SAAPS

Satellite Anomaly Analysis and Prediction System

Software Requirements Document

Version 0.3

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

P. Wintoft

3 April 2000

Document status sheet

SAAPS SRD	
Version	Date
0.1	12 December 1999
0.2	3 February 2000
0.3	3 April 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 Scope

1.3 Definitions, acronyms and abbreviations

1.4 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.

1.5 Overview

The structure of this document follows the document template as described in the ESA soft- ware 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

Lund Space Weather Model

2.2 Relation to predecessor and successor projects

Lund Space Weather Model and SPEE

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 Inter- net.

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

[WP 220]

2.3.3 Satellite Anomaly Prediction Module

[WP 320]

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.

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

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

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

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 handeled. 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 specifify which sampling that should be used. One way to acheive 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 exmaple 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.

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

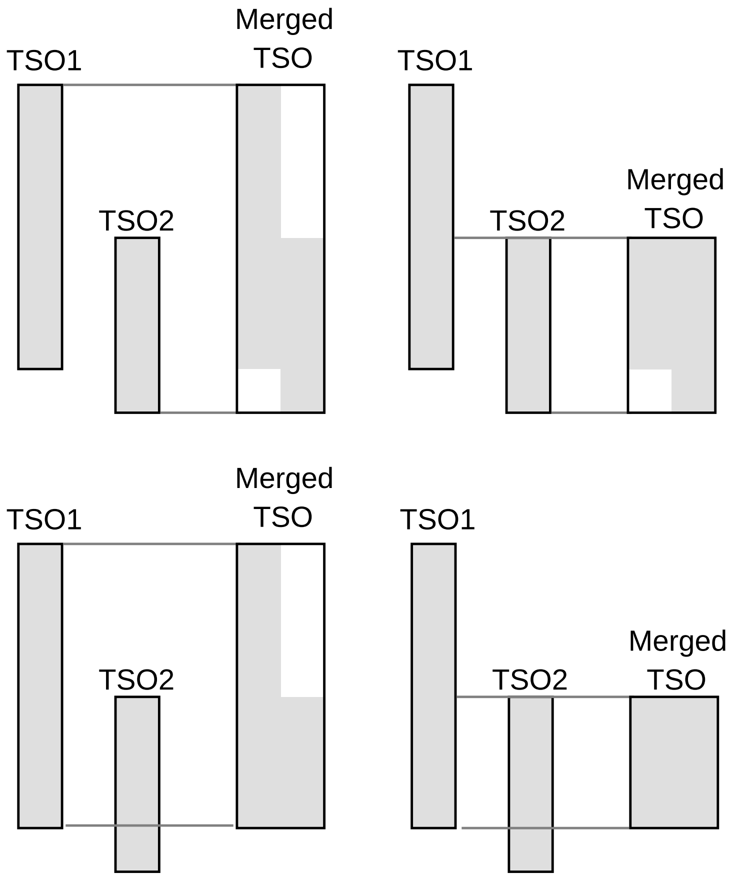

Figure 2.1: The merging of two TSOs. In this case the merging can be made in one of four possible ways. The shaded areas represents the data and the white areas NaNs.

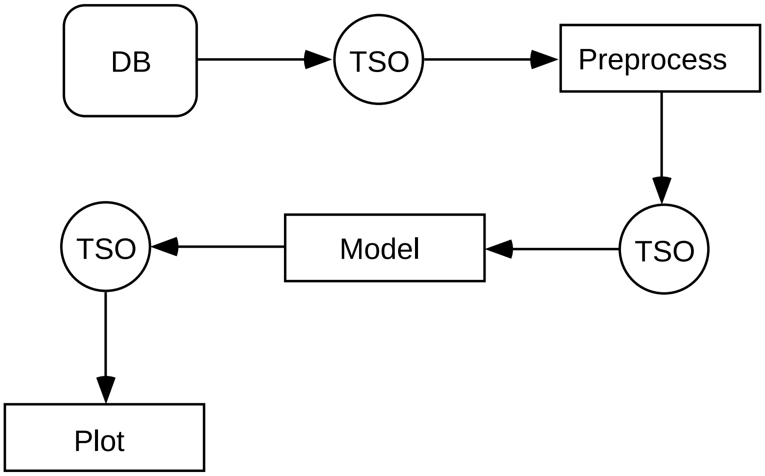

Figure 2.2: The flow of data from the database (DB) using a time series object (TSO) that is preprocessed and run in a model to produce a plot.

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 desribed 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 can not be set.

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 filesystem.

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.

When a DBO is created the first time the MDBO is also created. Then, the DBO can be updated from TSOs. 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 $7 \times 8 \times 60 \times 24 = 80640$ bytes per day (79 kB/day) or 29 MB/year.

Data can now be stored in the database object. 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.

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.

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 [*Wintoft,* 2000a]).

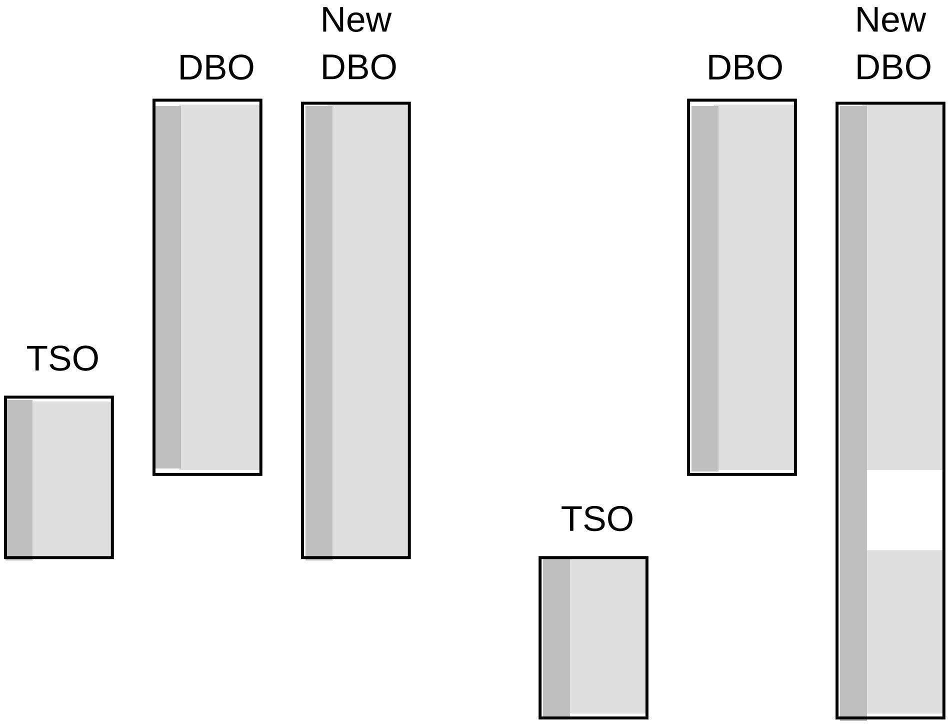

Figure 2.3: The storing of a TSO to the database. The time field is illustrated with the horisontal hatched area and the data fields by the diagonally hatched area. Missing data (NaNs) are left blank.

Table 2.3: Parameters needed for retreiving 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 retreive 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.

Retreiving resampled data is different from retreiving data followed by a resample; retreiving 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 retreive 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 retreiving 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 Retreiving 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 retreiving resampled data together with a list of database objects and the respective data fields. The parameters are summarized in Table 2.5.

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.

Table 2.5: Parameters needed for retreiving 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

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 invloves 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 [?].

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 [?] shall be included in the automatic update.

2.4.3 Satellite Anomaly Analysis Module, SAAM

All the functions described below 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 SAPM plotter will be a faster way to acheive 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 [*Wintoft,* 2000a].

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 [*Wintoft,* 2000a] 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 [*Wintoft,* 2000a] 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).

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 rigth 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 desrired time

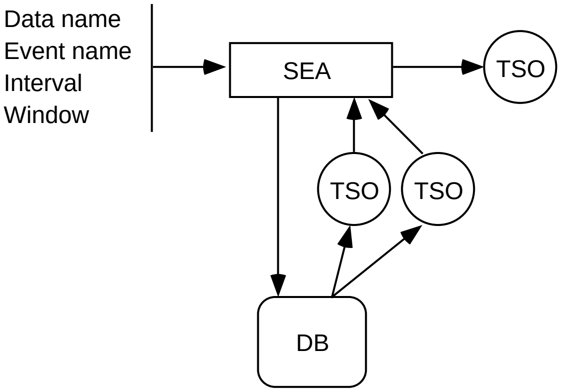

Figure 2.4: Superposed epoch analysis.

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

2.4.3.8 Pattern search

2.4.3.9 Guidelines

2.4.3.10 Estimate best model

2.4.3.11 Graphical interface

2.4.4 Satellite Anomaly Prediction Module, SAPM

2.4.4.1 Prediction model

There are many steps going from the database to produce a prediction. The prediction model will contain all the necessary functions to achieve this.

The model must contain a description of the DBOs and data fields that are needed, what the time resolutions are, if data gaps should be interpolated, and the normalization of the input. From this information a TSO can be produced.

The model must also contain the information of how the TSO should be transformed into a matrix that will be the actual input matrix to the neural network. The transformation describes which data fields goes to which input of the network and if there are any time delays. Usually it is only the data fields from the TSO that are used as the input. However, some model also need the time information explicitly as input. This is done by taking the sine and cosine of a normalized time. The normalized time could e.g. be the local time transformed over the interval 0 to 2π or the day of the year transformed to 0 to 2π.

The neural network must also be described. The network is defined by the number of layers, the weights connecting successive layers, and the transfer functions for each layer.

When the input matrix is run through the network it will produce an output matrix. This contains no time information but this must be added afterwards. The prediction times are a function of the times in the input TSO. The model shall thus add this time to the output and produce an output TSO.

Finally, the output TSO should be de-normalized to produce a TSO that contain the actual output values.

The prediction model shall contain all information needed to retreive the data from the database, perform preprocessing, run the neural network, and finally to perform postprocessing. A specific prediction model is charaterised by the items in Table 2.6.

Table 2.6: The prediction model parameters

dataFields	Database data fields
sampleInterval	Sample interval of the data
normPar	Normalization parameters
inputTransform	Description of transformation from TSO to the network input
neuralNetwork	Neural network weights, biases, and transfer fuc-ntions
predTime	Prediction time
outputTransform	Transformation from network output to a TSO

It shall be possible to store and retreive existing models to/from file. A new model can also be created from data stored as binary or ASCII files in a predefined format. The files must contain the model parameters.

To run a prediction model one simply needs the start and stop times over which the predictions shall be made. The model then retreives the data over the selected time period and returns a TSO with the predicted values.

2.4.4.2 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. A new network is thus created by giving the above information. This will lead to the creation of the network weight matrices and bias vectors containing random numbers. To obtain a useful network it must first be trained. A network can also be created by giving a set of weight matrices, bias vectors, and transfer functions, where the weights and biases are obtained from a previously trained network.

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

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

and

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

The number of inputs for layer l equals n_l and the number of neurons, which is the same as the number of outputs from the layer, equals m_l.

To run the neural network input data must be given. The form of the input matrix is

$$\mathbf{X} = \begin{bmatrix} x(1,1) & x(1,2) & \dots & x(1,q) \\ x(2,1) & x(2,2) & \dots & x(2,q) \\ \vdots & \vdots & \ddots & \vdots \\ x(n_1,1) & x(n_1,2) & \dots & x(n_1,q) \end{bmatrix}, \tag{2.3}$$

where n_1 is the number of inputs and q is the number of examples.

The network input data are generated from a time series object. How the time series are transformed is described in the network parameter object. The network parameter object contains how the data in the time series shall be normalized and if it has time delays.

2.4.4.3 Preprocessing time series objects for neural network

Before a neural network can be run the time series objects must be preprocessed. The steps are: 1) extract the data fields from one or several time series objects, 2) normalize each data field, and 3) reformat data into an input data matrix with possible time delays. This information resides in a network parameter object which also contain the weight matrices and bias vectors. The network parameter contain the fields shown in Table 2.7.

2.4.4.4 Prediction model

To run a prediction model two parameters are needed: the model name and the time interval over which the prediction is to be made. Based on the

Table 2.7: The network parameter fields

Name	Description
WM(i)	Weight matrix for layer i
BV(i)	Bias vector for layer i
TF(i)	Transfer function for layer i
TSO(j)	Time series object j
DF(j)	Data fields from TSO(j)
TDL	Time delay line description

model name the model definition (MD) is loaded. The MD contains which data should be used and various parameters needed for the normalization, pre-processing, and post-processing. The output from the model is a TSO containing the predicted values. Figure 2.5 shows the flow of data in the model.

The first step is to merge the data from possible several TSOs. The TSOs are averaged and resampled to match each other. Then the relevant data fields are extracted and a new TSO is created.

The TSO is then normalized so that all input values lie in the correct range for the network.

The data fields in the TSO are extracted (PreNet) to create the input matrix for the network. This step also includes possible rearrangement of the matrix elements to create a time delay line.

The input matrix is run through the neural network (NN) which generates the output matrix.

Finally, the output matrix is inserted into the merged TSO (PostNet) to generate the model output TSO. The data fields available in the output TSO is thus both the input data and the predicted data. The predicted values are the time stamps that corresponds to the time of the input data plus the prediction time.

2.4.4.5 On-line manuals

2.4.4.6 Graphical interface

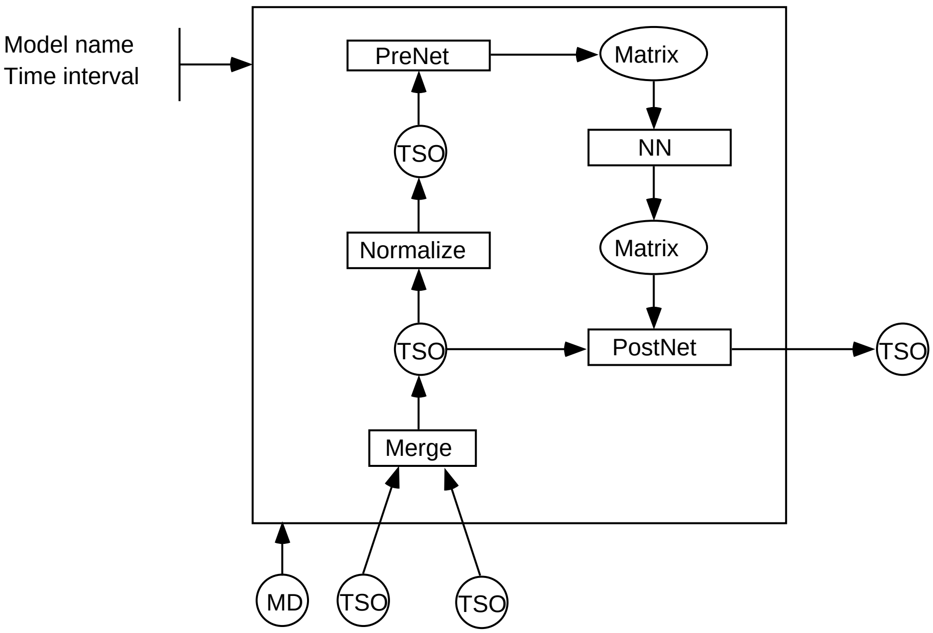

Figure 2.5: The prediction model. Inputs are model name and time interval. Then the model definition (MD) and time series objects (TSO) are loaded. The result is a new TSO.

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 retreive data in real time. The user interacts with SAAPS via the SAAM and SAPM subsystems. However, on the developer level we introduce two other subsystems that contain general software that are used manily by SAAM and SAPM, namely, UTIL and ANN. UTIL is the utility subsystem and ANN is the artificial neural network 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 excpected that this functions will be same or not over the project development.

Source Reference to the URD specific requirement.

Description A short and consice description of the function.

3.1 saaps.dbt

Requirement	Functional
Identifier	fun.dbt.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	fun.dbt.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	fun.dbt.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	fun.dbt.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	fun.dbt.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	fun.dbt.006
Need	Essential
Priority	High
Stability	Stable
Source	DB&T.CAP.9
Name	Metadata.retreive
Description	Retreive the meta data for a specific database object.

3.2 saaps.saam

3.3 saaps.sapm

Requirement	Functional
Identifier	fun.sapm.001
Need	Essential
Priority	High
Stability	Stable
Source	SAPM.CAP.1
Name	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	fun.sapm.002
Need	Essential
Priority	High
Stability	Stable
Source	SAPM.CAP.2
Name	Model.store
Description	Store a model to the hard disk.

Chapter 4

User requirements vs. software requirements traceability matrix

References

Andersson, L., L. Eliasson, and P. Wintoft, Prediction of times with increased risk of internal charging on spacecraft, *Workshop on space weather,* ESTEC, Noordwijk, WPP-155, 427–430, 1999.

Boberg, F., P. WInfoft, and H. Lundstedt, Real time Kp predictions from solar wind data using neural networks, To be published in *Physics and Chemistry of Earth*, 1999.

Baum, E.B., and D. Haussler, What size net gives valid generalization?, *Neural Computation 1,* 151–160, 1989.

Cybenko, G., Approximation by superpositions of a sigmoidal function, *Math. Control Signals Systems* 2, 303–314, 1989.

Hagan, M.T., and M. Menhaj, Training feedforward networks with the Marquardt algorithm, *IEEE Trans. Neural Netwworks, 5*, 989–993, 1994.

Haykin, S., *Neural Networks: A Comprehensive Foundation*, Macmillan College Publishing Company, Inc., New York, 1994.

Jacobs, R.A., Increased rates of convergence through learning rate adaption, *Neural Networks* 1, 295–307, 1988.

Freeman, J.W., T.P. O'Brien, A.A. Chan, and R.A. Wolf, Energetic electrons at geostationary orbit during the November 3-4, 1993 storm: Spatial/temporal morphology, characterization by a power law spectrum and, representation by an artificial neural network, *J. Geophys. Res., 103,* 26,251–26,260, 1998.

Koons, H.C., and D.J. Gorney, A neural network model of the relativistic electron flux at geosynchronous orbit, *J. Geophys. Res., 96,* 5,549–5,556, 1991.

Mayaud, P.N., *Derivation, Meaning, and Use of Geomagnetic Indices*, AGU Geophys. Monograph 22, Washington, D.C., 1980.

Nguyen, D., and B. Widrow, Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights, in *Proc. IJCNN,* 3, 21–26, 1990.

Rodgers, D.J., Correlation of Meteosat-3 anomalies with data from the spacecraft environment monitor, *Internal ESTEC Working Paper No. 1620*, ESTEC, Noordwijk, 1991.

Stringer, G.A., and R.L. McPherron, Neural networks and predictions of day-ahead relativistic electrons at geosynchronous orbit, *Proceedings of the International Workshop on Artificial Intelligence Applications in Solar-Terrestrial Physics, Lund, Sweden, 22-24 September 1993,* Edited by J.A. Joselyn, H. Lundstedt, and J. Trolinger, Boulder, Colorado, 139–143, 1993.

Stringer, G.A., I. Heuten, C. Salazar, and B. Stokes, Artificial neural network (ANN) forecasting of energetic electrons at geosynchronous orbit, in

Radiation Belts: Models and Standards, Geophys. Monogr. Ser., vol 97, edited by J.F. Lemaire, D. Heynderickx, and D.N. Baker, AGU, Washington, D.C., 291–295, 1996.

Swingler, K., *Applying neural networks: a practical guide*, Academic Press Ltd, London, 1996.

Wintoft, P., Satellite Anomaly Analysis Module, *SAAPS, Technical Note 2*, 2000.

Wintoft, P., Satellite Anomaly Prediction Module, *SAAPS, Technical Note 3*, 2000.

Wrenn, G.L., and R.J.K. Smith, The ESD threat to GEO satellites: empirical models for observed effects due to both surface and internal charging, *ESA Symposium Proceedings on Environment Modelling for Space-based applications,* ESTEC, Noordwijk, SP-392, 121-124, 1996.

Wu, J.-G., H. Lundstedt, L. Andersson, L. Eliasson, and O. Norberg, Spacecraft anomaly forecasting using non-local environment data, *Study of plasma and energetic electron environment and effects (SPEE),* ESTEC Technical Note, WP 220, 1998.