	1	5	5070	0	7	2	8.9
207596866.60	29	0.47	39	27.77	-82.63	1250017.67	451683.58	-22.65	44.93	138.1	-122.3	1.06	-0.31	1.05	1	5	4745	1	2	2	8.9
207596866.80	29	0.48	38.3	27.77	-82.63	1250007.48	451691.69	-22.65	44.14	144.5	-119.5	1.06	-0.19	1.06	1	5	4688	0	2	2	8.9
207596867.00	29	0.48	36	27.77	-82.63	1249996.57	451698.55	-22.64	43.54	151	-120.9	1	0.06	1.2	1.15	5	4755	0	10	2	8.9
207596867.20	29	0.48	31.8	27.77	-82.63	1249985.14	451703.99	-22.52	44.36	157.2	-119.5	1.06	0.19	1.07	1	5	4879	0	10	2	8.9
207596867.40	29	0.48	27	27.77	-82.63	1249972.6	451708.53	-22.48	46.11	162.4	-118.1	0.94	0.25	0.9	1	5	5038	0	12	2	8.9
207596867.60	29	0.48	21.7	27.77	-82.63	1249959.38	451712.04	-22.45	47.24	167.1	-98.4	0.75	0.31	0.94	1	5	5199	0	15	2	8.9
207596867.80	29	0.48	16.9	27.77	-82.63	1249945.63	451714.6	-22.45	48.44	171.2	-77.3	0.69	0.38	0.96	1	5	5405	0	16	2	8.9
207596868.00	29	0.49	13.1	27.77	-82.63	1249931.36	451716.26	-22.45	49.95	173.1	-57.7	0.75	0.38	1.03	1	5	5541	0	23	2	8.9
207596868.20	29	0.49	9.1	27.77	-82.63	1249916.6	451717.98	-22.41	51.67	176.3	-50.6	0.63	0.44	0.95	0.85	5	5607	0	25	2	8.9
207596868.40	29	0.49	6.8	27.77	-82.63	1249901.18	451718.06	-22.42	53.91	180.7	-42.2	0.56	0.56	0.89	1	5	5898	0	31	2	8.9
207596868.60	29	0.49	5.1	27.77	-82.63	1249885.06	451717.43	-22.43	56.43	182.9	-25.3	0.5	0.63	0.99	1	5	6255	0	36	2	8.9
207596868.80	29	0.49	3.8	27.77	-82.63	1249868.21	451716.21	-22.43	59.11	184.7	-22.5	0.5	0.69	0.99	1	5	6462	0	42	2	8.9
207596869.00	29	0.49	3	27.77	-82.63	1249850.61	451714.41	-22.43	62.02	186.1	-15.5	0.38	0.63	1.04	1	5	7295	0	55	2	8.9
207596869.20	29	0.5	2.3	27.77	-82.63	1249832.15	451712.18	-22.38	65.03	186.7	-4.2	0.19	0.75	1.05	0.85	5	7032	0	55	2	8.9
207596869.40	29	0.5	1.8	27.77	-82.63	1249812.85	451709.77	-22.33	68.13	186.7	15.5	0.06	0.88	1.01	0.85	5	7466	0	58	2	8.9
207596869.60	29	0.5	1.7	27.77	-82.63	1249792.59	451707.28	-22.26	70.74	186.6	5.6	-0.06	0.75	1.02	1	5	7882	0	59	2	8.9
207596869.80	29	0.5	2.6	27.77	-82.63	1249771.76	451704.76	-22.19	72.98	186.5	-4.2	-0.06	0.38	1	1	5	6906	0	66	2	8.9
207596870.00	29	0.5	3.8	27.77	-82.63	1249750.2	451702.21	-22.12	75.37	186.4	-1.4	0	0.63	0.94	1	5	7731	0	65	3	8.8
207596870.20	29	0.51	5.2	27.77	-82.63	1249727.97	451699.6	-22.09	78.09	186.3	-1.4	0	0.69	0.89	0.85	5	7882	0	65	3	8.8
207596870.40	29	0.51	6.7	27.77	-82.63	1249704.82	451696.91	-22.1	80.92	186.3	0	-0.06	0.63	0.95	0.85	5	7246	0	57	3	8.8
207596870.60	29	0.51	8.1	27.77	-82.63	1249680.94	451694.1	-22.11	83.55	186.4	-5.6	0	0.69	0.98	1	5	7763	0	58	3	8.8
207596870.80	29	0.51	9.6	27.77	-82.63	1249656.27	451691.19	-22.14	86.33	186.4	0	0	0.69	0.9	1	5	8431	0	70	3	8.8
207596871.00	29	0.52	11.1	27.77	-82.63	1249630.78	451688.15	-22.23	88.58	187.2	0	-0.06	0.56	1.02	1	5	8700	0	66	3	8.8
207596871.20	29	0.52	13.3	27.77	-82.63	1249604.9	451684.36	-22.24	91.17	187.1	-5.6	0	0.5	0.99	1	5	9202	0	67	3	8.8

14 - Sample DAQ ".csv" Data File

modaqSampleData.csv

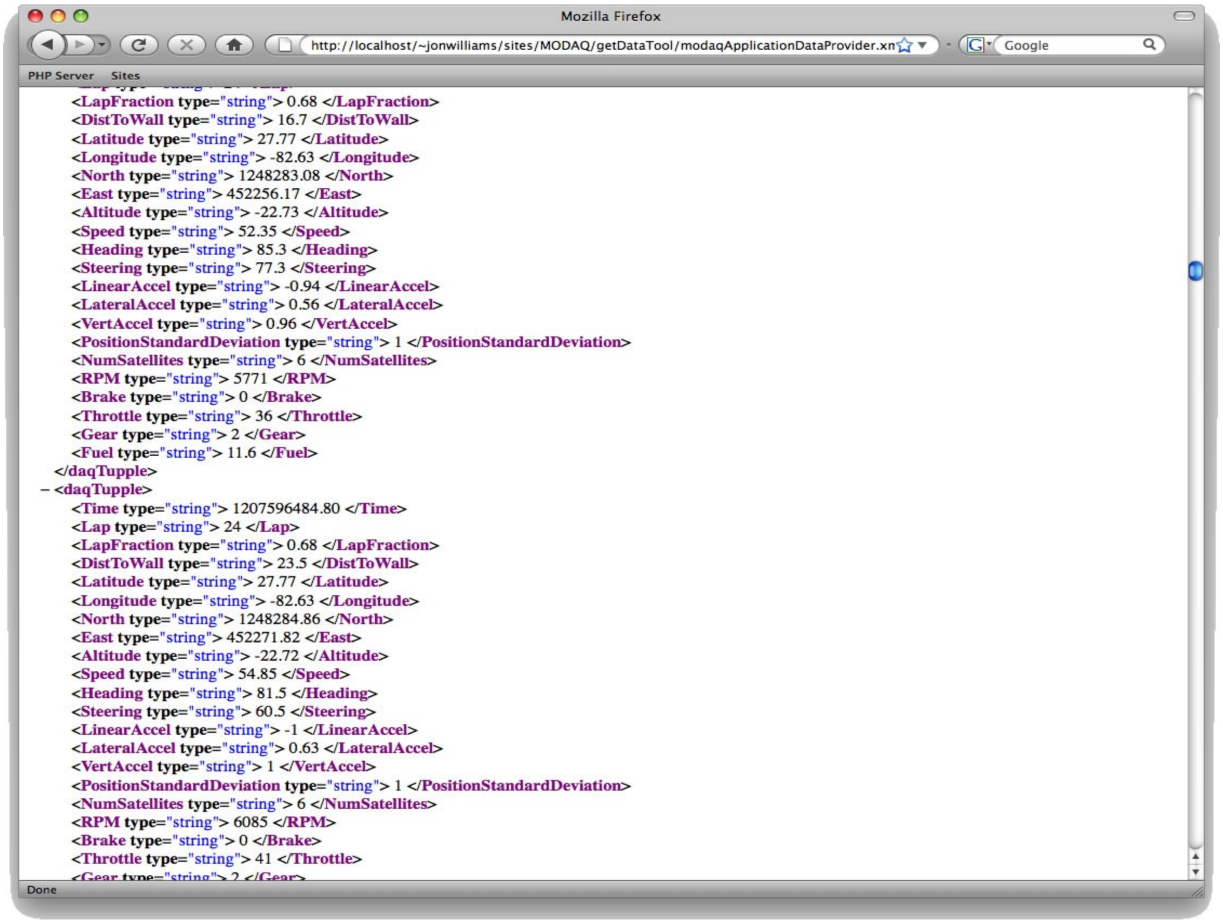
```
"Time","Lap","LapFraction","DistToWall","Latitude","Longitude","North","East","Altitude","Speed","Heading","Steering","LinearAccel","Lateral
Accel","VertAccel","PositionStandardDeviation","NumSatellites","RPM","Brake","Throttle","Gear","Fuel"
1207596267.80,21,0.99,56.4,27.77,-82.63,1247910.12,452717.77,-22.55,139.57,242.5,-5.6,0.19,0.31,0.98,0.1,7,9487,0,99,5,13.1
1207596268.00,21,0.99,56.8,27.77,-82.63,1247891.45,452681.08,-22.55,140.8,242.7,-2.8,0.06,0.31,1.05,0.1,7,9627,0,99,5,13.1
1207596268.20,21,1,56.8,27.77,-82.63,1247872.74,452643.98,-22.52,142.08,243,-5.6,0.06,0.31,1.05,0.1,7,9759,0,100,5,13.1
1207596268.40,22,0,57.4,27.77,-82.63,1247854.1,452606.42,-22.48,143.33,243.5,-8.4,0.19,0.25,0.99,0.1,7,9689,0,100,5,13.1
1207596268.60,22,0,58.7,27.77,-82.63,1247835.71,452568.35,-22.46,144.49,244.1,-8.4,0.38,0.31,1.03,0.1,7,9874,0,99,5,13.1
1207596268.80,22,0.01,60.5,27.77,-82.63,1247817.56,452529.79,-22.4,145.64,244.5,-5.6,0.25,0.31,0.98,0.1,7,9940,0,99,5,13.1
1207596269.00,22,0.01,62.9,27.77,-82.63,1247799.47,452490.83,-22.4,146.71,244.7,-1.4,0.06,0.25,0.98,0.1,7,10046,0,100,5,13.1
1207596269.20,22,0.02,65.4,27.76,-82.63,1247781.24,452451.62,-22.36,147.54,244.5,1.4,-0.13,0.25,1.05,0.1,7,10040,0,99,5,13.1
1207596269.40,22,0.02,67.5,27.76,-82.63,1247762.76,452412.3,-22.32,148.34,244.1,1.4,-0.31,0.19,0.96,0.1,7,10094,0,99,5,13.1
1207596269.60,22,0.03,69,27.76,-82.63,1247743.67,452372.98,-22.31,149.06,243.4,5.6,-0.44,0.19,0.99,0.1,7,10204,0,100,5,13.1
1207596269.80,22,0.03,69.6,27.76,-82.63,1247724.13,452333.71,-22.27,149.78,242.9,1.4,-0.38,0.19,0.97,0.1,7,10197,0,100,5,13.1
1207596270.00,22,0.04,69.3,27.76,-82.63,1247704.15,452294.35,-22.28,150.56,242.5,1.4,-0.25,0.19,0.99,0.1,7,10309,0,99,5,13.1
1207596270.20,22,0.04,68.8,27.76,-82.63,1247683.88,452254.94,-22.24,150.42,242.1,1.4,-0.25,0.19,1.03,0.1,7,10288,0,99,5,13.1
1207596270.40,22,0.05,67.7,27.76,-82.63,1247663.42,452215.94,-22.22,146.33,241.6,1.4,-0.38,-0.88,0.94,0.1,7,9920,11,19,5,13.1
1207596270.60,22,0.05,66.9,27.76,-82.63,1247643.55,452179.02,-22.22,138.69,241.1,-1.4,-0.38,-1.88,0.96,0.1,7,9469,23,3,5,13.1
1207596270.80,22,0.05,66.2,27.76,-82.63,1247624.59,452144.31,-22.23,130.71,240.8,-1.4,-0.25,-1.81,1.03,0.1,7,8870,24,2,5,13.1
1207596271.00,22,0.06,65.3,27.76,-82.63,1247606.57,452111.74,-22.2,122.86,240.5,-2.8,-0.13,-1.69,1.04,0.1,7,8902,23,25,4,13.1
1207596271.20,22,0.06,65.6,27.76,-82.63,1247589.57,452081.21,-22.18,115.15,240.3,-4.2,-0.19,-1.81,0.97,0.1,7,10295,22,13,3,13.1
1207596271.40,22,0.07,65.9,27.76,-82.63,1247573.53,452052.72,-22.17,107.95,240.2,-5.6,-0.13,-1.63,0.97,0.1,7,9505,21,3,3,13.1
1207596271.60,22,0.07,66.3,27.76,-82.63,1247558.49,452025.94,-22.17,101.15,240.4,-5.6,0,-1.56,1.03,0.1,7,9118,17,2,3,13
1207596271.80,22,0.07,66.9,27.76,-82.63,1247544.6,452000.82,-22.14,95.02,240.8,-5.6,0.13,-1.44,1,0.1,7,10080,16,17,3,13
1207596272.00,22,0.07,67.4,27.76,-82.63,1247531.59,451977.04,-22.14,90.14,240.8,-1.4,0.06,-1.06,0.98,0.1,7,9063,9,2,2,13
1207596272.20,22,0.08,67.9,27.76,-82.63,1247519.15,451954.39,-22.13,86.05,240.8,0,-0.06,-0.94,1,0.1,7,8965,8,2,2,13
1207596272.40,22,0.08,69.8,27.76,-82.63,1247507.23,451932.75,-22.13,81.72,240.7,0,-0.13,-0.88,1.03,0.1,7,8645,8,2,2,13
1207596272.60,22,0.08,72.4,27.76,-82.63,1247495.96,451912.36,-22.1,76.49,240.8,-5.6,0,-1.19,1,0.1,7,8038,14,2,2,13
1207596272.80,22,0.09,74.9,27.76,-82.63,1247485.61,451893.36,-22.1,71.32,241,-7,0,-1.25,0.98,0.1,7,7317,13,2,2,13
1207596273.00,22,0.09,77.4,27.76,-82.63,1247475.95,451875.56,-22.09,67.06,241.1,-2.8,0.06,-0.94,1.03,0.1,7,6890,10,2,2,13
1207596273.20,22,0.09,78.6,27.76,-82.63,1247466.83,451858.73,-22.07,62.81,241.6,-5.6,0,-0.94,0.98,0.1,7,6637,9,2,2,13
1207596273.40,22,0.09,79.3,27.76,-82.63,1247458.66,451842.97,-22.08,58.25,243,-22.5,0.25,-1.13,0.99,0.1,7,6075,14,2,2,13
1207596273.60,22,0.1,80.5,27.76,-82.63,1247451.51,451828.13,-22.07,54.69,244.6,-30.9,0.38,-0.81,1.02,0.1,7,5747,8,2,2,13
1207596273.80,22,0.1,79.6,27.76,-82.63,1247445.09,451813.84,-22.07,52.46,245.9,-15.5,0.25,-0.44,1.02,0.1,7,5482,4,2,2,13
1207596274.00,22,0.1,75.1,27.76,-82.63,1247439.13,451799.9,-22.05,50.95,247.5,-29.5,0.25,-0.31,0.98,0.1,7,5368,2,2,2,13
1207596274.20,22,0.1,70.5,27.76,-82.63,1247433.81,451786.1,-22.07,49.39,250.2,-43.6,0.38,-0.38,0.97,0.1,7,5315,2,2,2,13
1207596274.40,22,0.1,66.2,27.76,-82.63,1247429.46,451772.53,-22.08,47.49,253.9,-60.5,0.63,-0.44,1.01,0.1,7,5083,4,2,2,13
1207596274.60,22,0.1,62.6,27.76,-82.63,1247426.23,451759.23,-22.1,45.44,260.4,-81.6,0.75,-0.38,1.09,0.1,7,4926,4,2,2,13
1207596275.00,22,0.11,56.2,27.76,-82.63,1247422.99,451732.97,-21.94,44.29,265,-95.6,0.69,-0.19,0.9,0.25,7,4582,2,3,2,13
1207596275.20,22,0.11,53.8,27.76,-82.63,1247423.05,451720.28,-22.1,42.61,272.9,-105.5,0.88,-0.38,0.93,0.25,7,4623,2,3,2,13
1207596275.40,22,0.11,52.1,27.76,-82.63,1247424.4,451707.96,-22.09,41.45,278.7,-98.4,0.94,-0.25,1.16,0.4,7,4365,3,2,2,13
1207596275.60,22,0.11,51,27.76,-82.63,1247426.83,451696.2,-22.14,40.6,284.9,-112.5,0.94,-0.25,1.01,0.4,7,4378,1,3,2,13
1207596275.80,22,0.11,50.6,27.76,-82.63,1247430.64,451684.92,-22.13,40.77,291.5,-115.3,0.88,0.06,1.05,0.55,8,4377,0,7,2,13
1207596276.00,22,0.11,50.7,27.76,-82.63,1247435.69,451673.94,-22.15,41.46,298,-129.4,0.88,0.13,0.94,0.55,8,4519,0,14,2,13
1207596276.20,22,0.11,50.4,27.76,-82.63,1247442.15,451663.45,-22.19,42.52,304.2,-129.4,1,0.25,1.03,0.25,8,4612,0,13,2,13
1207596276.40,22,0.12,50.4,27.76,-82.63,1247449.79,451653.33,-22.19,43.71,310,-128,1.06,0.25,1.05,0.4,8,4863,0,13,2,13
1207596276.60,22,0.12,51.1,27.76,-82.63,1247458.7,451643.86,-22.2,44.44,316.2,-126.6,1.06,0.19,0.99,0.4,8,4849,1,10,2,13
1207596276.80,22,0.12,52.3,27.76,-82.63,1247468.65,451635.34,-22.21,44.13,322.3,-111.1,1.06,-0.13,1,0.55,8,4456,2,5,2,13
1207596277.00,22,0.12,54,27.76,-82.63,1247479.2,451628.08,-22.22,43.22,329.5,-109.7,1,-0.25,0.97,0.55,8,4612,1,2,2,13
1207596277.20,22,0.12,57.4,27.76,-82.63,1247490.51,451622.55,-22.26,43.1,335.7,-109.7,1.13,-0.06,1.03,0.7,8,4733,1,7,2,13
1207596277.40,22,0.12,61.1,27.76,-82.63,1247502.23,451617.76,-22.25,44.13,339.4,-84.4,0.75,0.38,1.05,0.7,8,4743,0,17,2,13
1207596277.60,22,0.12,65.1,27.76,-82.63,1247514.72,451613.54,-22.26,46.13,343.2,-87.2,0.69,0.44,1.02,0.7,8,4924,0,36,2,13
1207596277.80,22,0.12,67.8,27.76,-82.63,1247528.11,451610.06,-22.24,48.27,346.8,-68.9,0.69,0.63,0.84,0.85,8,5237,0,37,2,13
1207596278.00,22,0.13,67.5,27.76,-82.63,1247542.24,451607.18,-22.38,50.93,350,-57.7,0.63,0.63,0.95,0.1,8,5705,0,42,2,13
1207596278.20,22,0.13,67.1,27.76,-82.63,1247557.49,451605.03,-22.4,53.96,353.9,-60.5,0.63,0.56,1.08,0.1,8,5811,0,42,2,13
1207596278.40,22,0.13,68.2,27.76,-82.63,1247573.64,451603.95,-22.47,56.56,357.8,-74.5,0.94,0.56,1.03,0.1,6,6162,0,42,2,13
1207596278.60,22,0.13,70.9,27.76,-82.63,1247590.56,451603.93,-22.47,59.08,1,-50.6,0.81,0.69,1.08,0.1,6,6451,0,40,2,13
1207596278.80,22,0.13,74.5,27.76,-82.63,1247608.2,451604.73,-22.48,61.28,2.8,-29.5,0.56,0.63,0.94,0.1,6,6634,0,40,2,13
1207596279.00,22,0.13,78,27.76,-82.63,1247626.35,451605.88,-22.53,60.73,2.2,32.3,-0.06,-0.13,1.01,0.1,6,6137,8,2,2,13
1207596279.20,22,0.14,79.5,27.76,-82.63,1247643.69,451606.26,-22.53,57.98,359.4,36.6,-0.56,-0.63,1.04,0.1,6,6043,5,2,2,13
1207596279.40,22,0.14,78.6,27.76,-82.63,1247660.27,451605.7,-22.55,55.54,355.5,50.6,-0.88,-0.56,1.02,0.1,6,5825,6,2,2,13
1207596279.60,22,0.14,72.4,27.76,-82.63,1247676.1,451603.88,-22.53,53.93,351.2,57.7,-1,-0.38,1.02,0.1,6,5671,2,2,2,13
1207596279.80,22,0.14,61.4,27.76,-82.63,1247691.47,451600.98,-22.53,53.32,346.7,63.3,-1,0,0.99,0.1,6,5434,0,5,2,13
1207596280.00,22,0.14,47.4,27.76,-82.63,1247706.49,451596.8,-22.52,53.19,342,67.5,-1.06,0,1.03,0.1,6,5636,0,6,2,13
1207596280.20,22,0.15,31.9,27.76,-82.63,1247721.11,451591.45,-22.5,53.03,338,57.7,-0.94,-0.06,0.95,0.1,6,5601,0,6,2,13
1207596280.40,22,0.15,18.6,27.76,-82.63,1247735.31,451585.27,-22.53,53.28,334.5,49.2,-0.75,0.06,0.95,0.1,5,5753,0,8,2,13
1207596280.60,22,0.15,8.8,27.76,-82.63,1247749.31,451578.1,-22.55,54.15,331.2,52,-0.81,0.19,0.99,0.1,5,5800,0,12,2,13
1207596280.80,22,0.15,2.9,27.76,-82.63,1247763.15,451570.07,-22.59,55.04,327.6,56.3,-0.81,0.31,1,0.1,5,5912,0,13,2,13
```

15 - Generated XML Data Provider Document Extract

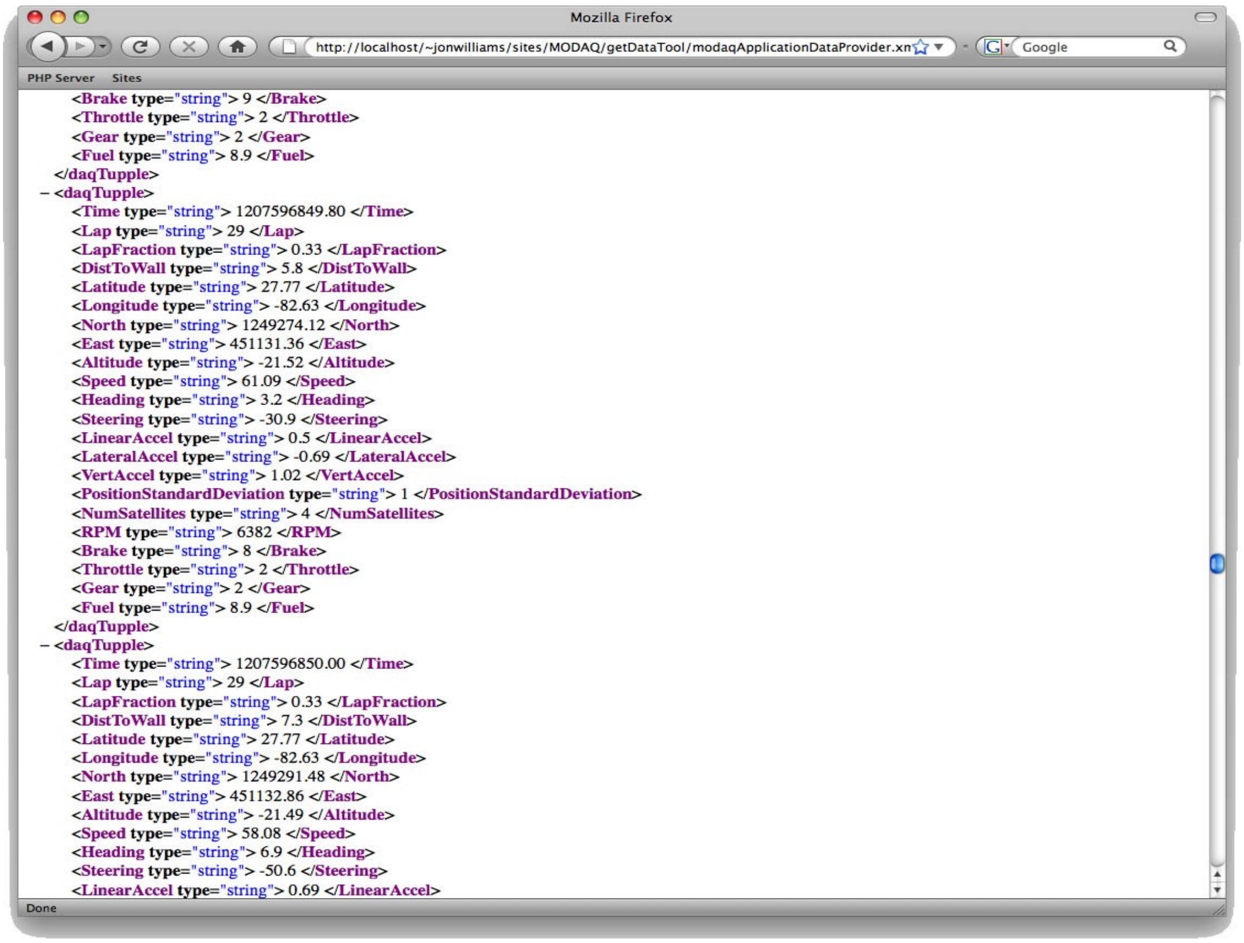


This XML file does not appear to have any style information associated with it. The document tree is shown below.


```xml
- <modaq>
  - <daqTupple>
      <Time type="string"> 1207596267.80 </Time>
      <Lap type="string"> 21 </Lap>
      <LapFraction type="string"> 0.99 </LapFraction>
      <DistToWall type="string"> 56.4 </DistToWall>
      <Latitude type="string"> 27.77 </Latitude>
      <Longitude type="string"> -82.63 </Longitude>
      <North type="string"> 1247910.12 </North>
      <East type="string"> 452717.77 </East>
      <Altitude type="string"> -22.55 </Altitude>
      <Speed type="string"> 139.57 </Speed>
      <Heading type="string"> 242.5 </Heading>
      <Steering type="string"> -5.6 </Steering>
      <LinearAccel type="string"> 0.19 </LinearAccel>
      <LateralAccel type="string"> 0.31 </LateralAccel>
      <VertAccel type="string"> 0.98 </VertAccel>
      <PositionStandardDeviation type="string"> 0.1 </PositionStandardDeviation>
      <NumSatellites type="string"> 7 </NumSatellites>
      <RPM type="string"> 9487 </RPM>
      <Brake type="string"> 0 </Brake>
      <Throttle type="string"> 99 </Throttle>
      <Gear type="string"> 5 </Gear>
      <Fuel type="string"> 13.1 </Fuel>
    </daqTupple>
  - <daqTupple>
      <Time type="string"> 1207596268.00 </Time>
      <Lap type="string"> 21 </Lap>
      <LapFraction type="string"> 0.99 </LapFraction>
      <DistToWall type="string"> 56.8 </DistToWall>
      <Latitude type="string"> 27.77 </Latitude>
      <Longitude type="string"> -82.63 </Longitude>
      <North type="string"> 1247891.45 </North>
      <East type="string"> 452681.08 </East>
      <Altitude type="string"> -22.55 </Altitude>
      <Speed type="string"> 140.8 </Speed>
      <Heading type="string"> 242.7 </Heading>
      <Steering type="string"> -2.8 </Steering>
      <LinearAccel type="string"> 0.06 </LinearAccel>
```



<LapFraction type="string"> 0.68 </LapFraction>
<DistToWall type="string"> 16.7 </DistToWall>
<Latitude type="string"> 27.77 </Latitude>
<Longitude type="string"> -82.63 </Longitude>
<North type="string"> 1248283.08 </North>
<East type="string"> 452256.17 </East>
<Altitude type="string"> -22.73 </Altitude>
<Speed type="string"> 52.35 </Speed>
<Heading type="string"> 85.3 </Heading>
<Steering type="string"> 77.3 </Steering>
<LinearAccel type="string"> -0.94 </LinearAccel>
<LateralAccel type="string"> 0.56 </LateralAccel>
<VertAccel type="string"> 0.96 </VertAccel>
<PositionStandardDeviation type="string"> 1 </PositionStandardDeviation>
<NumSatellites type="string"> 6 </NumSatellites>
<RPM type="string"> 5771 </RPM>
<Brake type="string"> 0 </Brake>
<Throttle type="string"> 36 </Throttle>
<Gear type="string"> 2 </Gear>
<Fuel type="string"> 11.6 </Fuel>
</daqTupple>
– <daqTupple>
<Time type="string"> 1207596484.80 </Time>
<Lap type="string"> 24 </Lap>
<LapFraction type="string"> 0.68 </LapFraction>
<DistToWall type="string"> 23.5 </DistToWall>
<Latitude type="string"> 27.77 </Latitude>
<Longitude type="string"> -82.63 </Longitude>
<North type="string"> 1248284.86 </North>
<East type="string"> 452271.82 </East>
<Altitude type="string"> -22.72 </Altitude>
<Speed type="string"> 54.85 </Speed>
<Heading type="string"> 81.5 </Heading>
<Steering type="string"> 60.5 </Steering>
<LinearAccel type="string"> -1 </LinearAccel>
<LateralAccel type="string"> 0.63 </LateralAccel>
<VertAccel type="string"> 1 </VertAccel>
<PositionStandardDeviation type="string"> 1 </PositionStandardDeviation>
<NumSatellites type="string"> 6 </NumSatellites>
<RPM type="string"> 6085 </RPM>
<Brake type="string"> 0 </Brake>
<Throttle type="string"> 41 </Throttle>
<Gear type="string"> 2 </Gear>



<Brake type="string"> 9 </Brake>
<Throttle type="string"> 2 </Throttle>
<Gear type="string"> 2 </Gear>
<Fuel type="string"> 8.9 </Fuel>
</daqTupple>
– <daqTupple>
<Time type="string"> 1207596849.80 </Time>
<Lap type="string"> 29 </Lap>
<LapFraction type="string"> 0.33 </LapFraction>
<DistToWall type="string"> 5.8 </DistToWall>
<Latitude type="string"> 27.77 </Latitude>
<Longitude type="string"> -82.63 </Longitude>
<North type="string"> 1249274.12 </North>
<East type="string"> 451131.36 </East>
<Altitude type="string"> -21.52 </Altitude>
<Speed type="string"> 61.09 </Speed>
<Heading type="string"> 3.2 </Heading>
<Steering type="string"> -30.9 </Steering>
<LinearAccel type="string"> 0.5 </LinearAccel>
<LateralAccel type="string"> -0.69 </LateralAccel>
<VertAccel type="string"> 1.02 </VertAccel>
<PositionStandardDeviation type="string"> 1 </PositionStandardDeviation>
<NumSatellites type="string"> 4 </NumSatellites>
<RPM type="string"> 6382 </RPM>
<Brake type="string"> 8 </Brake>
<Throttle type="string"> 2 </Throttle>
<Gear type="string"> 2 </Gear>
<Fuel type="string"> 8.9 </Fuel>
</daqTupple>
– <daqTupple>
<Time type="string"> 1207596850.00 </Time>
<Lap type="string"> 29 </Lap>
<LapFraction type="string"> 0.33 </LapFraction>
<DistToWall type="string"> 7.3 </DistToWall>
<Latitude type="string"> 27.77 </Latitude>
<Longitude type="string"> -82.63 </Longitude>
<North type="string"> 1249291.48 </North>
<East type="string"> 451132.86 </East>
<Altitude type="string"> -21.49 </Altitude>
<Speed type="string"> 58.08 </Speed>
<Heading type="string"> 6.9 </Heading>
<Steering type="string"> -50.6 </Steering>
<LinearAccel type="string"> 0.69 </LinearAccel>

16 - Data Provider PHP Script

uploadInsertExport.php (XHTML)

```php
<?php
    $target_path = ""; // set the target path to the folder root: ""

    // set the target path: fileName "uploadFile" + extension ".csv" = "uploadFile.csv" etc...
    $target_path = $target_path.basename( $_FILES['uploadedfile']['name']);

    // check if the file upload was successful - and id so deal with a successful result
    if(move_uploaded_file($_FILES['uploadedfile']['tmp_name'], $target_path)) {
        echo "The file ".basename( $_FILES['uploadedfile']['name'])." has been uploaded<br/><br/>"; // DEBUG TRACE

        //connect to MySQL db - in this case (at this porototpye stage) on my local box for testing
        $conn = mysql_connect("127.0.0.1","root","");

        if (!$conn) { // handle if a connection to the mySQL DB could not be established
            die('Could not connect: ' . mysql_error());
        }
        else {
            $fileName = basename($_FILES['uploadedfile']['name']); // set up the file for execution

            if(file_exists($fileName)) { //handle the file if it exists and can be read
                $handle = fopen($fileName, "r"); // open the file and read its contents

                // read (take) the first line containing the headers
                $columnTitleArr = fgetcsv($handle, 1000, ",");

                $dbColumnInsStr = ""; // used to create a string of column headers and types for table creation
                $colStr = ""; // used to create a string of just column header names

                // create table column name and type string
                for($i=0; $i<count($columnTitleArr); $i++) { // loop through the header array - should have a length of 1
                    if($i != (count($columnTitleArr)-1)) { // test to see if it is not the last element in the array
                        $dbColumnInsStr = $dbColumnInsStr.$columnTitleArr[$i]." varchar(255), "; // increment the header and type string
                        $colStr = $colStr.$columnTitleArr[$i].", "; // increment the header name string
                    }
                    else { // handle the last element of the array differetnly
                        $dbColumnInsStr = $dbColumnInsStr.$columnTitleArr[$i]." varchar(255)"; // increment the header and type string (last element)
                        $colStr = $colStr.$columnTitleArr[$i]; // increment the header name string (last element)
                    }
                }

                mysql_select_db("MODAQ", $conn); // select the mySql MODAQ DB

                $dbName = "MODAQ"; // set the DB name that is to be dealt with
                $tableName = "DAQData"; // set the table name that is to be dealt with

                // check if table exists
                $check = mysql_query("SELECT * FROM ".$dbName.".".$tableName." LIMIT 0,1");

                // if the table exists - drop it
                if($check) {
                    mysql_query("DROP TABLE ".$dbName.".".$tableName); // drop the table (I want to start from fresh)
                }

                // create table
                // construct create table SQL statemnt from the created "$dbColumnInsStr" containing the column names from the ".csv" file
                $createTableSQL = "CREATE TABLE ".$dbName.".".$tableName."(".$dbColumnInsStr.")";
                echo $createTableSQL."<br/><br/><br/><br/>"; // DEBUG TRACE

                mysql_query($createTableSQL, $conn); // execute the create table query

                // insert data from the .csv file into the newly created table
                $count = 1; // set a counter
                while (($data = fgetcsv($handle, 0, ",")) !== FALSE) { // loop through the reaming ".csv" contents (with headers removed)
                    $num = count($data); // get the length of the ".csv" file - (number of rows)
                    $count++; //increment the counter for current row reference

                    $valueStr = ""; // initialise a string to contain the individual ".csv" row values
                    for ($i=0; $i<$num; $i++) { // lop through each rows elements (tuples)
                        if($i == $num-1) { // check if it is the last element (tuple) being dealt with
                            $valueStr = $valueStr."'".$data[$i]."'"; // increment the value string (last element)
                        }
                        else {
                            $valueStr = $valueStr."'".$data[$i]."',";  // increment the value string
                        }
                    }
                    // insert row into DB table
```

7K / 1 sec

```php
                mysql_query($createTableSQL, $conn); // execute the create table query

                // insert data from the .csv file into the newly created table
                $count = 1; // set a counter
                while (($data = fgetcsv($handle, 0, ",")) !== FALSE) { // loop through the reaming ".csv" contents (with headers removed)
                    $num = count($data); // get the length of the ".csv" file - (number of rows)
                    $count++; //increment the counter for current row reference

                    $valueStr = ""; // initialise a string to contain the individual ".csv" row values
                    for ($i=0; $i<$num; $i++) { // lop through each rows elements (tuples)
                        if($i == $num-1) { // check if it is the last element (tuple) being dealt with
                            $valueStr = $valueStr."'".$data[$i]."'"; // increment the value string (last element)
                        }
                        else {
                            $valueStr = $valueStr."'".$data[$i]."',";  // increment the value string
                        }
                    }
                    // insert row into DB table
                    // construct an insert row SQL statemnt from the created "$valueStr" containing the row elements (tuples) from the ".csv" file
                    $insertDataSQL = "INSERT INTO ".$dbName.".".$tableName." (".$colStr.") VALUES (".$valueStr.")";
                    echo $insertDataSQL."<br/><br/>"; // DEBUG TRACE

                    mysql_query($insertDataSQL, $conn); // execute the insert row query
                }

                // lets now create an XML from all the data that has been insterted into the db
                $query = "select * from " . $tableName; // select all the data from the db

                $result = mysql_query($query, $conn) or die("Could not complete database query"); // process the SQL
                $num = mysql_num_rows($result); // get the number of returned values (the number of rows in the DB) for further reference
                echo $num."<br /><br />"; // DEBUG TRACE

                if($num != 0) {
                    $xmlFileName = "modaqApplicationDataProvider.xml"; // set the XML file name

                    // check to see if the file already exists... if so wdelete it and create a new file (saving on file space)
                    if(file_exists($xmlFileName)) {
                        unlink($xmlFileName); // delete the file is it already exists
                    }
                    $xmlFileHandle = fopen($xmlFileName, 'w') or die("can't open file"); //create a new XML file in the local dir named "modaqApplicationDataP

                    // start XML file content
                    $_xml ="<?xml version=\"1.0\" encoding=\"UTF-8\" ?>\r\n"; // set XML file header
                    $_xml .="<modaq>\r\n"; // start XML content

                    $fields = mysql_num_fields($result); // get the number of fields (columns) in the db
                    echo $fields; // DEBUG TRACE

                    $count = 0; // initialise a custom counter for column reference
                    while ($row = mysql_fetch_array($result)) { // loop through db results to get content to add to the XML file
                        $_xml .="\t<daqTupple>\r\n"; // create the first (of potentially many) XML parent nodes

                        for($count=0;$count<$fields;$count++) { // loop through db tuplles and db headings
                            $field = mysql_fetch_field($result,$count);

                            // build XML childeren content
                            $_xml .="\t<$field->name type=\"$field->type\">\r\n"; // create/open XML child node tupple
                            $_xml .=$row["$field->name"]; // add XML child node tupple content
                            $_xml .="\t</$field->name>\r\n"; // close XML child tupple
                        }
                        $_xml .="</daqTupple>"; // close XML parent node
                    }
                    $_xml .="</modaq>"; // close XML content
                }
                fwrite($xmlFileHandle, $_xml); // write the generated XML content to the created XML file "modaqApplicationDataProvider.xml"
                fclose($xmlFileHandle); // close the connection to the XML file for access and menory management

                echo "XML has been written.  <a href=".$xmlFileName.">View the XML.</a>"; // DEBUG TRACE
            }
        }
    }
    else { // handle if the file upload failed
        echo "There was an error uploading the file, please try again!"; // DEBUG TRACE
    }
?>
```

17 - Possible Testing Methods

For the PS a test plan will be created using selected testing method(s) to test the relevant areas and find and fix any problems that may exist.
Each testing method has its own individual strengths and weaknesses. Not one method will fully evaluate (*cover*) systems functionality, therefore the best practice is deemed to use a selection of methods and evaluate the overall results of each.

The testing methods selected for this project will be relevant for use mainly with and towards the "front-end" application. The selected and potential testing methods for used include:

- Unit testing
- System testing
- Validation testing
- Verification testing
- Security testing
- Usability testing
- Integration testing
- User acceptance testing

17.1 - Unit testing

Unit testing is performed by the actual system developer(s), as a good understanding of the system design, technical foundations any functionalities and any code is Required. The testing should start at the beginning of development and continue throughout the development life-cycle.
Unit testing is a structured approach towards system testing, as most testers do not have well defined unit-levels requirements to validate.

17.2 - System testing

System testing tests if a specific system and its design(s) meet all necessary/documented system requirements. For an effective system test, a concrete and testable specification is required. Use cases are used to create a complete system test plan.
The importance of a use case is rated on its operational frequency. The more they are used is assumed the greater potential for system failure therefore classed as critical to test.

17.3 - Validation testing

Validation testing tests that the data inputted into a system is supported and reacted to correctly. Efficient validation testing tests the smallest unit of the system first, working through the integration testing process and making sure that all units and modules are working together correctly.

17.4 - Verification testing

Verification testing reviews an overall system as a whole. Proposed users of specific system should attend the verification testing that is usually held using various meetings, inspections, and time monitoring techniques.
Verification testing tests the different system requirements to make sure that they are usable and functional. The developers of a system are key participants in the verification tests to ensure that the functional and internal specifications are working before any testing commences.

17.5 - Security testing

A test scenario should be created to test a systems security controls (if and where relevant), as users of sensitive systems need to feel secure (and safe) when using them.
A test scenario will insure that a systems security is responding correctly. For the PS application, an example scenario could be to test the application login panel in relation to usernames, passwords and access rights of the application are appropriate and functioning securely and correctly.

17.6 - Usability testing

Usability testing is used to test how users enjoy using a system. It ensures that users can find information easily and efficiently. Performing usability testing will keep users happy and satisfied during system useage therefore increasing system return rates, and thus popularity and profits.
The User's ability to navigate and find there way around an "application" is tested during usability

testing, and sometimes the usability of a proposed application is tested against the usability of a similar/existing application.

17.7 - Integration testing

Integration testing combines different components within a system and tests them as a "complete" system. "... *The units tested during "unit testing" are taken and tested working together as a complete system*".
The focus of Integration testing is on the interaction between the different objects and modules working together. As all the objects and modules are put together errors may occur that need to be corrected.

17.8 - User Acceptance Testing

User acceptance testing takes a full complete (final) system and tests its overall workings. As the user acceptance testing is walked through it will fulfill the requirements, assist the system training, and help in the development of documentation and guidelines for the system.
User acceptance testing treats all components of an system at a user level where the user can voice their opinion and either accept or reject the system as a whole. The system functionality is tested with the developers and users working together combining there expertise in ensuring that the functionality is correct and a system is performing as expected.
User acceptance testing is the last stage of testing before a system is released/deployed. If any faults are found at this level a regression testing phases is needed to correct them.

[95]-(Stottlemyer, 2003)

18 - Back-End System Test Plan

Project Details	
Title:	**M**otorcycle **D**ata **A**cquisition **S**ystem (MODAQ)
Section:	Back-End System
Date:	12th March 2009
Project Purpose	
System Overview:	The system is a DAQ based system that is used to acquire data from motorcycles traveling at high speeds around a racetrack. Different measurements should be acquired from a motorcycle (*both locational and statistical*) and further transmitted from the motorcycle to a CBS where the data is further handles and passed to a "front-end" application for further plotting and rendering.
Testing Overview:	The provided solution is a theoretical solution, therefore existing test methods of a relevant concept will be taken and adapted towards use at a theoretical level (*for example asking relevant questions opposed to performing relevant tasks*). The system will be tested to see if it meets its overall requirements allowing relevant measurements (data) to be acquired from a motorcycle during a racetrack lap, further storing and transmitting this acquired data to a CBS, further testing the approach taken to establish the solution, and what has been achieved in doing so.
Testing Methods	
Unit Testing:	Unit testing the systems elements/components during the investigation process, consisting of the routes taken and choices made at each "individual" stage of achieving the final result (*solution*).
Integration/Function testing:	The system as a complete "theoretical" model will be tested further, consisting of the same questions asked during Unit testing, however asked again as the system solution os a complete and solid object.
Requirements Testing:	Test to see if the system solution meets its set goals and requirements
Reporting Requirements	
Testing Deliverables:	A "theoretical" system solution meeting the specified requirements will be measured from successful unit testing. The integration testing will be detailed providing results on each of the "major" elements within the system stating the expected outcome, and any relevant comments. As the solution is not a "physical" artifact and cannot be tested at a "physical" level, the overall question of how realistic the solution is in comparison to "real-world" usage.

19 – Front-End Application Test Plan

Project Details	
Title:	**M**otorcycle **D**ata **A**cquisition **S**ystem (MODAQ)
Section:	Front-End Application (UI)
Date:	18th March 2009
Project Purpose	
Application Overview:	The application is a Flex based application that can be run in both desktop (AIR) and web (Flex) environments. The application loads data from an external XML document, saving the data as an internal XML object, and further using this object as a DP to drive the application. The purpose of the application is to allow acquired data (*from a motorcycle*) to be analysed using relevant application components.
Testing Overview:	The application will be tested to see if it meets its overall requirements allowing the user to load and analyse data from an external XML document, further testing the applications functionality, compatibility, usability and overall achievements.
Testing Methods	
Unit Testing:	Testing of the applications elements/components and code during the implementation process as the application was being created.
Integration/Function testing:	Full functional testing of the final application when all components (*units*) are working together as a whole
Requirements Testing:	Test to see if the application meets its set goals and requirements
Compatibility Testing:	Test to seen if the application is compatible with different and relevant OS's (*for the desktop runtime environment*) and browsers (*for the web runtime environment*).
Usability Testing:	To perform set (user based) tasks from the documented user class, and document accordingly.
Heuristic Evaluation:	To test and inspect the usability of the applications UI design
Test Environment	
Hardware:	Access to a computer that supports wither a web-based or desktop environment
Software:	Any internet browser or Adobe AIR client
Reporting Requirements	
Testing Deliverables:	A complete application requirements test highlight if all of the application requirements are met, and if not, why not. Each individual until, component and element should be tested within the application. tasked and users form the documents user classification and task modeling process should be tested. Final the application Usability should be tested with real users. Problems and areas identified will be solved, and if not justified why not.