7503080 Design of Measurement Information System

Practice work

Optimization of F1 - PIT STOP TACTICS

Petteri Pulkkinen
Jarno Riistama
Teemu Vesterinen

INDEX

Executive summary

A decision making support system is described in the following document. The system is a measurement information system that produces information to the user instead of pure values or graphs. The target system is a Formula 1, later abbreviated as F1, pit stop timing optimization tool. The tool utilizes information that is fed into it before the race; data from previous races of the season, qualification and preliminary tactics assumptions. In addition to this, the tool is able to learn from the data obtained along the race. This data corresponds best the actual and current situation on the track and reliability of the tool is greatly enhanced when it utilizes all the possible data.

The optimization of the pit stop timings is based on scenario testing instead of actual direct optimization. The predetermined pit stop strategy for both cars of the team is fed into the system followed by the assumed tactics of the other teams. Results of measurements are used to produce new quantities that can be used in the scenario probing. The scenarios can be altered by the team members if they desire to do that. The scenario is ran through and the result is shown on a new screen at the pit lane. A soft screen for the scenario probing results is initialized on this screen.

The goal in optimization is that the car leaving the pit is as high in the place list as possible in those circumstances. The optimization system gives also a suggestion about the new tactics if it notices that the suggested tactics by the team is lousy and could be improved. The team can change the tactics according to the hints given by the system or they can stick to what they have decided in before the race. The penalty function in optimization is time gap between team's car and the leader.

1. Description of the system

The target system is a proposal to solve the problem of right timing of pit stops in a F1-race. The system consists of a huge amount of measurements that are constantly or at regular intervals (time or distance) made during a F1-race. The measurements are made by the team for their own cars and for all the other co-racers cars. The results of these measurements are turned into information of some practical value with the measurement information system presented here. Results of direct measurements such as velocity and fuel left in the tank are post-processed thus creating new results. These new results may represent some totally new quantities that give more information about the system that is observed than the direct measurements. These new quantities are called soft sensors and they are not sensors by physical means but more like software sensors hence the name.

The problem in constructing F1-race strategy is that one can never be sure what the other teams will do and what their tactics is. If there would be an on-line measurement system that would convert the raw data into information the situation would be a lot easier for the teams. The teams have also to be sure that their own cars racing on the track do not come into the pit at the same time or any other way interfere each other.

The optimization goal is simple: After every pit stop the car coming to the pit and eventually leaving it will be on the best possible placement. If team succeeds in this mission, their job is perfectly accomplished. After the last pit stop there is nothing the pit stop strategy optimization system could do for the driver to win the race. It is a question of the car's reliability and the driver's abilities how the final result will look like thereafter.

There are plenty of measurements that are used in gathering data needed in the optimization. The velocity profile of the drivers in the race gives information on how their performance has been developing. From this information the team can estimate the pit stop strategies and timings of other teams. The velocity profile is a 2D-graph on velocity versus place on the track. For other teams only velocities at the intermediates can be measured which means that the velocity profile will be a discrete profile. Only for own cars the team is able to make accurate velocity and place measurements continuously.

Partially for this reason the optimization procedure is performed after every intermediate, i.e. 2-3 times per lap. Due to enormous amount of data the system includes, the optimization will take time for the computers to be ready. This also sets limits to the optimization frequency. This problem is made slightly easier by choosing only the six first cars to participate in the scenario probing. Third reason for the optimization to occur only few times per lap is that the pit stop strategy can be altered only once per lap, i.e. at the beginning of the pit stop drive lane. The team has to determine whether the car is coming in to the pit or not.

Other measurements that are used to support the decision making result are e.g. amount of fuel left in the tank and estimates such as tire condition.

However a good decision making support system we might construct it is not used if the interface related to it is complicated. To emphasize the usability of the interface, only one new monitor per car is needed to the pit. There are at least five different monitors already at the pit lane so one new does not add to the complexity of the system unreasonable much. Information to be shown on the monitor is too much compared to the space that is available on this one monitor. Therefore several sheets are implemented on the other half of the monitor to show desired soft sensors as graphs. These graphs typically include information given by the different soft sensors of the system, i.e. effectiveness of the driver by comparing fuel consumption to the distance driven etc. The user can change between different graphs simply by clicking at the corresponding sheet.

There must, however, be also pure values from the measurements that can be shown to the team personnel. Pure values may give more information about the car functionality to some members of the team than the graphs do. For this purpose the user interface is equipped with a part that shows the measurement results as well as results from the soft sensors as numerical values. These values are loaded from the system database, either from static or dynamic database. Because they are saved in a database, they are also easily accessible for all the other softwares that the team may desire to use.

Actual purpose of the software is to function as a supporting system in decision making. This task is fulfilled so that instead of direct optimization different scenarios are created and ran through to see their effect. In the beginning of the race one feeds in qualification results and preliminary tactics guesses for the other teams. As the race goes on, the system learns by itself from the results it gets from the measurements and as input from the team members.

The scenario testing is based on the previous data: how different drivers have succeeded with their cars in the race and what decisions their teams have made during the season etc. Based on all this knowledge, the team leader or some responsible for the tactics estimation in the team will make an assumption on how the other team will manage in the race and what their tactics will be. During the race, as it is seen in reality what the tactics of the others will be, the system is able to update the tactics automatically. The velocity profiles etc. soft sensors are used to build the estimate for the other drivers' performance in the race. Utilizing this knowledge about the performance of the other drivers and that of own drivers the scenario is run through with the assumption that own car comes into the pit according to some predetermined tactics. Then the result after pit stop according to the predetermined schedule is shown and a possibly better choice of tactics is suggested. As mentioned earlier, the goal of optimization is that the car comes out of the pits as number one. The objective can be penalized as time difference to the leader and the penalty function is minimized. The team can alter their tactics on-line and the system calculates the resulting situation with the proposed tactics.

2. User requirements, UR

This section contains the User Requirements for the tool.

User Requirements are classified to six groups:

- Optimization
- User interfaces
- Stochastic elements
- Communication
- Performance

2.1. UR-1 Optimization

UR-1.1	Optimize the number of the pit stops
Goal	Support the Team in selecting the optimal number of pit stops.
Description	The tool will support the team to select the best possible strategy for the race. The tool makes a suggestion for the best possible number of pit stops based on measurements, statistics and common knowledge provided by the experts of the team.
Output	Optimized number of the pit stops

UR-1.2	Optimize the timing of the pit stops
Goal	Support the Team in selecting the optimal timing for the pit stops.
Description	As the optimal number of pit stops is selected, the tool calculates the best possible time steps for the pit stops. Again the decision is based on measurements, statistics and common knowledge provided by the experts of the team.
Output	Optimized time steps for the pit stops

UR-1.3	Runtime optimization
Goal	Reoptimize the timing of the pit stops
Description	The timing of the pit stops is reoptimised during the race. The decision is based on previous decisions and the information gathered during the race.
Output	Reoptimized time steps for the pit stops

2.2. UR-2 User Interfaces

The user interface is important because it strongly affects how much the tool is utilized. If the user interface is difficult to use and it is poorly constructed, the tool is not used. The user interface should be intuitive and easy to use. The pit is loaded with monitors providing fragmented information. The idea of this tool is to manage all the needed information via one screen. The decision has to be made instantly. One screen with easy and fast functions provides the keys for the decision making.

UR-2	Only one new display
Goal	To present all the information on one screen
Description	All the information needed for the decision making is presented in one display.
Output	Stand alone system on one screen

2.2.1. UR-2.1 Configuration Interface

UR-2.1.1	Initialization of the optimization variables
Goal	To prepare the platform for the optimization
Description	Before the race begins, the tool is given all the a priori information needed by the optimization. The optimization variables are initialized based on: qualification results current state of the car current state of the track current weather conditions initial race strategy etc
Output	Initial values for the decision variables

UR-2.1.2	Defining the optimization constraints
Goal	To prepare the platform for the optimization
Description	Before the race begins, the tool is given all the a priori information needed by the optimization. The optimization constraints are identified based on: properties of the car properties of the track current weather conditions etc
Output	Values for the constraints

UR-2.1.3	Qualification results and statistics
Goal	To initialize the runtime graphics and figures
Description	The qualification results and statistics form the basis of the decision making.
Output	Initial figures and graphics for the runtime interface

2.2.2. UR-2.2 Runtime Interface

UR-2.2.1	Positions of the drivers
Goal	To visualize the standings.
Description	The position of each driver is clearly presented in a graphic screen. There is no need to follow multiple screens to get an overall view.
Output	Graphical presentation of the standings

UR-2.2.2	Race statistics
Goal	To present the on line statistics
Description	Clear representative figures of the on line race statistics: lap times intermediates gaps pit stop durations etc
Output	On line statistics on screen

UR-2.2.3	State of the car
Goal	To present the status of the car in figures
Description	Key figures of the status of the car: remaining fuel condition of the tires speed of the car etc
Output	Key figures of the state of the car on screen

UR-2.2.4	User's inputs
Goal	To give the user an access to the tool
Description	The user is able to modify the optimization variables during the race.
Output	Manual runtime inputs

2.3. *UR-3 Stochastic elements*

Uncertainties are always present, when talking about formula – 1 cars. Engineers have developed cars to maximize the speed. To reach that goal they have done solutions, which cover only certain conditions. Stochastic elements are the part where different conditions are playing very big role. Also external disturbances can cause trouble.

UR-3.1	Possibility of a crash
Goal	Minimize pit stop time, when break is unexpected
Description	During the race many different things can go wrong. In most races the first lap and especially the first corner causes loss of hair to the manager. Every racer tries to get in the first place in the first corner.

	Sometimes accidents are inevitable, even after the first lap. After the crash this part tells to the mechanics what part has broken. By telling the broken part before car comes in, mechanics can prepare with right parts. This action minimizes the time used in pit stop. This part is not an optional and a hint for manager. This section tells if car must be taken in. Other way to say it is that these are panic alarms.
Output	Broken part alarm

UR-3.2	Possibility of rain
Goal	To optimize the choice of right tires
Description	During the race weather conditions sometimes changes a lot. One part of the track can be rainy when other parts are sunny. It is possible to predict the weather so that we only change the tires when taking car normally in. Sometimes rain starts unexpected and mixes up strategy. If it starts to rain enough the risk is higher than the need of filling tank, car must be taken in. When the risk is tolerable last call comes from driver himself.
Output	Tyre choice and risk factor

UR-3.3	Possibility of the tires breakdown
Goal	Optimize the right moment of changing tires
Description	During the race every driver drives different way. This makes it harder to predict the right moment to change tires. It is possible to get on-line information from car, which tells how the driver has handled his car. If the rotation of the axel is faster than the expected speed, then the risk is higher to break tires.
Output	Probability of tires breakdown and lap number

UR-3.4	Difficulties on track
Goal	To optimize pit stop timing in case of safety car
Description	There can be accidents during the race. This often causes safety car to come in. Also flashers cause this… When safety car comes to track, overtaking is not allowed. Pit stops are though allowed. If safety car comes in too many times it is possible that the race exceeds its limits (2h time or 300 km). This must be taken under consideration.
Output	Driver in alarm

UR-3.5	Difficulties in pit
Goal	To give information if something goes wrong in pit stop
Description	In pit stop everyone are trying to act as fast as possible. This can cause trouble. Fuel pump can jam or hose connector does not work. Tire nut may jam. This also tells the manager if someone should be replaced from the team… This is given to use in optimization of future stops
Output	Need for extra pit stop

2.4. UR-4 Communication

In order to function as an on-line optimization tool, all parts of the system must be able to communicate with the rest of the tool. Lap times, amount of fuel left, weather conditions etc. factors are constantly observed and recorded and they altogether or separately may have a great impact on the performance of the car. Hence, all of these factors have to be taken into account in the decision making process. The whole optimization of the pit stop timings will be left out of basis if the car runs out of fuel, for example. Due to this, the system has to have access to the database and it has to be able to write and read to and from that database.

There is not only one but two or even three cars simultaneously on the track that all share the same pit stop area. With the rules and systems of today, it is impossible to overhaul more than one race car at a time on the pit. Therefore it is also of huge importance that there exists no double booking on the pit but every car comes in to the pit on the time slot determined by the team. The tool will have to be equipped with a system that links the position data and the pit stop strategies of both cars together.

UR-4.1	Data exchange with the existing system
Goal	To read and write data from/to the database.
Description	For the tool to function as an on-line tool with optimization capability based on the current situation it has to be in contact with the database of the existing system. It has also be able to write new data to the database, e.g. Lap times Velocity profile Place etc.
Output	Data written or read to/from the database

UR-4.2	Data exchange with the other car's system
Goal	To read and write data from/to the database.
Description	For the tool to function as an on-line tool with optimization capability based on the current situation it has to be in contact with the database of the existing system. The database must contain the information where is the other car of the team and it's future plans and tactics.
Output	Data written or read to/from the database

2.5. UR-5 Performance

The tool must provide on line information yet this is by no means an easy task to do. The huge amount of data that is stored into the system is heavy to handle and not even all the traced variables are continuous in measurement sense. The velocity profiles, for example, can be obtained only on the two or three intermediate points on the track. Lap times are also one such a quantity. The problem can be solved so that one defines that there is no possibility to obtain new optimization result based on this information more than twice or three times during the lap. This gives the tool time to refresh its suggestion of the strategy on which lap to come into the pit. Other data apart from the optimization result and the discrete measurement values should anyway be on-line information, e.g. amount of fuel etc.

UR-5.1	Optimization schedule
Goal	Optimization results are updated after each intermediate
Description	One important factor affecting the optimization result is the lap time which cannot be obtained continuously. This limits the repetition interval of the optimization that still makes sense. Hence the optimization task will be performed after each of the intermediates on the track (2-3 times per lap). This pace also leaves the team time to react on the possibly changed strategy.
Output	Optimized pit stop strategy for car in question 2-3 times per lap.

UR-5.2	Rapid data transfer between different parts of the tool
Goal	Transfer collected data rapidly between different functionalities.
Description	Some quantities can be continuously measured and recorded, amount of fuel for example, and it can be of great importance in some parts of the system that this data really is on-line data.
Output	Rapid data access for all parts of the system to the database of all the other parts.

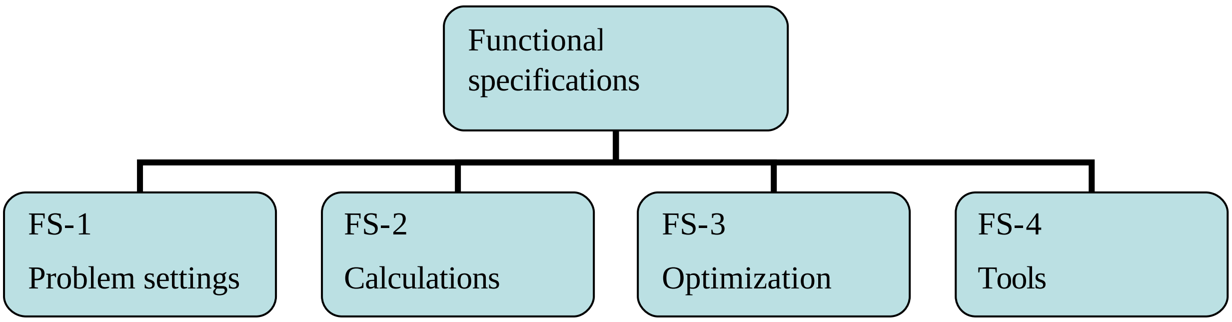
3. Functional Specifications

The user requirements have been analyzed to produce a set of functional specifications. Functional Specifications are the functions, which are needed to fulfill the User Requirements.

The Functional Specification block-diagrams are presented in the next sections. Functionalities are hierarchically organized from the topmost high-level functions to the lower level functionalities that actually will have to be realized by software modules. Diagrams are provided to clarify this hierarchical structure. The lower level functions will be described together with the specifications of the input and the outputs of the function.

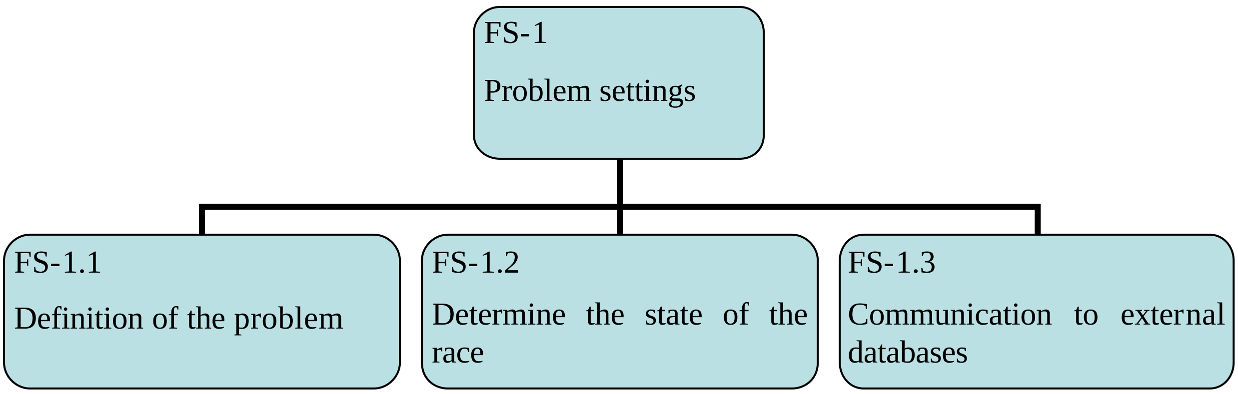
The functionalities have been classified into four main functions:

- Problem settings
- Calculations
- Optimization
- Tools



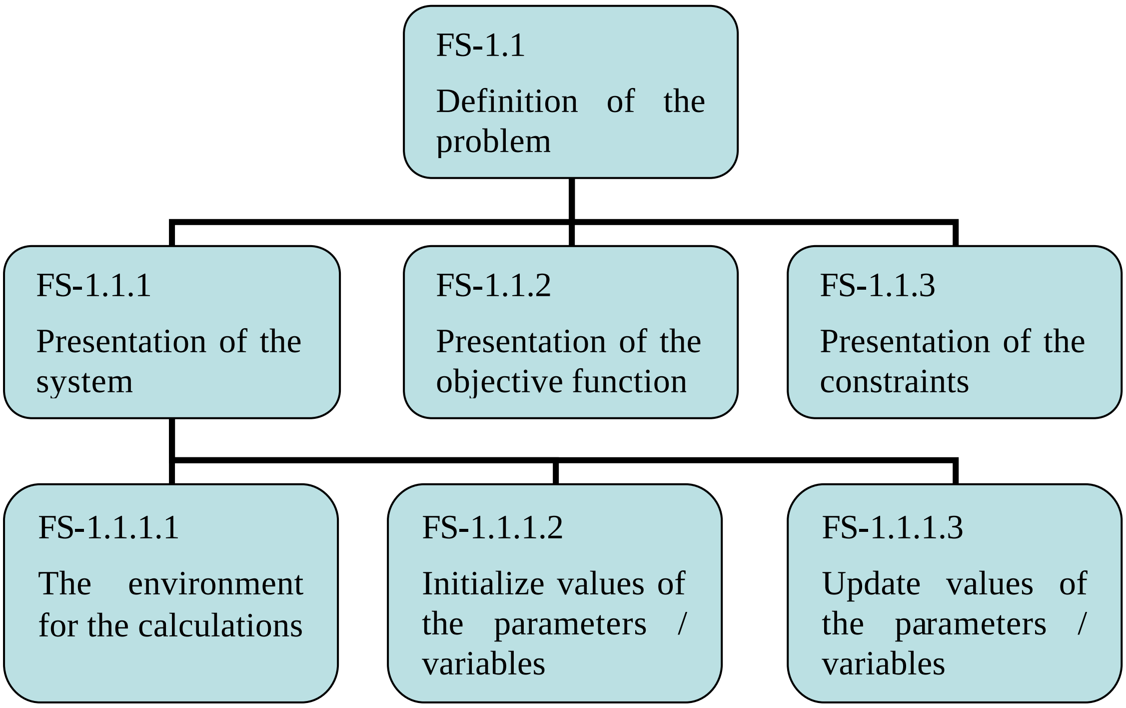
3.1. *FS-1 Problem settings*

These functions form the basis for the whole optimization. The optimization problem is defined and initialized and the rules for the communication are created.



3.1.1. FS-1.1 Definition of the problem

The first of the functional specifications are to set up the optimization problem. The set up procedure includes naming of parameters and variables, initializing them, updating them, etc.



FS-1.1.1.1	The environment for the calculations
Description	To do the calculations and get the resulting data, the variables and the parameters must be known.
Input	Void
Output	Names of the variables and parameters
Supported UR:s	UR-1, UR-2

FS-1.1.1.2	Initialize values of the parameters/variables
Description	The initialvalues of the optimization parameters and variables are stored to the data base before the race.
Input	Parameter and variable names
Output	Initial values of the parameters and variables for the calculations.
Supported UR:s	UR-1, UR-2

FS-1.1.1.2.1	Initialization of the results of the qualifying
Description	Results from the qualifying are fed into the static database. The results are needed in optimization and in determining opponent's pit stop strategy.
Input	Qualifying results for all cars in format m:ss:sss and placement.
Output	Static database including the qualifying results.
Supported UR:s	UR-2.1.1, UR-2.1.3

FS-1.1.1.2.2	Initialization of the car state
Description	State of the car is fed into the static database. By state is meant the variables of the car before the race: weight, amount of gasoline, tyres, etc. From these the maximum number of laps before first pit stop is calculated.
Input	Car weight, amount of gasoline, tyres.
Output	Static database including the initial conditions of the car.
Supported UR:s	UR-2.1.1

FS-1.1.1.2.3	Initialization of the track conditions
Description	Conditions of the track are initialized. The most important quantity of the track is temperature.
Input	Temperature of the track in the beginning.
Output	Static database including the temperature of the track.
Supported UR:s	UR-2.1.1

FS-1.1.1.2.4	Initialization of weather conditions
Description	The current weather conditions are stored into the static database to be used in the optimization process.
Input	The weather conditions just prior to the start.
Output	Static database with the weather conditions.
Supported UR:s	UR-2.1.1 and UR-2.1.2

FS-1.1.1.2.5	Initial pit stop strategies
Description	The initial pit stop strategies are first determined and then defined into the static database. Initial guesses are to be used in the optimization of the pit stop strategy.
Input	Initial pit stop strategies of own car's and opponent's cars. Strategy includes lap numbers of the predetermined stops.
Output	Static database including initial pit stop strategies.
Supported UR:s	UR-2.1.1

FS-1.1.1.2.6	Results of the previous races of the season
Description	Results from the previous races are defined into the database to be used in the risk analysis, e.g. possibility of the engine breakdown etc. distributions are defined with the help of previous data.
Input	Results and problems of the previous races.
Output	Static database including the statistics of the previous races and distributions determined from those.
Supported UR:s	UR-2.1.1

FS-1.1.1.3	Update values of the parameters/variables
Description	This function collects the specific data from the data base and updates the parameters and variables needed by the optimization.
Input	Parameter and variable names
Output	Updated values of the parameters and variables for the calculations
Supported UR:s	UR-1, UR-2

FS-1.1.2	Presentation of the objective function
Description	The objective function has to be formulated first mathematically, to make a function call which is able to evaluate its value.
Input	Mathematical formula of the objective function
Output	Creation of the function call which evaluates the value of the objective function
Supported UR:s	UR-1

FS-1.1.3	Presentation of the constraints
Description	The constraints have to be formulated first mathematically, to make function calls, which are able to evaluate their value.
Input	Mathematical formulas of the constraints
Output	Creation of the function call, which evaluates the values of the constraints.
Supported UR:s	UR-2

FS-1.1.3.1	Initialization of properties of the car
Description	Power of the car, acceleration etc. properties that affect directly performance of the car.
Input	Properties that have an effect on the performance of the car.
Output	Static database with initialized constraints created by the performance limits of the car.
Supported UR:s	UR-2.1.2

FS-1.1.3.2	Initialization of properties of the current track
Description	Properties of the track are fed into the system. Only properties that have some constraining effect on the driving are given. These are e.g. curves and slipperiness of the track.
Input	Properties of the track that have an effect on driving.
Output	Static database with initialized constraints created by the conditions on the track.
Supported UR:s	UR-2.1.2

FS-1.1.3.3	Constraints from current weather conditions
Description	Weather can slow down the speeds a lot and bad weather conditions in general makes it more difficult to race on the track. Constraints of the car and track can be tightened with this additional feature.
Input	Current weather conditions.
Output	Dynamic database initialized with the current weather conditions.
Supported UR:s	UR-2.1.2

3.1.2. FS-1.2 Determine the state of the race

The state of the race depends on the race statistics (standings, gaps, speeds), estimations (pit stop strategies, weather condition) etc. The state of the race is initialized before the beginning of the race and is updated during the race. The state of the race is an indication for the team of the overall situation. The indication of the state is used to select the objective for the optimization. The objective can be

- to maintain the position
- to improve the position by one
- to avoid traffic jam
- to benefit from the rain
- etc

FS-1.2	Determine the state of the race
Description	The state of the race is an indication for the team of the overall situation. The indication of the state is used to select the objective for the optimization.
Input	Race statistics, history data
Output	Indication of the state of the race
Supported UR:s	UR-1

3.1.3. FS 1.3 Communication to external databases

Communication of the measurement information system with the existing data is essential to its functionality. Databases, static and dynamic, must be editable when reason to it arises. Lap times of the race participants are one of the most important quantities in the optimization and decision making process and they have to be updated into the dynamic database.

Communication with the measurement information system of the team’s other car(s) is important when it comes to the pit stop timing and race strategy. Information systems share a great deal of information with each other but optimization and decision making are separate functions. Result of e.g. optimization of the other car(s) depends on the result of the other(s).

FS-1.3.1	Lap time updating
Description	Lap times given by the supporting measurement system are fed into the measurement information system. Lap times include both times from team's own cars and those of the others.
Input	Lap time from the measurement system.
Output	Dynamic database including lap times.
Supported UR:s	UR-4.1, UR-5.2

FS-1.3.2	Velocity profile updating
Description	Velocity profile of the cars of the team is easy to implement with the aid of continuous velocity and place measurement. The profile is somewhat more inaccurate for the other teams since velocity can be obtained only from the intermediate measurement places.
Input	Lap time from the measurement system.
Output	Dynamic database including lap times.
Supported UR:s	UR-4.1, UR-5.2

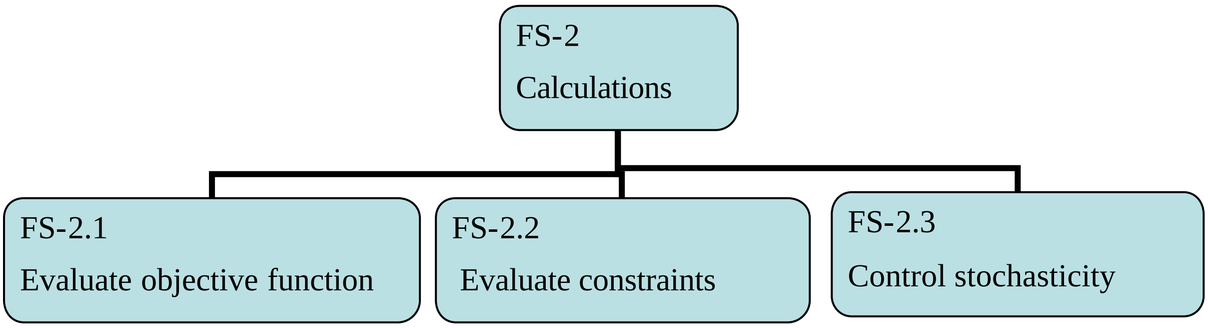
FS-1.3.3	Place on the track updating
Description	Place of the car on the track is important for the dynamic system to be able to optimize the pit stop timing. Velocity profile can be drawn on the basis of velocity and place measurements when exact time is known.
Input	Place at specific time from the measurement system.
Output	Dynamic database including place information.
Supported UR:s	UR-4.1, UR-5.2

FS-1.3.4	Placing of the other car of the team
Description	The race strategy can be adjusted along the race when the placing of the other car is known.
Input	Placing of the other car of the team.
Output	Dynamic database including placing of the other car of the team.
Supported UR:s	UR-4.2, UR-5.2

FS-1.3.5	Pit stop strategy of the other car of the team
Description	The possibility of coincident pit stops for the two cars is avoided by feeding information of both cars into the system.
Input	Pit stop strategy of the other car of the team.
Output	Dynamic database including the pit stop strategy of the other car of the team.
Supported UR:s	UR-4.2, UR-5.2

3.2. FS-2 Calculations

The optimization is based on the calculations done by these functions.

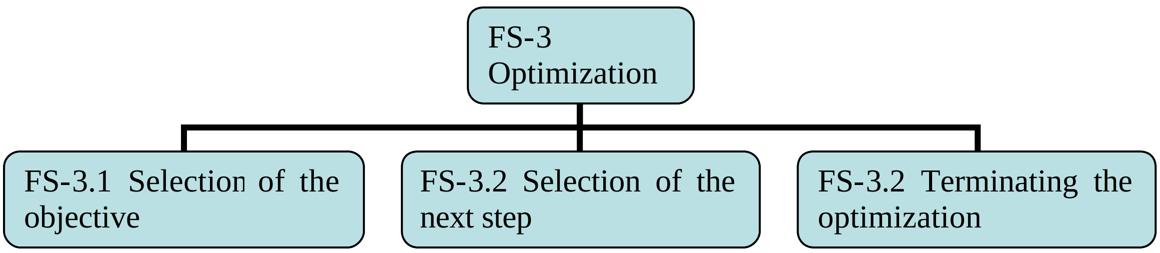


FS-2.1	Evaluate objective function
Description	The objective function must be evaluated during the optimization. In this case there are multiple objectives depending on the present race scenario and tactics.
Input	Names of variables
Output	Value of the objective function
Supported UR:s	UR-1

FS-2.2	Evaluate constraints
Description	In the optimization, there are constraints on some variables. Constraints must be evaluated during the optimization.
Input	Names of variables
Output	Values of constraints
Supported UR:s	UR-1

FS-2.3	Control stochasticity
Description	Stochastic elements have a strong impact on the race. History and on-line data is used to calculate indicators for the possibility of rain, tires breakdown, crash, etc.
Input	Names of stochastic elements
Output	Indicators
Supported UR:s	UR-3

3.3. FS-3 Optimization



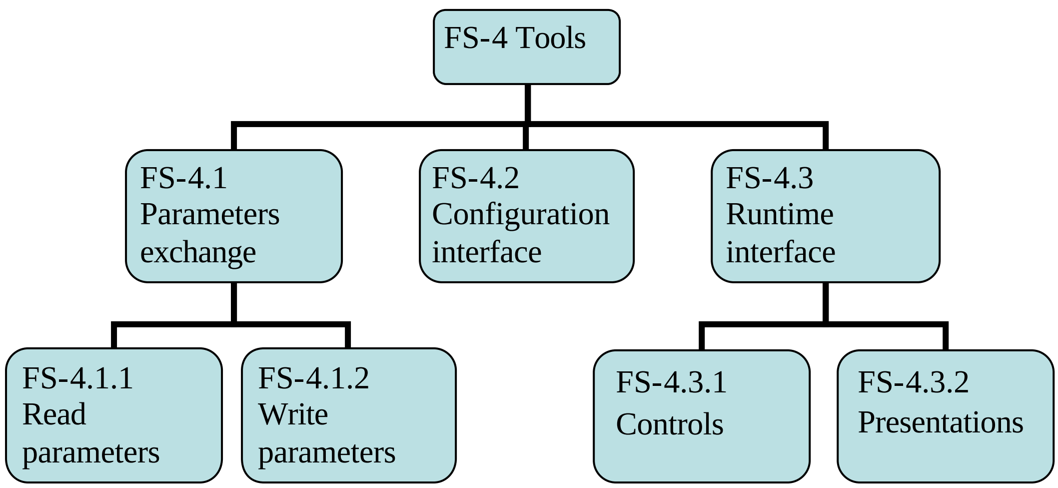
FS-3.1	Selection of the objective
Description	The optimization aims to benefit from the pit stop. The optimal timing depends on many aspects, so there are multiple objectives. The first step in the optimization is to choose the appropriate objective. As an example the objective could be to improve the position by one or to avoid traffic jam.
Input	Results from the calculations describing different scenarios
Output	The name of the selected objective function
Supported UR:s	UR-1

FS-3.2	Selection of the next step
Description	To select the next scenario to be tested by the optimization
Input	Results from the previous optimization step
Output	A new scenario
Supported UR:s	UR-1

FS-3.3	Terminating the optimization
Description	As the optimization is done on-line, it must be terminated rather fast. The aim is therefore not to find the global optimum, but to find a good feasible solution fast.
Input	History of objective function values
Output	Indication of termination
Supported UR:s	UR-1

3.4. FS-4 Tools

User interface that is shown to team personnel is essential in consideration how much the information system is being used. The interface includes different soft screens that show all the quantities and graphs that are important for the team.



3.4.1. FS-4.1 Parameters exchange

FS-4.1.1	Read parameters
Description	Parameters needed in the calculations are saved in the data base and can be read from the data base with parameter names.
Input	List of parameter names
Output	Values of the certain parameters
Supported UR:s	UR-4.1

FS-4.1.2	Write parameters
Description	Parameters are saved to the data base.
Input	Pairs of parameter names and their values
Output	Information if writing succeed or not
Supported UR:s	UR-4.1

3.4.2. FS-4.2 Configuration interface

FS-4.2	Configuration interface
Description	Configuration interface allows manual access to the database
Input	User defined parameters, variables, history data
Output	Values are stored to the database
Supported UR:s	

FS-4.2.1	User interface initialization
Description	To initialize the user interface that is shown on the one new screen needed for this application. The software is loaded from the computer.
Input	Void
Output	The screen initialized with the graphics that are available in the system.
Supported UR:s	UR-2

3.4.3. FS-4.3 Runtime interface

FS-4.3.1	Controls
Description	User can modify parameters and variables during the race. User can also choose the statistics and the graphs that are presented in the display.
Input	User’s request
Output	User’s request is fulfilled
URs supported	UR-2.2

FS-4.3.2.1	Race time
Description	Function gives and draws race time on the top center of the interface
Input	Time
Output	Time
URs supported	UR-2.2.2

FS-4.3.2.2	Section and lap times
Description	Function calculates section and lap times
Input	FS- 4.3.2.1, measurement points information
Output	Section time from race start in table format. One row for each driver
URs supported	UR-2.2.2

FS-4.3.2.3	Presentation of drivers positions
Description	Function to give graphical view about each drivers position based on given position and driver information.
Input	Position, drivers.
Output	Graphical curves about positions in vector formats
URs supported	UR-2.2.1

FS-4.3.2.4	Presentation of drivers sector and lap times
Description	Function to present live information about different drivers sector and lap times in table format. Real time, difference to previous lap or best lap tables can be chosen.
Input	FS-4.3.2.2 sector and lap times in vector format
Output	Numerical presentation about chosen values for each driver
URs supported	UR-2.2.2

FS-4.3.2.5	Driver gaps
Description	Function uses total times to calculate differences between chosen drivers. Presentation in table format
Input	Drivers, total time from each sector
Output	Difference from chosen driver to other drivers
URs supported	UR-2.2.2

FS-4.3.2.6	Speed of car
Description	Function gives car speeds on every measurement point. Teams own cars speeds are updated constantly. Presentation in table format.
Input	Speed, drivers
Output	Speed of driver at different parts of track
URs supported	UR-2.2.2

FS-4.3.2.7	Positions of drivers pit stop durations
Description	Gives time used in pit stop and total time loss caused by pit stop. This is calculated using previous laps last sector time and next laps firs sector time. Difference is then compared to last race lap times. Presentation in table format
Input	Pit stop time, sector times
Output	Pit stop time and total time loss caused by stop
URs supported	UR-2.2.2

FS-4.3.2.8	Car fuel amount
Description	Function gives amount of fuel in tank. Presentation for teams each car in table format
Input	Measurement about fuel
Output	Amount of fuel
URs supported	UR-2.2.3

FS-4.3.2.9	Tire conditions
Description	Function gives estimate about tire condition for each team driver. Estimate is calculated using speed profile on tracks every lap, tachometer value from axels and driven laps with these tires. If tachometer gives high vales against speed tires get worse. If speed on curves is high G-forces are high and tires get worse.
Input	Speed profile on driven tracks, amount of driven laps, tachometer from axels
Output	Tire condition estimate
URs supported	UR-2.2.3

FS-4.3.2.10	Speed profile
Description	Speed versus location on the track
Input	Sector times and positions on track
Output	Present speed profile and estimate for future laps. Each lap consists of speed vectors
URs supported	UR-2.2.2

4. Measurements

4.1. Availability

There are more measurements that can be done in a F1-car than are needed to operate the tool. From the own cars the team has every possible quantity tracked thus they obtain enough information from their cars. The opponent teams are the problem. Only some quantities can be obtained from their cars. These quantities are lap and intermediate times and velocity at the intermediates. Additionally one obtains time that one pit stop approximately takes for every team and how much fuel is loaded into the car when the volume flow is known. From these one can deduce the fuel consumption or fuel left in their cars.

4.2. Uncertainty

The measurements used in the application can be assumed to be very accurate since it is a question of F1-race where time gaps are very marginal. Lap and intermediate times are shown to 1/1000:th of second and the error can be estimated to be less than 0.0005 seconds. Velocity measurements are done probably with a high accuracy radar where measurement errors are in the range of 0.1 km/h. Tire pressure and temperature measurements from own cars can be assumed to be as well accurate but their absolute value is not so determining factor within certain limits. Only relative change will be of importance because the gradient tells about the change in the tire properties.

There are plenty of quantities that are important in the optimization but which cannot be directly measured. These are e.g. estimation of the tire condition and amount of fuel left in opponent's tank. In these measurements the uncertainties are much greater than in direct measurements from own car. Uncertainties must be taken into account in the optimization and the weight of the suggestion is lowered thereby. Some of these estimations can be made more accurate as the race goes further, e.g. amount of fuel left on the opponent can be estimated after his first pit stop better when amount of fuel filled is approximately known.

4.3. Importance

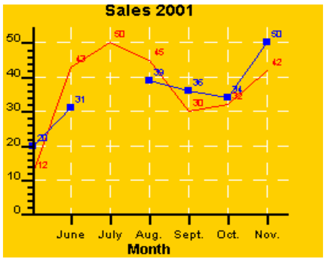
The estimations of the fuel left and tire condition for both own car and co-racers is an essential part of the optimization. This determines namely the maximum number of laps the opponents can drive before entering the pit. It is therefore of huge importance that this estimate can be made as accurately as possible. Velocity profile soft sensor derived from velocity and time measurements at the intermediates describes the conditions of the car and driver. These profiles are important when evaluating the time slot when it would be reasonable to enter the pit and there are one-on-one race going on with the opponent.

5. Preparation and preprocessing of data

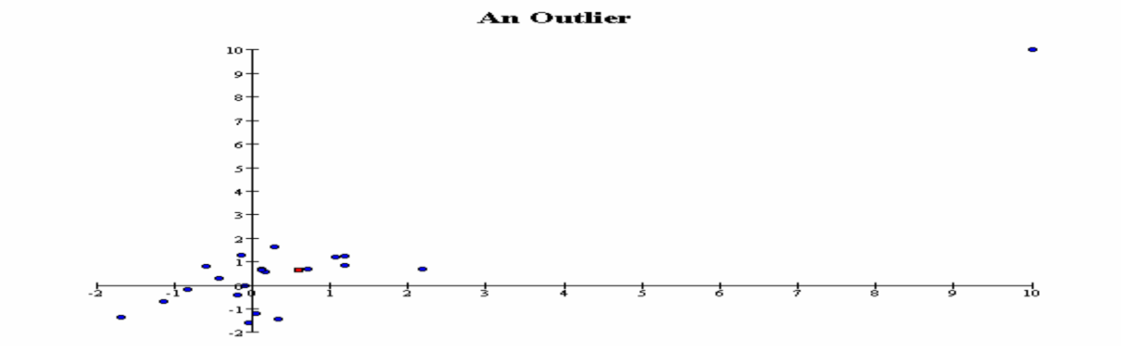
When talking about wireless transmitters and receivers, some possible failure situations must be taken into consideration. Cars with wireless transmitters are running fast and they can be quite far from receivers. Sensors are also affected by huge forces and different kinds of phenomena. These are just a few things that cause false signals and this is why signal itself is seldom in a suitable form for analysis.

Some analysis methods are quite sensitive and non-robust, so there must be a way to handle false signals first. This is why preprocessing facilities should be available both before and during the analysis.

Missing values can be real problem. If many following values are missing, analysis can be hard. Replacing missing values with mean value can be used sometimes. Sometimes zero values or possibly error mark can be used. Depending on case and analysis methods, missing values can also be left out of analysis. On a right hand picture is an example of missing values.



Outliers are almost same kind of problem values as missing values. Sometimes these kind of values just pop up, but if this happens systematically, it is important to specify the reason for outliers. This analysis also tells if these values really are outliers or is there reason for them to exist. One way of getting outliers off, is filtering. One must know what to do, so that important information is not lost during this operation. On a picture below, we can find an example of an outlier.



Usually every analysis method requires trend removal. In this case slow trends can not necessarily be removed. This tells if engine is going on wrong direction or something else slow actions happens. It is also important to know what kind of time window is used. Too tight window might give wrong information about the nature of a signal.

When comparing signals and their effects to each other, it is important to rescale signals. This means that averages and trends should be removed from signals. Also missing values and outliers must be handled first. After that, scaling is possible.

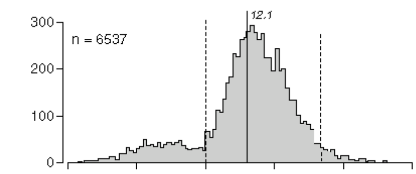
6. Data analysis methods needed to generate the functionality

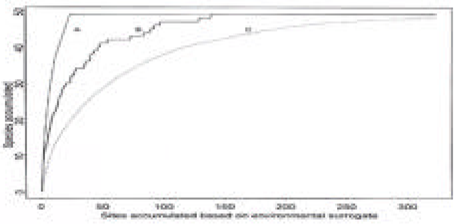
On analysis of Formula-1 car condition very much information is needed rapidly and constantly. Some of this information is needed to specify the condition of engine, amount fuel or velocity etc. Every now and then effects and correlations between signals are needed.

Difficult and complex drawings and curves are not always needed. Sometimes basic bars give enough information. For example, a bar chart gives enough information about the amount of fuel. Also pure value would be enough. Since history data is not used, this is enough only when observing current values.

A value itself does not necessarily always give the information that is needed to analyze different states of a car. To get better view of current state, deeper analysis with limits and earlier values are needed. This is the most important reason why data analysis methods are utilized.

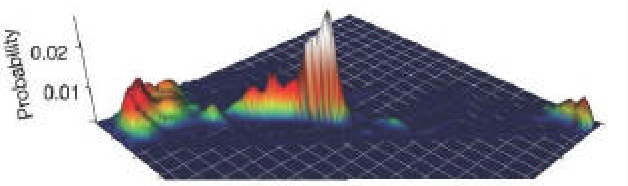
In many cases bare value or bar gives only valid information for certain situations. If we want to compare values or series of values, we'll need to give some freedom for values to vary. By examining signal histograms and comparing those to different distributions, we can get information about signal variations. Distributions also give an insight if signal is varying between allowed values. Right hand top picture shows a histogram of a measured signal. From that plot, it is quite straightforward to see the state of the car in current time.



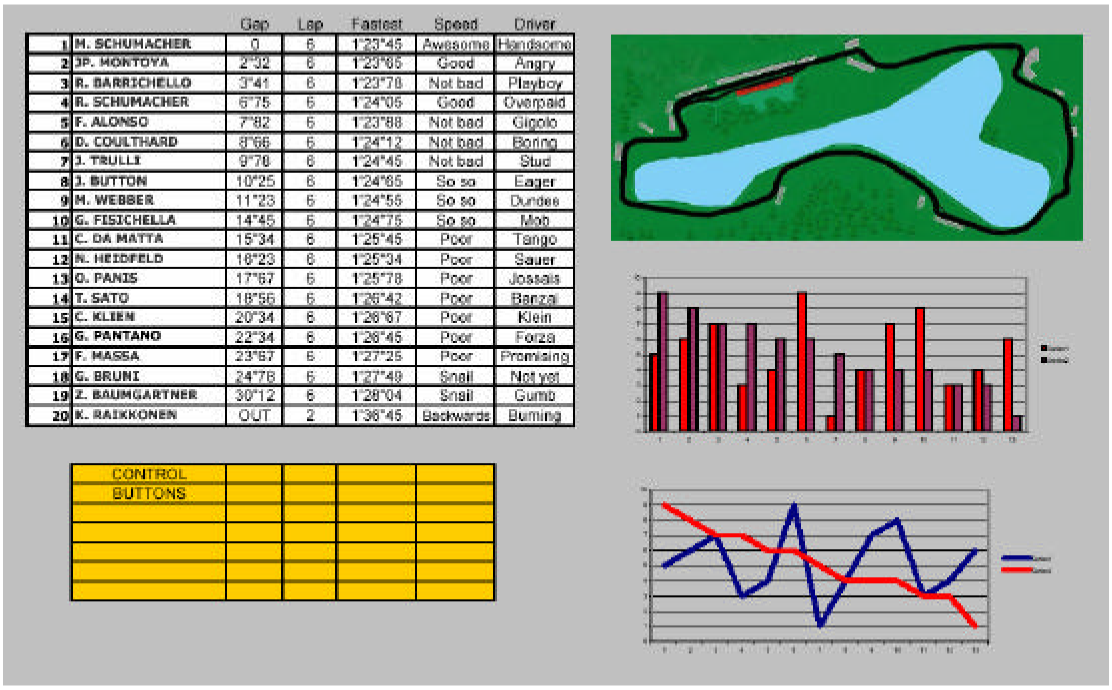


When observing signal values, the histogram and distributions are important tools. It is also necessary to give confidence limits to variation of a signal. This way we can get e.g. alerts when some important limits are broken. From right hand pictures can be seen two different signals and their confidence limits. If those limits have been exceeded, system gives alerts to the user.

Sometimes it is important to make a guess what happens in the future. Future values can be estimated by using probability distributions. We can make a guess depending on earlier measurements. It is also possible to combine different methods. For example, lap velocity can be plotted in 3D. Velocity is basically a time dependent line which is repeated on every lap. Since every lap is slightly different we can use Gaussian distribution to allow some variations on velocity data. By adding confidence limits, we can get alarms if the driver has difficulties at some point(s) of track.



7. Supporting interpretation of results



		Gap	Lap	Fastest	Speed	Driver
1	M. SCHUMACHER	0	6	1'23"45	Awesome	Handsome
2	JP. MONTOYA	2"32	6	1'23"65	Good	Angry
3	R. BARRICHELLO	3"41	6	1'23"78	Not bad	Playboy
4	R. SCHUMACHER	6"75	6	1'24"05	Good	Overpaid
5	F. ALONSO	7"82	6	1'23"88	Not bad	Gigolo
6	D. COULTHARD	8"66	6	1'24"12	Not bad	Boring
7	J. TRULLI	9"78	6	1'24"45	Not bad	Stud
8	J. BUTTON	10"25	6	1'24"65	So so	Eager
9	M. WEBBER	11"23	6	1'24"55	So so	Dundee
10	G. FISICHELLA	14"45	6	1'24"75	So so	Mob
11	C. DA MATTA	15"34	6	1'25"45	Poor	Tango
12	N. HEIDFELD	16"23	6	1'25"34	Poor	Sauer
13	O. PANIS	17"67	6	1'25"78	Poor	Jossais
14	T. SATO	18"56	6	1'26"42	Poor	Banzai
15	C. KLIEN	20"34	6	1'26"67	Poor	Klein
16	G. PANTANO	22"34	6	1'26"45	Poor	Forza
17	F. MASSA	23"67	6	1'27"25	Poor	Promising
18	G. BRUNI	24"78	6	1'27"49	Snail	Not yet
19	Z. BAUMGARTNER	30"12	6	1'28"04	Snail	Gumb
20	K. RAIKKONEN	OUT	2	1'36"45	Backwards	Burning

Figure 1. Main display of tactics monitor

Figure 1 shows the user interface of the decision making support system. All results are presented on a single monitor. The display is roughly divided into two parts in vertical plane. The left hand side contains numerical values for measurements and the right hand side includes graphical presentations about current situations.

Live results and control buttons item are always presented and can not be hided or changed. Other tables and figures are modified according to desires of the user. From the control buttons –item the user can change the outlook of the display.

Below the current race situation graph the result of the scenario probing is shown if desired. The result is calculated based on predetermined pit stop strategy of the opponents and own cars. Current situation and user input values are taken into account in the scenario probing.

All the measurement results can be presented as numerical values on top left hand tables. All information can be downloaded from team's database. Suggestions for better tactics can also be shown in tables.

8. Project proposal

This proposal has been done to help making new and more efficient tactics to F1-race. This proposal includes ca. nine man years, pressed in to 23 months. Time table of the project is presented on Appendix B.

First three stages require 6 full time employees. Next two needs 10 full time workers. System will be ready to use in ten months. After that starts user guidance and full support for next year. Last section requires two full time workers.

Background research provides easy and informative start to the project. During this part customer gives enough information to the contractor about measurements etc.

UR is composed in co-operation with the customer. This section collects all information that customer needs and requires on the system.

FS is functional specification done from URs. This tells programmers how things should work.

Detailed design specs are composed from functional specifications and are used to get the background for programming the system.

Implementation is done after the system has reached certain stability aims.

When the implementation is done, starts customer guidance. This gives customer adequate information about the system. The training aims at problem free and effective use of the tool.

Full support starts when system is implemented to the customer's main unit and customers has been trained to use the tool.

Appendix A: List of measurements

Velocity at intermediates

Time at intermediates

Amount of fuel left in the tank

Tire pressure

Tire temperature

Ambient conditions measurements:

- Temperature
- Air pressure
- Humidity

Appendix B: Project proposal time table

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Background research																							
UR																							
FS																							
Detailed design specs																							
Implementation																							
User guidance																							
Full support																							