SAAPS

Satellite Anomaly Analysis and Prediction System

Software Requirements Document

Version 1.0

ESA/ESTEC Contract No. 11974/96/NL/JG(SC)

P. Wintoft

18 August 2000

Document status sheet

SAAPS SRD	
Version	Date
0.1	12 December 1999
0.2	3 February 2000
0.3	3 April 2000
0.4	22 June 2000
1.0	18 August 2000

Contents

Chapter 1

Introduction

1.1 Purpose

The Satellite Anomaly Analysis and Prediction System (SAAPS) contain 3 subsystems: the database and database tool (DB&T), the satellite anomaly analysis module (SAAM), and the satellite anomaly prediction module (SAPM). The user requirements for each subsystem are described in the URD-WP110 (DB&T) [1], URD-WP210 (SAAM) [2], and the URD-WP310 (SAPM) [3]. The purpose of this document is to specify the requirements of the software system from the developers point of view. The SRD incorporates the user requirements described in the URDs.

1.2 Definitions, acronyms and abbreviations

DB&T Database and database tools

DBO Database object

DBT Database tool

MDBO Meta-database object

SAAM Satellite anomaly analysis module

SAAPS Satellite anomaly analysis and prediction system

SAPM Satellite anomaly prediction module

SPEE Study of Plasma and Energetic Electron Environment and Effects

TSO Time series object

URL Uniform resource locator.

1.3 References

1. URD-WP110.
2. URD-WP210.
3. URD-WP310.
4. ESA software engineering standards, ESA PSS-05-0 Issue 2, February 1991.
5. Guide to the user requirements definition phase, ESA PSS-05-02 Issue 1, October 1991.
6. Lundstedt, H., The Swedish space weather initiatives, *Workshop on Space Weather, 11-13 Nov. 1998, ESTEC*, WPP-155, 197–205, 1999.
7. Wintoft, P., Satellite Anomaly Prediction Module, *SAAPS, Technical Note 3*, 2000.
8. Wintoft, P., Satellite Anomaly Analysis Module, *SAAPS, Technical Note 2*, 2000.

1.4 Overview

The structure of this document follows the document template as described in the ESA software engineering standards [4] and user requirements guide [5].

Section 2 describes, in general terms, the software requirements of SAAPS. In Section 3 the specific requirements are given.

Chapter 2

General description

2.1 Relation to current projects

At IRF-Lund a programme called the Lund Space Weather Model (LSWM) [6] have been running for several years. The model aims at modelling and describing several processes that are important for the space weather. It also takes into account the effects of space weather on technological systems.

Since April 2000 the ESA Space Weather Programme is running. The main contractor is Alcatel Space and one of the subcontractors is IRF-Lund.

The SAAPS project fits well into both programmes.

2.2 Relation to predecessor and successor projects

New research results and ideas have emerged from the Lund Space Weather Model which are incorporated into SAAPS.

The direct predecessor to SAAPS was the ESA/ESTEC project Study of Plasma and Energetic Electron Environment and Effects (SPEE). The findings in SPEE led to the initiation of SAAPS.

2.3 Function and purpose of SAAPS

The SAAPS user requirements are specified in three separate URDs ([1],[2],[3]), each capturing the requirements of the three subsystems: database and database tools (DB&T), satellite anomaly analysis module (SAAM), and satellite anomaly prediction module (SAPM).

The purpose of SAAPS is to provide a tool for the analysis and prediction of satellite anomalies. As the users of SAAPS will work in many different geographical regions, and at the same time they share a common database, it is decided that SAAPS shall be operated over the Internet.

2.3.1 Database and database tools

The purpose of the DB&T is to maintain a database of space weather and anomaly data, and to provide this data to the SAAM and SAPM. The DB&T shall therefore have functions to build the database from existing databases. To keep the SAAPS database up to date with real time data the DB&T shall also have functions to automatically retrieve and store new data into the database. This updating ranges from once every 10 minutes to once every day.

2.3.2 Satellite Anomaly Analysis Module

The SAAM shall make it possible to perform various analyses on the data and models in SAAPS. The model has interfaces both to DB&T and SAPM.

2.3.3 Satellite Anomaly Prediction Module

The SAPM shall consist of several different models for the prediction of both space environment parameters and satellite anomalies. The models are driven by the data in the SAAPS database. There are two modes of operation: run models on historic data, run models in real time.

2.4 Model description

The SAAPS model is naturally divided into the three subsystems defining the function of SAAPS, namely the database and database tools (DB&T), the satellite anomaly analysis module (SAAM), and the satellite anomaly prediction module (SAPM). The DB&T can work by itself and do not depend on SAAM or SAPM.

From a users point of view the entrance point to SAAPS will be a web page where he can select whether to run the real time predictions module or the analysis module. In addition to this there shall be links to help pages. The regular user shall thus never be able to directly run the DB&T.

From the developers point of view it is convenient to add another subsystem which contain objects and functions that are shared by the other subsystems. This is the UTIL subsystems and contain a class for time series objects and plotting functions. Thus, we have four subsystems which will be discussed below.

2.4.1 UTIL

2.4.1.1 Time series objects

A central part in the SAAPS are the handling of time series data. We therefore define a class of time series objects (TSOs). The TSO shall always have the format described in Table 2.1. The time field contains the time of the observations and the data fields the values. The data field can e.g. be the electron flux or spacecraft position. To each data field there shall also exist associated field name and unit. The field name and unit could e.g. be "Bx" and "nT" for the solar wind magnetic field component in the x-direction in nT. The time field shall increase strictly monotonically, i.e. $t_1 < t_2 < \ldots < t_m$. If the TSO is continuous then the observations are sampled at equidistant times, i.e. $t_i = t_{i-1} + \Delta t$ where Δt is the sample interval. If there are any data gaps these will be indicated with NaN, Not a Number.

Table 2.1: The format of the time series object (TSO).

Time field	Data fields			
t_1	$x_{1,1}$	$x_{1,2}$	$\cdots$	$x_{1,n}$
t_2	$x_{2,1}$	$x_{2,2}$	$\cdots$	$x_{2,n}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
t_m	$x_{m,1}$	$x_{m,2}$	$\cdots$	$x_{m,n}$

When a TSO is created from an external source, e.g. via ftp, it shall be ensured that the above holds. Typically, the data received extends over a certain time period and contain several fields. It must thus be checked that the time field increases strictly monotonically. If this is not the case it must be sorted and possible double times be removed. If the time series is continuous it shall also be checked that all times in the received time interval exist. If some times are missing these shall be created and the data fields shall be set to NaN.

It shall also be possible to merge two TSOs to create a new TSO. This is useful for e.g. a situation when one wish to create a plot from two TSOs. As the two TSOs generally do not have the same sample interval or time coverage this has to be handled. The situation for TSOs with different time coverage is illustrated in Figure 2.1. One thus has to state whether the first or the second TSO should determine the start time of the new TSO, and whether the first or the second TSO should determine the end time. If the two TSOs have different sample rate and sample times one must specify which sampling that should be used. One way to achieve this is to resample the times in one TSO to that of the other TSO. Another way is to explicitly specify the sample rate and times with a possible resample of both TSOs.

With the TSO we have a well defined description of the data that flows within SAAPS. As an example Figure 2.2 shows the steps from retrieving a TSO from the database, perform preprocessing, running a model, and presenting the result as a plot.

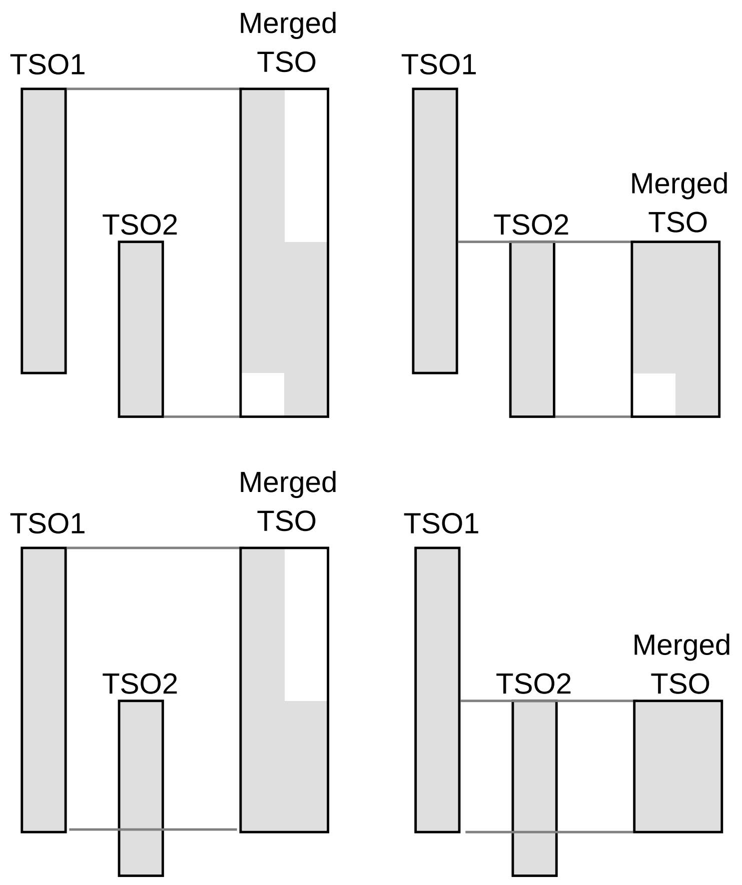

Figure 2.1: The merging of two TSOs. In this case the merging can be made in one of four possible ways. The shaded areas represents the data and the white areas NaNs.

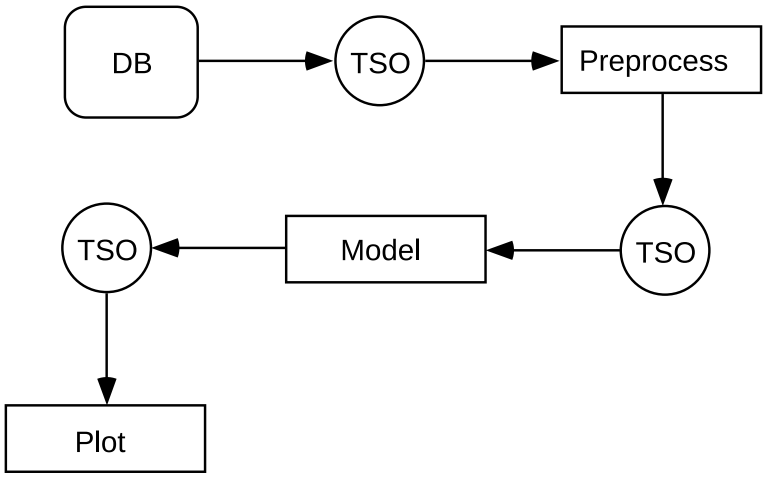

Figure 2.2: The flow of data from the database (DB) using a time series object (TSO) that is preprocessed and run in a model to produce a plot.

2.4.1.2 Plotting functions

The plotting functions applies to TSOs. A plot can be produced as a time series plot or as a general plot.

In a time series plot the data values are plotted as a function of time. The time field is formatted so that the meaning becomes clear on the plot. The formatting can be automatic or manual. With automatic formatting appropriate labels and tick marks are determined from the length of the time series. If e.g. the plot extends over a few days then the tick marks should be in days while the label should contain the year and month. With manual formatting the tick marks and labels must be defined. It must also be stated which data fields that should be plotted.

In a general plot one defines which field that should be on the horizontal axis which field(s) that should be used on the vertical axis.

The plot shall also contain a legend that show the names and units of the data. This information is extracted from the TSO.

Frequently the numerical values of the different data fields in a TSO have very different ranges which makes it difficult to place the curves in one figure. Therefore, there shall also be a function for producing stack plots.

Other functions that shall be implemented are the possibilities for log-plots and selecting line and marker style.

2.4.2 Database and database tools, DB&T

The access to the database is via the database tools (DBT). The DBT can be run by the system manager, automatically by a schedule program, and by the SAAM and SAPM.

The database contain two types of objects: database objects (DBO) and meta-data of the DBO (MDBO). These two object types will first be described and then the various functions for reading and storing data from/to the database will be explained.

2.4.2.1 Meta database objects, MDBO

The MDBO contains a description of the DBO. This includes the name of the DBO, a description of the data, the times of the first and last instances of data, whether it is a continuous time series and if so the sample interval, and the names and units of the data fields. The description of the data shall contain information so that the data source is well described. E.g. if the data comes from a satellite then it should include the satellite name, instrument name, and principal investigator. The MDBO fields are summarized in Table 2.2. For each DBO that shall be stored into the SAAPS database the system manager must enter the metadata fields. The fields that must be entered so that the database will work are the DBO name, Continuous, and Sample interval. The Description, Field names, and Field units are not necessary for the function of the database but should also be entered. The Start and End times are not set manually, but extracted from the data.

Table 2.2: The format of the meta data.

Name	Description
DBO name	Name of the DBO
Description	Description of the data
Start time	Time of the first data in the DBO
End time	Time of the last data in the DBO
Continuous	True if the DBO is continuos, else false
Sample interval	If continuous, the time step between samples
Field name 1	Name of the first data field
Field name 2	Name of the second data field
⋮	⋮
Field name n	Name of the nth data field
Field unit 1	Unit of the first data field
Field unit 2	Unit of the second data field
⋮	⋮
Field unit n	Unit of the nth data field

2.4.2.2 Create metadata

The first time a DBO is stored to the database the MDBO is also created and stored. All the necessary fields must have been entered as described above. The fields Start and End times will be extracted from data that are to be stored in the database.

2.4.2.3 Update metadata

All fields in the metadata can be changed. However, in practice it is only the time of the last observation that will be altered when the database is updated with new data.

2.4.2.4 Remove metadata

Removing metadata is to be considered as a very rare event. The system manager always have the possibility to remove a MDBO from the file system.

2.4.2.5 Read metadata

The metadata is read by supplying the name of the corresponding DBO. This is useful when inspecting the database and to provide information to other functions.

2.4.2.6 Metadata interface

There shall be a graphical interface so that the metadata can be easily inspected. The program will only run on the SAAPS server and not be available from any web page. For the SAAPS system manager it shall also be possible to change the metadata in the database. The interface shall list all the DBOs and after one has been selected the metadata will be displayed. If it is the SAAPS system manager who runs the program he can then alter the data which are not critical to the function of the database. Changeable items includes the description of the data, the field names, and the field units.

2.4.2.7 Database objects, DBO

The DBO are stored in the database and has the same structure as the TSO. It shall only be possible to store TSOs as DBOs. This ensures that the data that are written to the database are well defined and has a known structure.

The database objects can generally by of one of two types: contiguous data or non-contiguous data. As a major part of the database will contain contiguous data the model will be optimized for this. One database object might e.g. contain several years of 1 minute resolution data and it must be ensured that any given time period can be uploaded quickly.

The satellite anomaly data in the database shall have a special status due to the confidentiality issues. The system manager shall always have full access to the database. A user, who always enters via the SAAM or SAPM, shall only retrieve summary information about the anomalies in a way so that the source will never be disclosed.

2.4.2.8 Create a DBO

Initially the database does not exist, but must be built. To perform this the data source must be selected, which e.g. can be a URL. Then the data format and the relevant parameters must be identified. This information is provided to the DBT and an object in the database is created. This involves the writing of code that contain the metadata information.

Data can now be stored in the database object. When a DBO is created the first time the MDBO is also created. In the process of building the database the system manager must manually run the DBT to add new data for an existing database object.

2.4.2.9 Remove a DBO

Removing data is to be considered as a very rare event. The system manager always have the possibility to remove a DBO from the filesystem.

2.4.2.10 Update a DBO

Existing database objects are updated from TSOs. The TSO can e.g. be created automatically at a specified interval by a scheduling program. When the DBO is automatically updated it will grow with time. Taking the ACE magnetic field data as an example, which contain the time field plus six data fields for every minute, it will grow with (7 fields) $\times$ (8 bytes/field) $\times$ (60 samples/h) $\times$ (24 h/year) $=$ 80640 bytes per day (79 kB/day) or 29 MB/year.

2.4.2.11 Interface to SAAM and SAPM

The DBT shall provide interfaces to the SAAM and the SAPM. The interfaces are of the type that a request for data comes from SAAM/SAPM and the DBT then returns the data. The SAAM/SAPM can thus only retrieve data from the database and never alter the database.

2.4.2.12 Storing data

For a continuos TSO that shall be stored into the database a few things must be checked. If there is a time overlap between the TSO and the database object (DBO) then the times in the TSO must be the same as the times in the DBO for the overlapping period, and if this is not the case a new DBO will be created from the TSO and the system manager shall be notified.

The data values from the TSO shall be written into the existing places of the DBO for the overlapping period. This means that any NaN in the DBO may be replaced by a value from the TSO. If a value in the DBO exist and is overwritten by a new and different value from the TSO then the system manager shall be notified.

If there is no overlap between the TSO and the DBO then the times between the last time in the DBO and the first time in the TSO shall be created and the data fields shall be set to NaNs.

The three situations are illustrated in Figure 2.3.

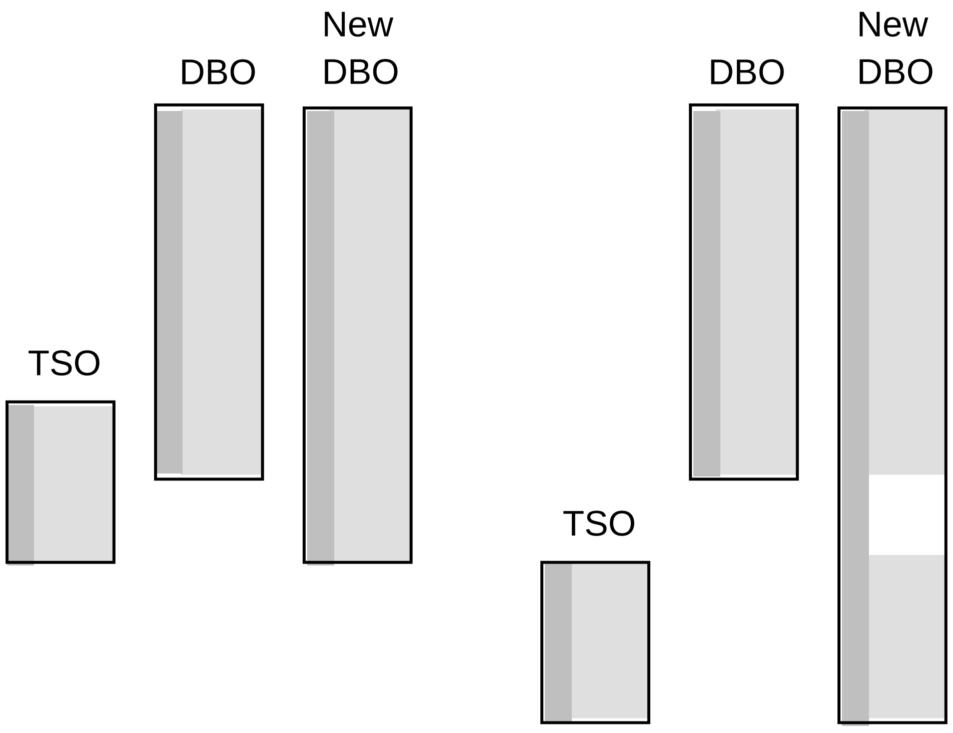

Figure 2.3: The storing of a TSO to the database. The time field is illustrated with the dark gray area and the data fields by the light gray area. Missing data (NaNs) are left blank.

2.4.2.13 Retrieving data

A TSO can be retrieved from the database. Five parameters must be given: the name of the DBO, the data fields, the start time, the end time, and the interval type (open, closed, semi-open, see [8]). The parameters are summarized in Table 2.3.

Table 2.3: Parameters needed for retrieving data

DBO name	A string with the DBO name to be accessed
Data fields	An array of strings or integer pointing at the data fields
Start time	The start time of the data to be read
End time	The end time of the data to be read
Interval type	Open, closed, open-closed, or closed-open

2.4.2.14 Retrieving resampled data

Many times the desired time resolution of a TSO does not match the time resolution of the DBO. Therefore, it shall be possible to retrieve resampled data from the DBO. Again, the start and end times, and the interval type must be given. In addition the sample interval the averaging window must also be given. The parameters are summarized in Table 2.4.

Retrieving resampled data is different from retrieving data followed by a resample; retrieving resampled data only needs memory for the final TSO while the retrieve and resample needs memory for the intermediate TSO. To exemplify we might want to plot one year of hourly solar wind velocity. This data will need $365 \times 24 \times 8 = 70080$ bytes of memory (68 kB). If we first were to retrieve the full time resolution data we would need $365 \times 24 \times 60 \times 8 = 4204800$ bytes (4 MB).

Table 2.4: Parameters needed for retrieving resampled data

DBO name	A string with the DBO name to be accessed
Data fields	An array of strings or integer pointing at the data fields
Start time	The start time of the data to be read
End time	The end time of the data to be read
Interval type	Open, closed, open-closed, or closed-open
Sample interval	The sample interval of the final TSO
Window start	The start of the averaging window from each sample point
Window end	The end of the averaging window from each sample point
Skip NaNs	True if any NaNs in the DBO should be removed before averaging

2.4.2.15 Retrieving data from multiple database objects

Another common situation is were a model needs data from different data fields in different database objects. As different DBOs usually have different time resolutions they need to be resampled to a resolution defined by the model or the user. The same information is needed as when retrieving resampled data together with a list of database objects and the respective data fields. The parameters are summarized in Table 2.5.

Table 2.5: Parameters needed for retrieving multiple database objects.

DBO names	An array of strings with the DBOs to be accessed
Field names	An array of an array of strings or integers with the data fields to be accessed
Start time	The start time of the data to be read
End time	The end time of the data to be read
Interval type	Open, closed, open-closed, or closed-open
Sample interval	The sample interval of the final TSO
Window start	The start of the averaging window from each sample point
Window end	The end of the averaging window from each sample point
Skip NaNs	True if any NaNs in the DBO should be removed before averaging

2.4.2.16 Converting to/from ASCII

The data in the database are stored in an internal SAAPS format. It shall be possible to convert the data to/from ASCII data.

To convert a DBO to ASCII the following parameters must be given: the name of the DBO, the time interval and the fields. This will produce a space delimited table with one row for each observation. The ASCII output will contain the given fields. To specify how the time should be formatted a combination of the following field names can be used: `YEAR MONTH DAY HOUR MINUTE SECOND`. The metadata can also be included in the head using `#` as the comment symbol.

To convert ASCII data to a DBO the format of the ASCII data must conform to the metadata description. The ASCII data shall be in a space delimited format.

2.4.2.17 Building the database

To build the database involves writing code that converts the data from some source to the SAAPS database. The code will be provided for the data described in the DB&T URD [1].

2.4.2.18 Automatic update of the database

A schedule service shall run on the SAAPS server that automatically runs data downloads at regular intervals, e.g. every 10 minutes or every hour. The program shall connect to the server that provides the data, download the data, create a TSO, and store it in the SAAPS database. Various checks and preprocessing are made, as described in Section 2.4.1.1, when the TSO is created.

The data described in the DB&T URD [1] shall be included in the automatic update.

2.4.3 Satellite Anomaly Analysis Module, SAAM

The functions in the SAAM are described below. All the functions shall be available from a web browser such as Netscape Communicator or Microsoft Internet Explorer.

2.4.3.1 Data plotter

The data plotter provides a tool for plotting the data in the database. This function can not be used on the satellite anomaly data. This function under SAAM is dedicated to analyse data. There shall also be a data plotter function under SAPM that is used for real time plots. Essentially the two functions are the same but the SAAM plotter have more plotting options while the SAPM plotter will be a faster way to achieve real time plots.

The first step is to collect the data from the database. The user must specify the time period and the time resolution of the data. Then the DBOs and data fields must be selected and finally the data is fetched. This will produce a TSO.

The next step is to produce the plot from the TSO. The user should select which data field that should go on the x-axis of the plot and which data field or fields that should go on the y-axis. The user can also select whether one or both of the axes should have a logarithmic scale.

2.4.3.2 B-L coordinate transformation

It shall be possible to transform from geographic cartesian or spherical coordinate systems to the B-L coordinate system. A tilted dipole magnetic field will be assumed. The positional data from a TSO are passed to the function which will return the new coordinate values in the B-L coordinate system.

2.4.3.3 Linear interpolation

To further process the TSO it may be necessary to first replace data gaps with linear interpolation. To perform this it must be stated how many contiguous positions with data gaps that shall be replaced by the linearly interpolated value [8].

2.4.3.4 Average and resample

To average a time series three parameters must be given: window size, time offset, and sample interval. The window size determines the resolution of the averaged time series. The time offset is the time with which each average is associated with. Finally, the sample interval is the time steps of the averaged time series. To give an example we have the ACE 1-minute solar wind data that we wish to average to 1-hour resolution. Then the window size is 1 hour. If we want a central average [8] then the time offset is 0 hours, whereas a lagging average would have a time offset of +0.5 hours. With a sample interval of 1 hour means that we will have one 1-hour-average per hour, whereas a sample interval of 1 minute will give sixty 1-hour-averages per hour.

2.4.3.5 Statistics

Various functions to calculate statistics on the data shall be available. These are the mean, standard deviation, root-mean-square error, and linear correlation.

The mean and standard deviation takes a vector or matrix as the argument and returns a scalar value or a row vector, respectively. For the vector input the calculations are carried out over each element and the result is thus a scalar value. For the matrix input the calculations are carried out over each column in turn and the result is a row vector with the number of columns equal to the number of columns in the input matrix.

The root-mean-square error and linear correlation takes two vectors of equal length or a matrix as the input argument. The output is the value of the correlation coefficient in the first case, and a square symmetric matrix with correlation coefficients in the second case where each row in the input matrix is an observation.

2.4.3.6 Superposed epoch analysis

In superposed epoch analysis (SEA) two TSOs are compared to each other. The first TSO is the data that should be superposed and the second TSO contain the list of events that should be used for the analysis. The time interval over which the SEA should be made shall also be given. Finally, the SEA time window [8] shall be given. Thus, to perform SEA four parameters must be given: data TSO, event TSO, time interval, and time window. The result from the SEA is a new TSO which extends over a time interval equal to the SEA window (See Figure 2.4).

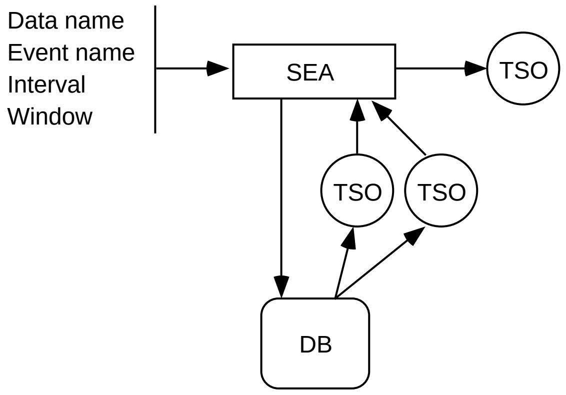

Figure 2.4: Superposed epoch analysis.

The components of the user interface to perform SEA are the event list, the start and end times of the SEA window, the DBO list, the data field list, the time resolution, and an option to use interpolated data.

The event list is a space delimited table with one event on each row having the format `YYYY MM DD hh mm ss`. Fields from the right may be omitted so a valid line could be `YYYY MM DD hh`. The event times do not need to be sorted. The event list is created by inserting the text directly into the window which can be done by directly typing the text or by copy-and-paste from another text window. Then the size of the SEA window must be entered, which can be given in minutes, hours, or days.

Next, a DBO and a data field must be selected and the desired time resolution (in minutes, hours, or days). One may also select whether there should be any interpolation over data gaps or not.

Finally, a calculate button is pushed and the plot appears in a new window. In this window one can select whether the data should be displayed as average or sum, and if the y-axis should be logarithmic.

2.4.3.7 Cluster analysis

The cluster analysis function shall provide tools to plot anomaly data as function of any two parameters in the database. E.g. plotting the occurrence of the anomalies in a longitude-latitude coordinate system would provide information whether they are clustered around the south atlantic anomaly. Examples of other coordinate systems could be the magnetic B-L or local time-electron flux.

The anomaly data can be either the from the SAAPS database or from user input via the web browser.

2.4.3.8 Pattern search

The pattern search function enables the user to search for specific patterns in the database. A search pattern may be the result from a previous superposed epoch analysis. The pattern describes how a specific parameter varies over a limited time period. To perform a search limits must be set on how close a possible match must be to the search pattern. The pattern may also be normalized so that it is only the shape that is important and not the level.

2.4.3.9 Guidelines

From each web page there shall be associated help pages that guides the user through the various steps to perform an analysis.

2.4.3.10 Estimate best model

The user shall be able to estimate which anomaly prediction model that best fits his own data. From the web page he provides his own anomaly set. Then different SAAPS models are run for the submitted time periods and the model that best match the anomalies is suggested.

2.4.4 Satellite Anomaly Prediction Module, SAPM

The SAPM shall contain different models for the prediction of satellite anomalies and the space environment. A model uses data from the database and runs a neural network to produce the prediction. The models can either be run from the SAAM or from a web page to produce real time predictions based on the latest data.

Here we will describe the components of SAPM. The main component of SAPM is the prediction model, which is further divided into the pre-processing, neural network, and post-processing. There shall also be a web interface to run the models in real time.

2.4.4.1 Prediction model

To run a prediction model two things are needed: 1) the model name, and 2) the time interval over which the prediction should be made. Then the model is loaded which contain a description of which data fields that shall be used, how it should be preprocessed, the parameters of the neural network, and how the output should be post-processed. Next, the TSOs are loaded from the database (Figure 2.5) for the specified time interval and passed to the pre-processor (Section 2.4.4.2). The output from the pre-processor is a matrix which is the input to the neural network (Section 2.4.4.3) which in turn produces the output matrix. Finally, the output matrix is post-processed (Section 2.4.4.4) to produce the TSO. The TSO can then be used by e.g. the SAAM or the real time web interface.

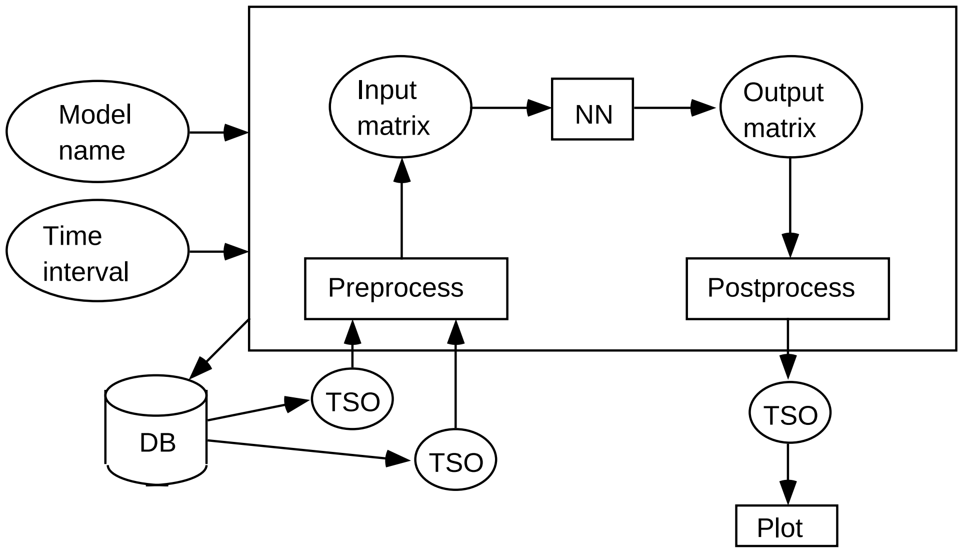

Figure 2.5: The prediction model. Inputs are model name and time interval. Then the data is loaded from the database and passed through the model to produce a TSO.

2.4.4.2 Pre-processing

To be able to run the neural network the data must first be preprocessed.

If time is needed explicitly it must be transformed into a useful format. It is always assumed that it is some cyclic representation of time that is needed. E.g. if there is a local time dependence then transformation is $\cos(2\pi\mathrm{LT}/24)$ and $\sin(2\pi\mathrm{LT}/24)$.

The data fields of the TSOs must be normalized to put the different input parameters into the same numerical ranges. The data is transformed from the range (low, high) to the range $(-0.8, +0.8)$. The transformation is either linear or logarithmic. The low and high values for each parameter are defined in the prediction model.

Then the TSOs must be rearranged into a matrix that fits the input of the neural network. The pre-processor extract the data fields that are to be used and introduces possible time delays. The prediction model contains the description of the transformation from TSO to the input matrix.

2.4.4.3 Neural network

The standard feed-forward neural network shall be implemented. The network is specified by the number of inputs, the number of layers, the number of neurons for each layer, and the different layers transfer functions. The actual function that the network implements is described by its weights, biases, and transfer functions. The prediction model contains definition of the neural network.

The network weight matrix and bias vector for layer l have the following forms:

$$\mathbf{W}_l = \begin{bmatrix} w_l(1,1) & w_l(1,2) & \dots & w_l(1,n_{l-1}) \\ w_l(2,1) & w_l(2,2) & \dots & w_l(2,n_{l-1}) \\ \vdots & \vdots & \ddots & \vdots \\ w_l(n_l,1) & w_l(n_l,2) & \dots & w_l(n_l,n_{l-1}) \end{bmatrix} \tag{2.1}$$

and

$$\mathbf{b}_l = \begin{bmatrix} b_l(1) \\ b_l(2) \\ \vdots \\ b_l(n_l) \end{bmatrix}. \tag{2.2}$$

The number of inputs to each neuron in layer l equals n_{l-1}. The number of neurons, which is the same as the number of outputs from the layer, equals n_l. The input layer has $l = 0$, the hidden layer $l = 1$, and the output layer $l = 2$. The weights connecting the input layer to the hidden layer are thus described in the matrix $\mathbf{W}_1$.

The input matrix is obtained from the pre-processing function and has the form

$$\mathbf{X} = \begin{bmatrix} x(1,1) & x(1,2) & \dots & x(1,q) \\ x(2,1) & x(2,2) & \dots & x(2,q) \\ \vdots & \vdots & \ddots & \vdots \\ x(n_0,1) & x(n_0,2) & \dots & x(n_0,q) \end{bmatrix}, \tag{2.3}$$

where n_0 is the number of inputs and q is the number of examples.

The output from the neural network is a new matrix as

$$\mathbf{Y} = \begin{bmatrix} y(1,1) & y(1,2) & \dots & y(1,q) \\ y(2,1) & y(2,2) & \dots & y(2,q) \\ \vdots & \vdots & \ddots & \vdots \\ y(n_2,1) & y(n_2,2) & \dots & y(n_2,q) \end{bmatrix}, \tag{2.4}$$

where n_2 is the number of output neurons.

2.4.4.4 Post-processing

The first step in the post-processing is to transform the output matrix in to a TSO.

The time field in the output TSO is calculated from the time field in the input TSO by adding the ahead-prediction time. The prediction model contains the information of the ahead-prediction time.

The output must also be de-normalized into real physical values (e.g. electron flux) or classes (e.g. no-anomaly, anomaly). The de-normalization information is in the prediction model.

2.4.4.5 Interface

There shall be a web interface from which the different prediction models can be accessed and run. It shall use the latest data to produce real time forecasting. The data shall be visualized in a suitable format such as plots or text.

2.4.4.6 On-line manuals

To guide the user on how to use SAPM there shall be online manuals.

Chapter 3

Specific requirements

Here we list the specific requirements as derived from the URDs and the model description.

The complete SAAPS systems contain the subsystems DB&T, SAAM, and SAPM. The DB&T interacts with other systems to retrieve data in real time. The user interacts with SAAPS via the SAAM and SAPM subsystems. However, on the developer level we introduce another subsystem that contain general software that are used mainly by SAAM and SAPM, namely, UTIL. UTIL is the utility subsystem.

Each requirement has the following components:

Identifier	A unique identifier to be used for
Need	How essential the function is.
Priority	The priority of the function.
Stability	If it is expected that this functions will be same or not over the project development.
Source	Reference to the URD specific requirement.
Name	A unique name.
Description	A short and concise description of the function.

3.1 saaps.util

3.1.1 Functional requirements

Requirement	Functional
Identifier	util.fun.001
Need	Essential
Priority	High
Stability	Stable
Source	All
Name	Util.TSO
Description	The time series object is the basic container for time series data. The time series object is used in many functions by DB&T, SAAM, and SAPM.

Requirement	Functional
Identifier	util.fun.002
Need	Essential
Priority	High
Stability	Stable
Source	SAAM.CAP.2, SAPM.CAP.7
Name	Util.Plot
Description	Plot time series data. The function provides basic operations such as possibility to select logarithmic scaling, axis limits, line and/or point plots, and zooming.

3.2 saaps.dbt

3.2.1 Functional requirements

Requirement	Functional
Identifier	dbt.fun.001
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.7
Name	Database.storeNew
Description	Store a TimeSeries object into the database for a new database object.

Requirement	Functional
Identifier	dbt.fun.002
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.8
Name	Datebase.store
Description	Store a TimeSeries object into the database for an existing database object.

Requirement	Functional
Identifier	dbt.fun.003
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.10
Name	Database.retreive
Description	Retreive a TimeSeries object from the database for a single database object for a specified time interval and the specified data fields.

Requirement	Functional
Identifier	dbt.fun.004
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.10
Name	Database.retreiveResampled
Description	Retreive a resampled TimeSeries object from the database for a single database object for a specified time interval and the specified data fields. The sampling rate and averaging window must be given.

Requirement	Functional
Identifier	dbt.fun.005
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.10
Name	Database.retreiveMulti
Description	Retreive a resampled TimeSeries object from the database for multiple database objects for a specified time interval and the specified data fields. The sampling rate and averaging window must be given.

Requirement	Functional
Identifier	dbt.fun.006
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.11
Name	Database.remove
Description	Remove a database object from the database.

Requirement	Functional
Identifier	dbt.fun.007
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.12
Name	Database.saaps2ascii
Description	Convert SAAPS data into an ASCII file. The name of the database object and the time interval must be specified. The output is either to the terminal window on the SAAPS server or to a file.

Requirement	Functional
Identifier	dbt.fun.008
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.13
Name	Database.ascii2saaps
Description	Convert ASCII data to a SAAPS time series object. The format of the input must be specified, either using an existing database object definition or explicitly specifying number of fields, field formats, and field units.

Requirement	Functional
Identifier	dbt.fun.009
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.9
Name	Metadata.retreive
Description	Retreive the meta data for a specific database object.

Requirement	Functional
Identifier	dbt.fun.010
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.3
Name	Metadata.store
Description	Store the meta data for a specific database object.

Requirement	Functional
Identifier	dbt.fun.011
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.4
Name	Metadata.create
Description	Create the meta data for a specific database object.

Requirement	Functional
Identifier	dbt.fun.012
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.5
Name	Metadata.update
Description	Update the meta data for a specific database object.

Requirement	Functional
Identifier	dbt.fun.013
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.6
Name	Metadata.remove
Description	Remove the meta data for a specific database object.

Requirement	Functional
Identifier	dbt.fun.014
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.2
Name	Database.RealTime.ACE.mag
Description	Retreive the real time ACE magnetic field data and store into the database. The updating is made every 10 minutes.

Requirement	Functional
Identifier	dbt.fun.015
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.2
Name	Database.RealTime.ACE.swe
Description	Retreive the real time ACE plasma data and store into the database. The updating is made every 10 minutes.

Requirement	Functional
Identifier	dbt.fun.016
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.2
Name	Database.RealTime.GOES.part
Description	Retreive the real time GOES particle data and store into the database. The updating is made every 10 minutes.

Requirement	Functional
Identifier	dbt.fun.017
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.2
Name	Database.RealTime.GOES.xray
Description	Retreive the real time GOES X-ray data and store into the database. The updating is made every 10 minutes.

Requirement	Functional
Identifier	dbt.fun.018
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.2
Name	Database.RealTime.Dst
Description	Retreive the real time provisional D_{st} and store into the database. The updating is made every 24 hours.

Requirement	Functional
Identifier	dbt.fun.019
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.2
Name	Database.RealTime.Kp
Description	Retreive the real time estimated K_p and store into the database. The updating is made every 3 hours.

3.2.2 Interface requirements

Requirement	Interface
Identifier	dbt.int.001
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.1
Name	Database.OMNI
Description	Database object for the OMNI 1-hour resolution solar wind data. The object shall contain the following fields: B, F, B_x, B_y-GSM, B_z-GSM, n, V, K_p, and D_{st}. It shall also contain the field units which are nT for the magnetic field data, cm^{-3} for the density, km/s for the velocity, and nT for D_{st}. K_p is dimensionless.

Requirement	Interface
Identifier	dbt.int.002
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.1
Name	Database.GOES.part
Description	Database object for the GOES 5-minute resolution particle data. The object shall contain the following fields: > 0.6 MeV, > 2 MeV, > 4 MeV electrons, > 1 MeV, > 5 MeV, > 10 MeV, > 30 MeV, > 50 MeV, > 60 MeV, > 100 MeV protons, and geopgraphic eastern longitude. It shall also contain the field units which are cm^{-2}s^{-1}sr^{-1} for the electron flux data and degrees east from central meridian for the longitude.

Requirement	Interface
Identifier	dbt.int.003
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.1
Name	Database.GOES.xray
Description	Database object for the GOES 5-minute resolution X-ray data. The object shall contain the following fields: 1-8 ÅX-rays, 0.5-4 ÅX-rays, and geopgraphic eastern longitude. It shall also contain the field units which are $\mathrm{Wcm}^{-2}\mathrm{s}^{-1}$ for the X-ray flux data and degrees east from central meridian for the longitude.

Requirement	Interface
Identifier	dbt.int.004
Need	Optional
Priority	Medium
Stability	Stable
Source	DB&T.CAP.1
Name	Database.LANL
Description	Database object for the LANL electron data.

Requirement	Interface
Identifier	dbt.int.005
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.2
Name	Database.ACE.mag
Description	Database object for the ACE 1-minute resolution solar wind magnetic field data. The object shall contain the following fields: B, B_x, B_y-GSM, and B_z-GSM. It shall also contain the field units which are nT.

Requirement	Interface
Identifier	dbt.int.006
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.2
Name	Database.ACE.swe
Description	Database object for the ACE 1-minute resolution solar wind plasma data. The object shall contain the following fields: n, and V. It shall also contain the field units which are cm^{-3} for the density and km/s for the velocity.

Requirement	Interface
Identifier	dbt.int.007
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.2
Name	Database.Dst
Description	Database object for the D_{st} 1-hour data. The object shall contain the following fields: D_{st}. It shall also contain the field units which is nT.

Requirement	Interface
Identifier	dbt.int.008
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.2
Name	Database.Kp
Description	Database object for the K_p 3-hour data. The object shall contain the following fields: K_p. It shall also contain the field units which is dimensionless.

3.2.3 Operational requirements

Requirement	Operational
Identifier	dbt.ope.001
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.14
Name	Database.inspectMeta
Description	A window to inspect the meta data for each database object in the database. All information in the meta data will be displayed.

Requirement	Operational
Identifier	dbt.ope.002
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.15
Name	Database.inspectData
Description	A window to inspect the data for each database object in the database. The data for a selected time period will be displayed.

3.3 saaps.saam

3.3.1 Functional requirements

Requirement	Functional
Identifier	saam.fun.001
Need	Essential
Priority	High
Stability	Stable
Source	SAAM.CAP.5
Name	SAAM.BL
Description	A function to calculate the position from cartesian (x,y,z) or spherical (r,θ,ϕ) coordinates to magnetic B-L coordinates. A tilted dipole is assumed.

Requirement	Functional
Identifier	saam.fun.002
Need	Essential
Priority	High
Stability	Stable
Source	SAAM.CAP.6
Name	SAAM.Interpolate
Description	A function to replace data gaps (NaN) in a time series object with linear interpolated values.

Requirement	Functional
Identifier	saam.fun.003
Need	Essential
Priority	High
Stability	Stable
Source	SAAM.CAP.7
Name	SAAM.Average
Description	A function to calculate the temporal average of a time series object.

Requirement	Functional
Identifier	saam.fun.004
Need	Essential
Priority	High
Stability	Stable
Source	SAAM.CAP.8
Name	SAAM.SEA
Description	A function to perform superposed epoch analysis on a time series object.

Requirement	Functional
Identifier	saam.fun.005
Need	Essential
Priority	High
Stability	Stable
Source	SAAM.CAP.9
Name	SAAM.Correlation
Description	A function to calculate the linear correlation between to parameters.

Requirement	Functional
Identifier	saam.fun.006
Need	Essential
Priority	High
Stability	Stable
Source	SAAM.CAP.11
Name	SAAM.PatternSearch
Description	A function to for patterns in a time series object. The pattern is a series of values that are matched against the values in the time series object. Each value in the pattern shall also have limits to be used in determining whether the time series value belongs to the pattern.

Requirement	Functional
Identifier	saam.fun.007
Need	Essential
Priority	High
Stability	Stable
Source	SAAM.CAP.13
Name	SAAM.EstimateBestModel
Description	From a time series object with anomaly events the best anomaly prediction model is found. Each available model are run for the times in the anomaly time series object and the model that gives the highest correlation is considered to be the best.

3.3.2 Interface requirements

Requirement	Interface
Identifier	saam.int.001
Need	Essential
Priority	High
Stability	Stable
Source	SAAM.CAP.1
Name	SAAM.DatabaseConnection
Description	The tools in SAAM shall operate on data from the SAAPS database. The data uploaded from the database are contained in a time series object.

3.3.3 Operational requirements

Requirement	Operational
Identifier	saam.ope.001
Need	Essential
Priority	High
Stability	Stable
Source	SAAM.CAP.10
Name	SAAM.ClusterAnalysis
Description	A user submitted list of events, e.g. anomalies, are plotted in a selected coordinate system. The coordinate system is chosen from the data or functions in SAAPS.

Requirement	Operational
Identifier	saam.ope.002
Need	Essential
Priority	High
Stability	Stable
Source	SAAM.CAP.12
Name	SAAM.GuideLines
Description	On each web page there shall be a link to guide lines and help pages to help the user in using the SAAPS tools.

Requirement	Operational
Identifier	saam.ope.003
Need	Essential
Priority	High
Stability	Stable
Source	SAAM.CAP.2
Name	SAAM.Plot
Description	Function to plot two dimensional data. The data are represented as time series objects. The scaling of the axis are automatic and also handles time data. The axes can have a linear or logarithmic scaling. There shall be a zoom capability.

Requirement	Operational
Identifier	saam.ope.004
Need	Essential
Priority	High
Stability	Stable
Source	SAAM.CAP.3
Name	SAAM.SelectData
Description	Function to select data from the database that shall be studied. It shall be possible to select data fields from several different database objects.

Requirement	Operational
Identifier	saam.ope.005
Need	Essential
Priority	High
Stability	Stable
Source	SAAM.CAP.4
Name	SAAM.SelectTimePeriod
Description	Function to select the time period over which the data shall be studied.

Requirement	Operational
Identifier	saam.ope.006
Need	Essential
Priority	High
Stability	Stable
Source	SAAM.CAP.8
Name	SAAM.SEA.Interface
Description	Web interface to perform superposed epoch analysis. The user must enter a list of events and select a parameter from the database to study. The result will be presented in a plot.

3.4 saaps.sapm

3.4.1 Functional requirements

Requirement	Functional
Identifier	sapm.fun.001
Need	Essential
Priority	High
Stability	Stable
Source	SAPM.CAP.1
Name	SAPM.Model
Description	Model for the prediction of satellite anomalies. The model contain a complete description of the database objects needed, how it should be preprocessed (normalization, time delays), and the algorithm producing the predictions.

Requirement	Functional
Identifier	sapm.fun.002
Need	Essential
Priority	High
Stability	Stable
Source	SAPM.CAP.8, SAPM.CAP.9
Name	SAPM.ModelLoad
Description	Load a prediction model from the model database.

Requirement	Functional
Identifier	sapm.fun.003
Need	Essential
Priority	High
Stability	Stable
Source	SAPM.CAP.2
Name	SAPM.ModelStore
Description	Store a new prediction model to the model database.

Requirement	Functional
Identifier	sapm.fun.004
Need	Non essential
Priority	Low
Stability	Stable
Source	SAPM.CAP.10
Name	SAPM.ModelTrain
Description	This function shall provide a tool to train a new neural network from data in the database. Due to the complexity to create the data set for training and validation it is open whether thsi function will be implemented.

3.4.2 Interface requirements

Requirement	Interface
Identifier	sapm.int.001
Need	Essential
Priority	High
Stability	Stable
Source	SAPM.CAP.3
Name	SAPM.ModelDB
Description	A database object that contain a prediction model.

Requirement	Interface
Identifier	sapm.int.002
Need	Essential
Priority	High
Stability	Stable
Source	SAPM.CAP.5
Name	SAPM.ModelSAPMInterface
Description	The models shall be accessible from SAAM. A request is made to run one model over a certain time period after which SAAM receives a time series object of the model output.

3.4.3 Operational requirements

Requirement	Operational
Identifier	sapm.ope.001
Need	Essential
Priority	High
Stability	Stable
Source	SAPM.CAP.4
Name	SAPM.Web
Description	The models in SAAPS shall be available from a web browser. The user can run the models in one of two modes: selected time period or real time.

Requirement	Operational
Identifier	sapm.ope.002
Need	Essential
Priority	High
Stability	Stable
Source	SAPM.CAP.6
Name	SAPM.WebPlot
Description	The result from a model run shall be presented as a plot. The user shall be able to control the scaling of the axes and the size of the plot.

Requirement	Operational
Identifier	sapm.ope.003
Need	Essential
Priority	High
Stability	Stable
Source	SAPM.CAP.7
Name	SAPM.WebList
Description	The result from a model run shall be presented as a list.

Requirement	Operational
Identifier	sapm.ope.004
Need	Essential
Priority	High
Stability	Stable
Source	SAPM.CAP.8
Name	SAPM.RealTime
Description	The models shall be available for real time operation and presented as plots. Each time new data are available the plots shall be updated.

Requirement	Operational
Identifier	sapm.ope.005
Need	Essential
Priority	High
Stability	Stable
Source	SAPM.CAP.9
Name	SAPM.AnyTime
Description	The models shall be available to run for a selected time period.

Requirement	Operational
Identifier	sapm.ope.006
Need	Essential
Priority	High
Stability	Stable
Source	SAPM.CAP.11
Name	SAPM.GuideLines
Description	On each web page there shall be a link to guide lines and help pages to help the user in using the SAAPS tools.

Chapter 4

User requirements vs. software requirements traceability matrix

4.1 Database and database tools, DB&T

Table 4.1: The relation between user requirements and software requirements. The left column indicates the DB&T.CAP.# number.

User requirements	Software requirements
1	dbt.int.001,002,003,004
2	dbt.fun.014,015,016,017,018,019 dbt.int.005,006,007,008
3	dbt.fun.010
4	dbt.fun.011
5	dbt.fun.012
6	dbt.fun.013
7	dbt.fun.001
8	dbt.fun.002
9	dbt.fun.009
10	dbt.fun.003,004,005
11	dbt.fun.006
12	dbt.fun.007
13	dbt.fun.008
14	dbt.ope.001
15	dbt.ope.002

4.2 Satellite anomaly analysis module, SAAM

Table 4.2: The relation between user requirements and software requirements. The left column indicates the SAAM.CAP.# number.

User requirements	Software requirements
1	saam.int.001
2	saam.ope.003, util.fun.002
3	saam.ope.004
4	saam.ope.005
5	saam.fun.001
6	saam.fun.002
7	saam.fun.003
8	saam.fun.004 saam.ope.006
9	saam.fun.005
10	saam.ope.001
11	saam.fun.006
12	saam.ope.002
13	saam.fun.007

4.3 Satellite anomaly prediction module, SAPM

Table 4.3: The relation between user requirements and software requirements. The left column indicates the SAPM.CAP.# number.

User requirements	Software requirements
1	sapm.fun.001
2	sapm.fun.003
3	sapm.int.001
4	sapm.ope.001
5	sapm.int.002
6	sapm.ope.002
7	sapm.ope.003, util.fun.002
8	sapm.ope.004
9	sapm.ope.005
10	sapm.fun.004
11	sapm.ope.006