CHALMERS

SAP Notifications Data Model

Automatic Reporting of Maintenance and Defect Data from Electrical Power Grids

Alexander E. Jansson
Johan C. Andersson

ISSN 1401-6184

INSTITUTIONEN FÖR ELTEKNIK
Examensarbete 92E
2004

Titel
SAP Notifications Data Modell
Automatisk Rapportering av Underhålls- samt Avbrottsdata från Elnät

Title in english
SAP Notifications Data Model
Automatic Reporting of Maintenance and
Defect Data from Electrical Power Grids

Författare/Author
Alexander E. Jansson
Johan C. Andersson

Utgivare/Publisher
Chalmers Tekniska Högskola
Institutionen för elteknik
412 96 Göteborg, Sverige

ISSN
1401-6184

Examensarbete/M.Sc. Thesis No.
92E

Ämne/Subject
Elsystem

Examinator/Examiner
Jaap Daalder

Datum/Date
2004-02-11

Tryckt av/Printed by
Chalmers tekniska högskola
412 96 GÖTEBORG

Abstract

The changed conditions on the electricity market due to deregulation have created a need for smarter maintenance and asset monitoring. This is to supply the customers with high quality energy and also to reduce costs for utilities. A step in the direction to please these needs is to go from time based maintenance to condition based maintenance. A fundamental condition for Powerlink taking this step is to make use of all data collected from the electrical power grid stored within Plant Maintenance Repository (*SAP*). By using the data, the strategy engineers can perform a more effective work by checking the condition of assets and trace trends. They will receive feedback on the maintenance being performed on objects in the electrical power grid.

By removing the manual steps in the process, in going from the collection of data to putting it into reports, the transition to condition based maintenance will be facilitated. With the software application package provided by *Hummingbird BI* it is possible to automatically collect data from the Plant Maintenance Data Repository on a daily basis and raise exceptions for deviating data exceeding certain thresholds. The data is put into web based reports in a data model called *SAP Notifications* and the values exceeding the thresholds are highlighted to make the user aware of feasible problems. Once every quarter, the numbers of exceptions for each technical object type and report are send out to the strategy engineers via e-mail. This is a scheduled process and it highlights areas where additional attention is needed and also serves as a reminder for the strategy engineers to make use of the data in the reports.

SAP Notifications introduces a new way of making use of data stored within the database. There are big advantages with this new method for extracting and presenting data compared with the old method. Time is saved and the risk of introducing errors in the data is minimised, both parts are essential for effective maintenance and asset monitoring. A possible concern with the new technology is that reliance on people’s instinct and knowledge of the electrical power grid and its components may be reduced. Instead a total acceptance of statistical analysis of the raw data may be introduced.

For the moment, this is the only method for automatic updating of reports and sending of e-mails to the users. The way the reports can be created upon request from the users is also a feature that makes the model a powerful tool for the improvement of maintenance procedures. However, when other maintenance software packages catch up with SAP Notifications it is important to investigate if the model is the most suitable tool for Powerlink.

Table of Contents

List of Illustrations and Diagrams

List of Tables

Nomenclature

ACMS	-	Asset Condition Monitoring System
AMT	-	Asset Monitoring Team
CMMS	-	Computerised Maintenance Management Systems
EMS	-	Energy Management System
FMECA	-	Failure Mode Effects and Critically Analysis
FOD	-	Forced Outage Database
KPI	-	Key Performance Indicator
LCC	-	Life Cycling Costing
OLAP	-	On-Line Analytical Process
OSTRAC	-	Outage System – Transmission Reporting And Co-ordination
RCM	-	Reliability Centred Maintenance
SAP	-	Systems, Applications & Products in Data Processing
SVC	-	Static Var Compensator

Acknowledgments

We would hereby wish to acknowledge the persons that have helped and supported us before and during the work with our Master Thesis Work at Powerlink Queensland. Without the support, the expedition to the other side of the world would not have been possible, neither would it have been possible to perform the work in a successful manner.

A number of persons have contributed with support in different ways, but our grateful thanks are foremost dedicated to Dr. Stewart Bell and Mr. Andrew Bannister, both at Powerlink Queensland, for their uninterrupted, positive and inspiring service during the work. Dr. Bell has managed to combine professional leadership and academic supervision. His innovative ideas have put us in the right direction when the path has been undefined. We are thankful for the patience shown by Mr. Bannister in answering our questions, and his knowledge has served as an important source for writing of this thesis. Both gentlemen have also made contributions on a social level outside work, making our stay in Australia very pleasant.

We would also like to forward our grateful thanks to the following persons and organisations,

Professor Jaap Daalder	Academic Supervisor, Chalmers University of Technology, Gothenburg
Ph.D Richard Thomas	Chalmers University of Technology, Gothenburg
Mr. Brian Pokarier	Powerlink Queensland
Mr. Patrick Wilson	Powerlink Queensland
Mr. Garry Hoey	Powerlink Queensland
Mr. Brian Sharp	Powerlink Queensland
AMT staff	Powerlink Queensland
Powerlink Queensland	Brisbane, Australia
AB Ångpanneföreningen	Sweden
Merlin Gerins Stipendiefond	Sweden

________________ ________________

Alexander E. Jansson Johan C. Andersson

Date: ______________

1 Introduction

Power quality has become extremely important in modern society due to increased quantities of intelligent electronic equipment. A consequence of this is that smarter maintenance has to be performed by the transmission and distribution companies to be able to supply the customers with high quality energy. Following deregulation, when generation and distribution became separate entities, the transmission companies have had a stronger focus on cost reduction. This, together with the fact that penalties for low power quality are starting to be introduced, means the need for proper preventive maintenance has never been so important.

The increasing effort in improving maintenance is also due to the fact that as much as 25% of today's infrastructure is regarded as old and has to be replaced within 10 years (Gellings, 2002). Old equipment in general means more failures, for example due to wear out, which in turn means that the amount of maintenance of the equipment has to be increased to ensure a satisfactory level of performance and reliability.

However, this causes problems for the utilities since the cost of maintenance, in most cases, increases with increased maintenance. A more effective preventive maintenance system needs to be found in order to reduce the costs. This can be achieved by using maintenance and defect data in a more effective manner. There are a number of strategies in use by utilities to achieve more effective preventive maintenance. One of them is to take the step from manually interrogating data for the production of reports to automatically interrogating data for the production of automatic reports.

Removing the manual steps in creating graphs and presentations is a significant enabler to being informed about your network assets, which is essential when going from time based maintenance to condition based maintenance. The idea behind condition based maintenance is to perform maintenance when and where it is needed based on the equipment's condition and not as before, scheduled to fixed time intervals. (Gellings, 2002)

At Powerlink, a large amount of data is collected from the electrical power grid by the Energy Management System (EMS). The gathered data can be viewed live, but is also stored within the EMS for seven days to be used later on. To ensure this data is available for historical purposes, the Asset Condition Monitoring System (ACMS) interrogates the EMS everyday. The ACMS is the long-term storage system for electrical power grid data. The data stored within the ACMS is an averaged one, five and 30-minute data for all different plant types on the system such as transformers, SVCs, capacitor banks etcetera. Examples of data found might be a tap changer position as a function of time, power transferred through a feeder, relay status etcetera. The ACMS also contains data entered by the maintenance service provider such as plant breakdown or maintenance data, which is obtained from SAP (Systems, Applications & Products in Data

Processing). Below is an overview of the process when gathering data from the electrical power grid. After a couple of steps the data is populated in the tables used in this project.

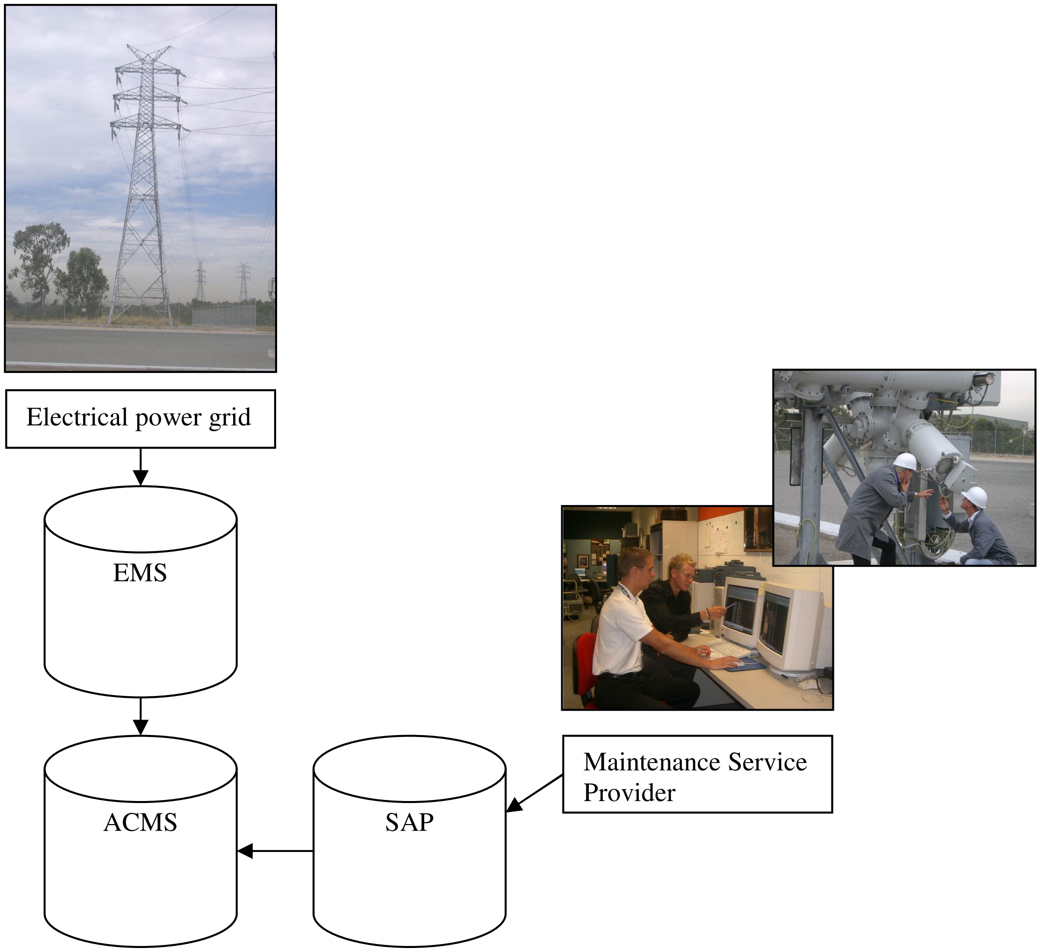


Figure 1-1 Overview of some of the electrical power grid data flow in Powerlink

(Haak, 2003; Hoey, 2003)

It has been shown that to be a leader in asset management (AM), life cycle cost (LCC) analysis and good knowledge about the assets conditions are essential ingredients. Another key in asset management is to have the information easily accessible (Moller, Becconsall, 1998). It is in the automatic generation of information on the maintenance and defect data in Figure 1-1 that this work focuses on.

1.1 History of Maintenance and Asset Monitoring

Maintenance and asset monitoring, generally in industry, has faced several major changes during history. One of the biggest causes of change is probably the invention of the computer. It introduced the possibility of storing large amounts of data in a relatively accessible manner. In the case of a transmission utility, data is collected from the electrical power grid in an efficient manner. The computer also made it possible to create visual aids like graphs to serve as a helpful tool for engineers.

To be able to improve asset monitoring and maintenance it is important to be aware of how they were performed in the past. For the last century until today the maintenance can be summarised according to Table 1-1. (Sharp, 2003)

Table 1-1 The development of maintenance through history

Stage	First half of 20th century and earlier	Immediate Post WWII	After WWII
Requirements	Keep plant running	Reduce production loss	Optimise costs and quality
Maintenance	Fix it when it breaks	Time based preventive	Condition monitoring, concept reliability centred maintenance (RCM), Life cycle costing
Work Control	Stand-by for unplanned jobs	Program preventive, and stand-by corrective	Full maintenance planning with computerised maintenance system

As seen in the table above, the need for maintenance has been increasing since the last century, which has caused the amount of asset monitoring to increase. The different time intervals in the table above will be examined further in the following paragraphs.

First half of 20th century and earlier

Before the twentieth century mass production was not very common, and many products were produced by hand. This, together with the fact that most technical systems were simple but over-designed, resulted in low number of failures on the equipment. The need for maintenance management was limited, but proper maintenance was of course still important. When the equipment approached its end of life, most of the failures turned up as wear out and was quite easy to predict, which reduced the need for proper monitoring. The monitoring itself was included in the maintenance, performed when an engineer was on site and for instance inspecting and oiling.

After the turn of the century, a couple of changes developed. The mix of the systems used became more complex and the evolution of trade based workforce for craftsmen ended up with a split between operation and maintenance. However, maintenance management was still rare and the philosophy was often "operate-to-failure and repair" (Kelly, 1989).

Immediate Post WWII

The size and complexity of the systems continued to increase during the first half of the twentieth century, and after World War II the systems became more vulnerable for line stoppages. Therefore, the concept of *preventive maintenance* turned up as a way to prevent stoppages, and with that minimise loss of production and loss of profit. The definition of preventive maintenance was "to find and correct any condition that may cause machine failure before such a failure occurs" (Kelly, 1989). Compared to the actions performed today, the maintenance was time based instead of condition based since most failures still were considered to be wearing out failures.

After WWII

A couple of decades after the Second World War, actions were taken to optimise scheduled preventive maintenance and with that achieve cost and quality optimisation. Also, during this period the concept of Reliability Centred Maintenance (RCM) was introduced as a result of studies by the Federal Aviation Agency (FAA) of the USA. FAA still maintains records of maintenance on commercial airlines used to alert authorities and manufacturers and also used when crashes are investigated. The study the FAA conducted showed that:

- scheduled overhaul has limited or little effect on complex items (unless the items have a dominant wear out failure mode)
- there exists items for which there is no effective form of preventive maintenance

(Kelly, 1989; Levitt, 2003)

A decade later, in the 1970's, sophisticated condition monitoring came into practice within the industry. The monitoring was seen as "one of a number of alternative maintenance actions" (Kelly, 1989) and had the objectives defined by the following points:

- minimise the risk of unintentional breakdown
- maintain a certain level of reliability by predicting the amount of maintenance required
- run diagnosis to determine the state of health of items and decide how much work required to maintain the item at a certain operating level.

(Kelly, 1989)

Another concept, LCC, was also introduced under this period. LCC is a method to include all costs associated with the total life cycle of an item, and is used for replacement issues. That is, also maintenance issues as well as plant design studies. In a graph for LCC the cost is plotted cumulatively or per unit time versus the age, and at the age where a straight line from the origin touch the curve is called *the Economic Life*. That age represents the time when it is most economical to replace the equipment. (Hastings, 2000)

The new era of monitoring was strengthened by the implementation of computers into the maintenance work in the late 1960's. Different software and computer systems for more efficient preventive maintenance were introduced onto the market, which supported storage of data (entered by tradesmen from a simple fault data sheet) and some analysis. Some of the systems had features for showing the top ten items, with longest down time and labour cost etcetera. Developments by the end of the 1980's provided the maintenance and monitoring area with condition monitoring modules and features to efficiently aid engineers when working with the replacement of equipment and components. (Gellings, 2002; Sharp, 2002; Kelly, 1989; Levitt, 2003)

1.2 Current Trends in Maintenance and Asset Monitoring

The previous section discussed the history of asset monitoring till the 1990's. This section will focus on contemporary and future trends. In general the major trends are on-line monitoring, web-based user interface, walk around monitoring that eventually leads to fully automatic monitoring, condition based maintenance and sensors that trend the data. The following paragraphs will briefly describe these trends.

<u>On-line monitoring</u>

The on-line monitoring process continuously performs measurements on the equipment and updates the database with the newest data available, and by doing this reduces the chance of missing problems. This process has mainly been used for expensive or essential assets, and also in dangerous or inaccessible areas. Development of modern electronics has made on-line systems more common and nowadays even other equipment than the most essential is cost-effective to supervise. An example of this is the company Trendmaster, which is delivering systems for supervision of less critical assets. These systems will reduce the probability that a possible cause of failure is missed. An example of where this system is used is for transformer monitoring by ElectraNet, Australia. (Krieg, 2001;Trendmaster Pro, 2003)

<u>Web-based user interface</u>

The collected data, either from on-line systems or discrete sampling, should be easy to use otherwise it is of limited use. Also the data should be fed back to the people who gather it otherwise they might think it is a waste of time since it is not used. This can easily be done using a web-based user interface, which all employees can have access to. A distributed system will be more flexible and efficient and also gives the user more freedom with the data. Technically this can be done with an HTML page, XML or as in this project, indirectly with java via the Business Intelligence software used at Powerlink. (Plant Monitoring, 2003; Thurston, 2001)

<u>The others</u>

The overall trend in asset monitoring adopted by utilities is to automate the whole process. The automated process should also monitor the condition of assets to enable condition-based maintenance, which is becoming more common. The modern sensors trend data to discover changes in the assets condition rather than to perform complicated exact measurements. This often allows for more simple measurements that can be made on-line and automatically, and an indication can be given before a failure occurs, but exact measurement is still needed. There is also a movement towards saving the waveforms for the different parameters so that after a fault has occurred, full analysis of the cause can be conducted.

1.3 Drivers for Maintenance and Asset Monitoring

The core reason for asset monitoring is to, via measurement and analysis of data, achieve a better knowledge of the system and how it performs, identify trends helping strategy engineers to decide when to do maintenance and replacement. "The only reason we conduct maintenance is that we perceive it will cost us more if we don't do it" (Childs, 2001), and hence it is obvious that the main reason to improve maintenance is to be able to cut cost a little bit more. The reduced cost is achieved by using the information received from measurements to improve maintenance strategies. Part of the improvement phase is to use the information to check the efficiency of maintenance performed, to "close the loop" (Bannister, 2003). It is important to check the maintenance efficiency to ensure items in the electrical power grid operate without failing, prevent deterioration due to corrosion and keep the plant safe. Another advantage of asset monitoring is that the step of going from time based- to condition based preventive maintenance becomes easier since the condition can easily be accessed if asset monitoring is performed.

As seen discussed above, History of Maintenance and Asset Monitoring, sophisticated asset monitoring came into history first in the late 1970's. The driver for more efficient monitoring was the need for optimisation of cost and quality of maintenance since systems had become more and more complex.

In the last couple of years the driver has been of a more political type. It is well known that deregulation has been introduced in a great number of countries, but it may not be known that this has had a big influence on asset monitoring and maintenance. Transmission utilities have become more profit focused and they also operate under the risk of penalties if providing customers with low quality power. To maximise profit it is essential that the plant can operate for a long time without failures.

1.4 Risks of Applying New Technology to Maintenance and Asset Monitoring

The way asset monitoring is performed today, engineers' gut feeling and knowledge about the electrical power grid plays an important role for the maintenance and monitoring. That is, people's knowledge serves as trigger for the start of an investigation. However, when introducing new technology no history exists so more trust is put into the raw data. Using only raw data for analysis is the big risk of applying new technologies. An error in the data might not be discovered which for instance could end up with improper maintenance being performed.

2 Industry Practice

To clarify why the model in this project is so important and what the advantages and disadvantages for Powerlink are compared to the currently used method, this chapter contains discussion about the latest work performed within the asset monitoring team (AMT) considering asset monitoring. It also includes some aspects about the available software for asset monitoring and preventive maintenance that can be found on the market today, and the software that has served as tools to create the model in this project.

2.1 Powerlink

Today, the data used for analysis is retrieved predominantly from the ACMS or SAP. In the first case there is a lack of information that can be displayed, not all details are available, and to be able to increase the amount of information a great deal of time is required. Regarding SAP, the way to extract data today is to locate the right tables and interrogate them, for example via BI Query (SAP data via the backdoor), or to perform a query within SAP.

For the staff within the AMT at Powerlink today, a specific analysis of an object using SAP requires great effort and is quite time consuming. This is the greatest disadvantage since time for other important tasks is reduced. Reports are created for specific items of plant upon request from different engineers. The method, going from the request of an investigation of an object to the analysis of the data and generation of reports is not automatic in any way, except maybe the fact that there in some cases exist spreadsheets from earlier investigations. Also the request for an investigation relies on the knowledge and gut feeling of engineers that something needs to be investigated instead of being automated, which would decrease the probability that anomalies are missed. For instance if they notice that a certain circuit breaker has failed many times compared to others within the same area, they might want to follow this up and contact the AMT crew and ask for an investigation about the specific breaker. This is the triggering of the manual analysis process, which today takes about two to three days and follows the pattern in Figure 2-1.

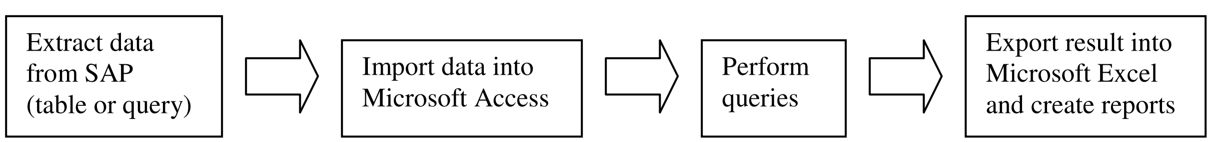


Figure 2-1 Flow chart of how analysis work is done at Powerlink

However, the analysis contains conclusions about the item in question together with recommendations on how to conduct further investigations, which also is a part in following up the result created by the automatic model. (Bannister, 2003)

2.1.1 Work Already Performed

When it comes to work already performed within the area of automatic generation of reports and exception raising in other parts of the electrical power industry, no other system with the same functions is, at the date of writing this thesis, known by the authors. Instead, Powerlink serves as the trendsetter by the creation of this data model.

Within Powerlink there has been effort to automate the work. For example, Mr. Andrew Bannister has automated the analysis of Forced Outages Database (FOD) data, but not to the same extent as for the notifications data for this project. For FODs the process sends out a reminder e-mail every day, that tells if there are some FODs still open after 24 hours or if the closure time (end time) is zero, the last information is important to make sure the data is correct. The automatic process with FODs is first to interrogate the database, process the data retrieved and then send out the e-mail.

There are other processes that also have been automated, but they work the similar way as the FOD reports. (Bannister, 2003)

2.2 Maintenance Software Packages

Efficient preventive maintenance has not been fully realised until recent years, therefore the right tools for transmission utilities to use are still not well known. From the earlier chapters it is known that the need for better diagnosis and analysis methods are increasing, and with this the authors assume that the number of relevant software packages will increase. However, a few prominent packages exist today, and these are briefly presented below. To further understand what advantages and disadvantages are involved with the analysis method in this project, some of the present industry practice is presented.

Maintenance software packages are used within many different business areas to present valuable or new information from old data, focusing on maintenance information. The software packages are useful for companies that need better cost control, have bad reliability, check if their maintenance strategies work or just want to verify that they are performing well.

Valuable information that the maintenance software can give is for example, when it is most economic to replace equipment, or how long before a failure is likely to occur. All of this helps companies to increase their profit by decreasing the costs, tweak their performance, achieving better reliability and availability and also to empower the employees since the data can be presented easily in an understandable way.

The different softwares on the market are equal from the point of view that they all for instance support a way to keep records of each piece of equipment, schedule preventive maintenance and make decisions about cost-effective replacement and repair.

Since the preventive maintenance area is rather new, information about relevant software is rather limited. The software packages presented below are the most popular ones.

2.2.1 MAXIMO

One of the most used applications for maintenance and asset monitoring today is MAXIMO, it was developed by the company MRO Software. MAXIMO also serves as an on-line analytical processing (OLAP) tool, which allows the users to drill up and down in the data for further information. Standard reports for equipment are for instance:

- Equipment failure summary by machine/equipment
- Maintenance cost by equipment
- Failure analysis graphs

The analysis part of the software enables the user to view the data at different levels (often referred to as "slice and dice"). For instance it is possible to look at a certain parameter for the total population of one technical object type, and thereafter drill down to study the data for one specific piece of equipment.

MAXIMO also supports ad hoc querying and interactive reporting, which allows the user to put up their own queries in order to create reports containing any kind of data from the database.
(MAXIMO 4I, 2003)

2.2.2 MainMan

MainMan is an Oracle based software application developed by the company KEMA. It is basically a generic tool for reducing maintenance costs and avoids failures due to aging etcetera. The following features are available with the MainMan software application:

- Smooth change from time base maintenance to condition based maintenance.
- Runs Failure Mode Effects and Critically Analysis (FMECA), analyses the loss of system functions due to the failure of an object, to decide which amount of maintenance that is required and indicates how soon the maintenance is required.
- Diagnosis, optimisation of maintenance interval, inspection and trend analysis
- Provides technical alarm status and economical interval optimisation.
- Reporting of for instance component analysis customised for each company individually

(Ackerman, Smit, 1997; MainMan, 2003)

2.2.3 CIMPLICITY EAM

The CIMPLICITY EAM software application is developed by Datastream and GE Fanuc. CIMPLICITY EAM provides tools for generation of work orders and a number of maintenance and asset management tasks. The software application works a little bit differently compared to the model developed within this project and the other software applications presented in the two previous paragraphs. It does not generate any reports, instead it uses key performance indicators (KPIs) implemented in a dashboard-like gauge interface. (Datastream, 2003)

2.2.4 SAP (Systems, Applications & Products in Data Processing)

The SAP system was introduced at Powerlink in 1999 and provides a complete business solution, everything from payment to maintenance management, for business operation.

SAP can be interrogated both from the front end or the back end. The front end serves as the interface towards the ordinary users and the engineers searching for and also filling in fault data, while the back end involves direct access to the database tables. For the actual project, both SAP interfaces have been used. The back end for the tables building up the core of the project, the data model (see section 3.5), while the front end merely has served as a way to verify the results achieved by using the data model. (Wharf, 2003; The SAP fan club, 2003)

2.3 Analysis Tools

One important thing for a company to achieve best industry practise is to ensure they have good functioning tools as aids to the work and business structure. This chapter will briefly describe some tools that have been useful in this project and for sure are useful for other companies.

2.3.1 Hummingbird BI

The package from Hummingbird BI includes software applications for querying, reporting and HyperCube purposes, which have all been used in this project. For queries, BI Query has been used. The program uses data tables from the data model created from SAP and it supports different kinds of joins between the tables. The result from a query is easily exported to the program BI Query Report for reporting and to BI Cube Creator to create a multi-dimensional data source. This gives the user the possibility to shape the cube the way he wants, that is decide whether or not to include calculated fields or which part of the result should serve as a metric. From this step or directly from BI Query, BI Analyse can be launched to view the resulting cube. Some design work with the cube can also be performed here, but not to the same extent as in BI Cube Creator.

For the ordinary user of the report, the people that the reports are being created for, BI Web is used as user interface between the raw data in the reports and the users. With an ordinary web browser BI Web can be used to run the data model, look at reports or do your own queries. (BI Query™ Data Models, 2002; BI Query™ Query, 2002; BI Query™ Reports, 2002; BI Cube Creator™, 2002; BI Web™, 2002)

2.3.2 Hummingbird Genio Suite

The previous section described the tools that were used for creating the graphical result that the users see. Another part of the project is the scheduled reminder that the target group is to receive, for this purpose Genio Designer and Genio Scheduler is used. With Genio Designer data sets, modules and processes to be run can be created. The data sets are merely a joining between different data tables while the modules manipulate the data received from either a data set or a table to get proper results. Each individual module can be executed and run. However, within a process different modules can be run in a specific order and it is possible to arrange so that e-mails are sent out depending on, for example, the result from a module. Genio Designer also serves as a tool to create the raw data used for reports when BI Query can not solve the task.

Genio Scheduler is used for scheduling and real time administration of processes in Genio Designer. It is also possible to view the history of previous executions. With this program the processes created within Genio Designer can be scheduled to run at a specific time instant, with a certain time interval or when triggered from another event. (Genio Designer, 2002)

3 Analysing Maintenance and Defect Data

The main goal of this project was to develop a generic methodology for the analysis of SAP data for different pieces of transmission plant. The benefit of a generic process is that it can easily be adapted to many different cases, which in this thesis is represented by the different areas within electrical power transmission. The areas included in this project are substation plant, communications, secondary systems, transmission lines and maintenance service providers. The generic process consists of the way to collect data, how to analyse it and how to present the information. Data is gathered from the company's databases the same way for all areas. The raw data does not reveal all information, therefore by processing it, for example by normalisation, hidden information might be identified. Processing and presentation of the data can be done in many ways and is very dependent of what the different areas want. Another important issue is that information has to be presented in a way that is suitable for its purpose, which means that the way the data is presented in the reports can differ due to the areas of interest.

This chapter will introduce the reader to the generic process developed, all parts from collecting the data to the data model created, the user interface between the user and the raw data. To prepare the reader for the case example in chapter 4, the last part of this chapter describes the different parameters (graphs) chosen together with information on how exceptions have been raised for them.

3.1 Overview

Originally the concept was to create the generic model with one single multi-dimensional data source containing the data for the reports. In BI User a single query should have generated a HyperCube containing all information requested by the users for the different plants. The cube was then to be published to the repository (where all data models and other published BI material is stored and retrieved) and viewed via BI Web with certain pre-defined views of the data saved. However, because of the following problems the idea had to be abandoned:

- The user can easily change views of the cube saved in the web-based system. This makes it not suitable to use since many users could modify the cube.
- The cube became too vast and sparse, many null values, and also proved hard to implement when including all interesting parameters (equipment, notifications, cost etcetera). When creating one HyperCube with all data for all plants, it was difficult for the users to get the information they required easily.

This resulted in changes of the approach and instead of using a cube as source for standard views, a number of separate standard reports had to be created in BI Report and collected together in a specific data model. By this the users have some possibilities to change the information in the reports by entering the desired values in the prompts. This made the cube in some ways unnecessary but it was decided to keep it so the user can manipulate and find data that is not presented in the other standard reports. Using a simplified cube not including notification information solved the problem with the size.

The authors research resulted in a generic model built up by two bigger blocks as shown in Figure 3-1 below. One of the blocks consists of the actual data model, the ad hoc reports and the HyperCubes, while the other block contains the Genio part. The Genio section schedules scripts for populating tables, calculation of thresholds, which are compared with the actual data values in the reports, and raises exceptions based upon the comparisons. This block also sends out e-mails to remind the users about data on the web pages, and if exceptions have been raised, this will be included in the e-mail. This will be further described in section 3.5.6.

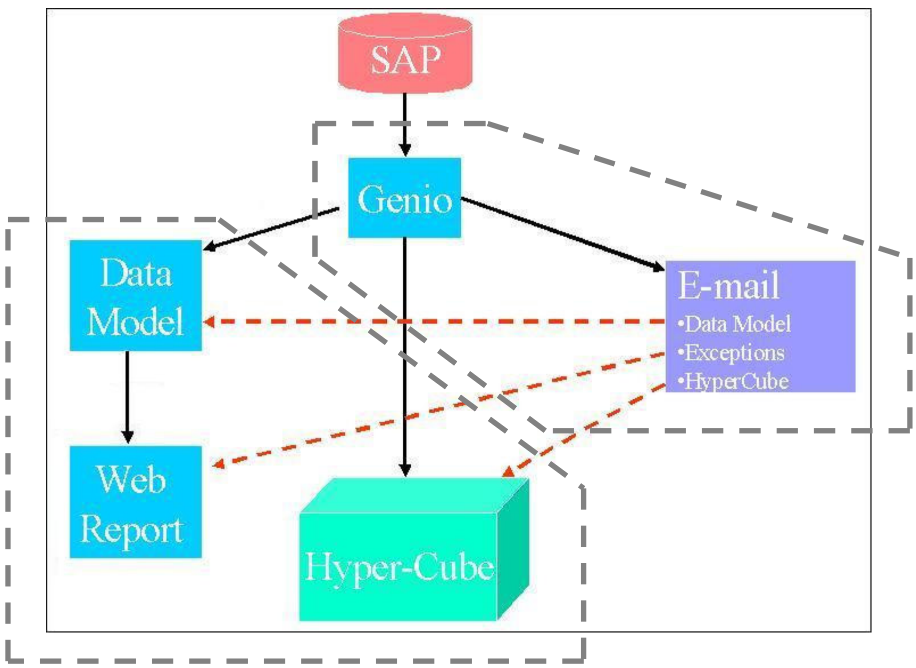


Figure 3-1 Overview of the process behind the data model

The following sections will describe the different parts of the generic model, everything from the collection of data to the report created and e-mail sent to the strategy engineers.

3.2 Data Collection

The data that is collected from the electrical power grid for Powerlink is stored in many different tables in a couple of databases. One big source of data is the SAP database, which has been the main source of data for this project. SAP stores the data in different ways including ordinary tables, internal tables and views. Of these three data objects only the ordinary tables are accessible from the outside. Internal tables and views are made up from a collection of columns from the ordinary tables and calculated data, and can only be used within SAP.

For this project a new database called *AMT database* has been created that contains tables created from the SAP database. The tables only contain relevant maintenance and defect data and logical English column names instead of the German shortenings used within SAP. This means that the amount of data that has to be searched every time a parameter is needed is limited and will therefore ease the work for the technicians usually interrogating SAP for data.

The process in collecting relevant data from the AMT database is quite similar to the old method, with the big difference that the new method works automatically. The old method interrogated SAP to retrieve data, Microsoft Access was used to join tables represented by spreadsheets and Microsoft Excel was used to create graphs used for the analysis. Instead of using these two different software applications, Hummingbird BI is used which makes it easier to generate reports, HyperCubes and exceptions directly from the query result.

The tables within the AMT database are updated on a daily basis with fresh data from SAP. When users are running a query to collect data, the result will therefore always be up to date. The queries are built in such a way that the desired columns are displayed once they have been run and the result can be modified by certain calculations. (Bannister, 2003)

3.3 Processing and Using the Data

As mentioned earlier, the main process can mainly be divided into two parts:

- One part with queries put up and run and then used for generation of reports
- The scheduled process involves running scripts for calculation of exceptions and the sending of e-mail. Also some of the data for certain reports are generated here.

The tables put up in the AMT database are joined in such a way that one column can be viewed as a function of any of the others, for instance the Equipment and Notification tables can be joined for viewing the number of notifications per model. Once a query has been created and a result has been achieved, it can be used as a source for the web based reports. The step in going from the result to start creating the actual report that the user will see, is just a loading of data to BI Reports and then choose the way the report should be presented, that is as a table, crosstab, graph or a combination of these three. In the report the data can then, to some extent, be processed and used to put up new columns, for example if a ratio between two different columns is to be displayed.

Once a report has been created it can easily, with a single click, be updated from its web page via BI Web. This simply means that the query is run again and the old data in the report is replaced with new.

3.4 The Automated Process

The main goal with the project, as mentioned in earlier paragraphs, is to make the process from the collection of data to the generation and publishing of reports on the web automatic. The automatic parts in the two above mentioned different blocks differ a little bit from each other. The outcome is the strategy engineers within the different plant areas will receive a number of e-mails with a chosen time interval, containing:

- the number of exceptions raised for each report and technical object type
- hyperlinks to the different web based reports
- an address to the homepage containing all the reports
- an address to the web based HyperCube

3.4.1 BI Query User and BI Reports

Once a parameter has been collected or calculated within BI Query, the result is easily exported to create a report. The underlying query for the report is not stored in BI Query, instead it is saved within the report and simply launched into BI Query when the report is opened.

To make the updating process of the reports automatic, together with limiting the number of reports needed, ad hoc reports are used instead of static ones. In these kinds of reports prompts appear in which the users for example type in the specific object and date range they want to look at when opening a report. This requires the query for the graph to be run again, and therefore updates the report with new data.

3.4.2 Genio

The second block, generation of exceptions and sending reminders to the users, is scheduled to run with an interval based upon customers request. It has been decided to begin to run the process every three months. This block is divided into two natural parts:

- where the queries are set up and run in Genio Design. When running the modules that contain the queries, thresholds are calculated and the parameters are compared with them. An exception is raised if the parameter in question reaches beyond a critical value. This part is also used for setting up the e-mail that is sent out to the users with a predefined interval, containing the raised exceptions and links to the reports. The separate parts, modules and mailing, are combined together into a process that can be executed and run in one step.
- the use of Genio Scheduler where the process within Genio Design is called on at a desired time instant. This makes it possible to schedule the creation of exceptions and sending of e-mails.

3.5 The Model Created

The model created consists of several parts as mentioned above. Some are visible for the users while some are in the background and generate data, exceptions and are only dealt with if the AMT engineers need to maintain the data model.

3.5.1 SAP Notifications Data Model

SAP Notifications is a model created in BI Query and published to the repository where it is stored and can be viewed with BI Web, the user interface. When opening up the model, the user comes to the welcome window, see Figure 3-2 below, where each strategy engineer can find his own area represented by a button.

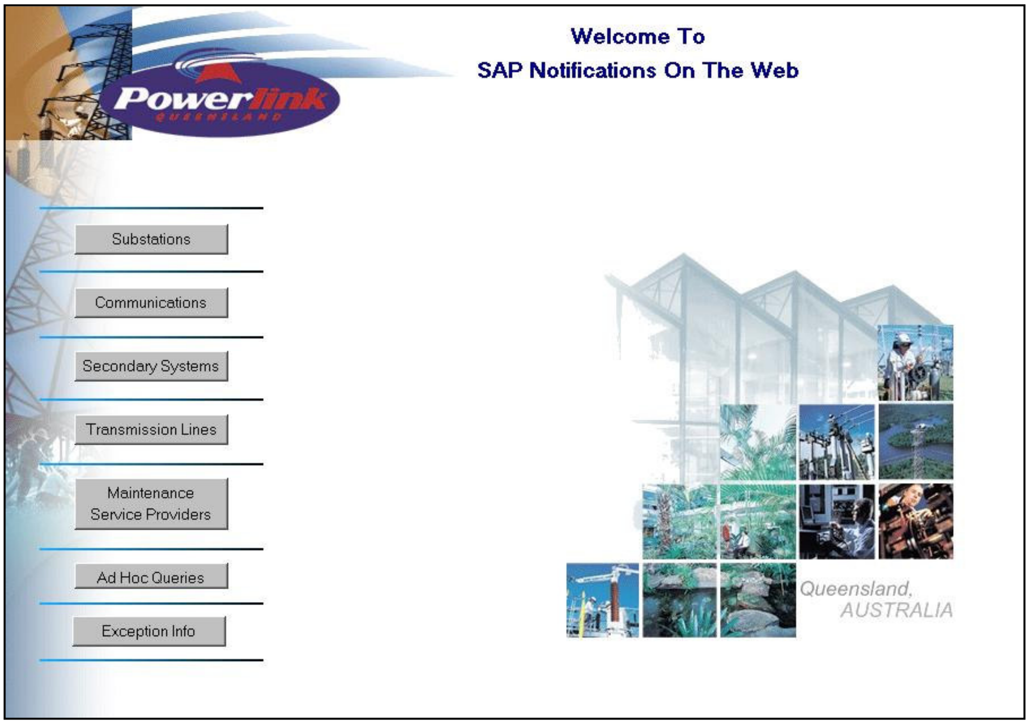


Figure 3-2 The welcome window of SAP Notifications data model on-line

This makes all areas available to all engineers, a good thing, since in some cases they can face problems extending over many areas. There is also a button for ad hoc queries where the engineers can set up their own queries to get data not displayed with the standard reports. The last button in the welcome window takes the user to an information page where theory behind exception raising is explained.

Once a specific area is chosen, the user comes to the window containing all parameters that are represented for the area. As seen in Figure 3-3, for the areas substations, communications and secondary systems the users will meet the same structure; graphs divided into objects, models and regions and also one part for raw data which takes the user directly to a query. There is also a query in the model data and one in region data, used for creating the population tables for the respective area of plant.

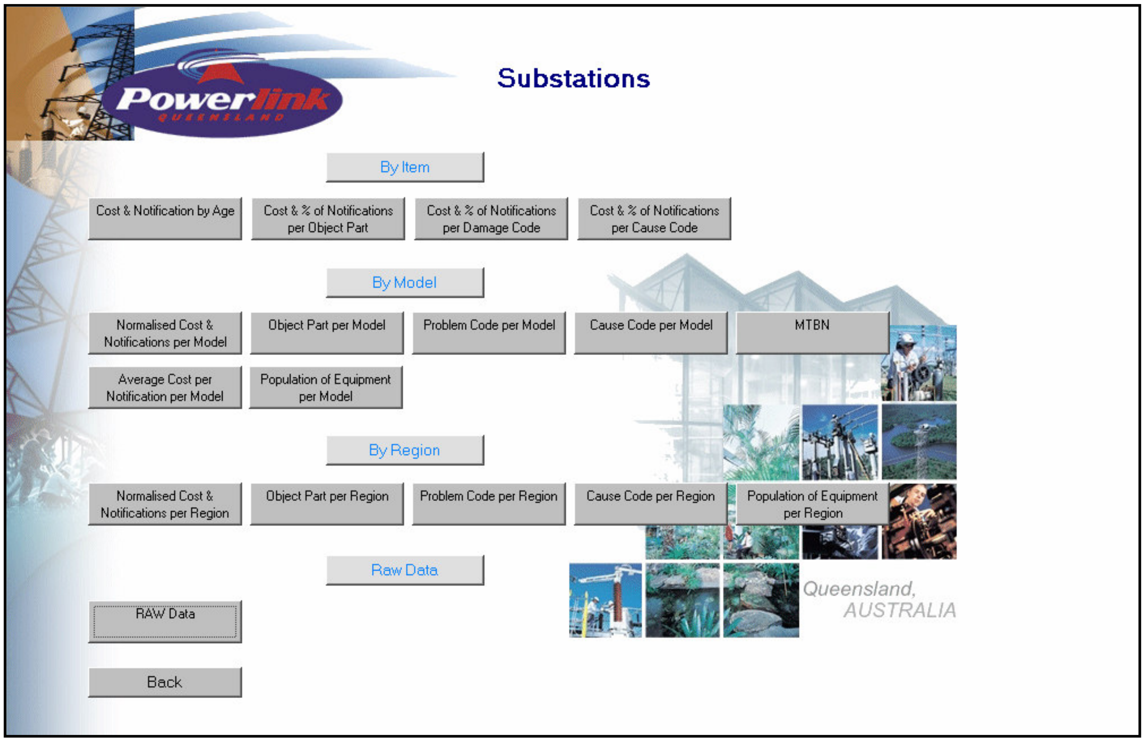


Figure 3-3 The substation plant window with all available reports

When it comes to the appearance of the Transmission Lines window, it differs from the above ones since the reports are divided into current financial year and previous financial year. A financial year begins the 1^{st} of July one year and ends 30^{th} of June the next year. The reports are also divided into easement maintenance cost and plant maintenance cost on request from the Transmission Lines strategy engineer.

The window for maintenance service providers is also quite different since it expresses a lot of information about the planner groups. Each region within Powerlink's electrical power grid is divided into several technical areas such as substations, communication systems and the group that manages the maintenance in one of these technical areas within a region is a called a planner group. There is also a parameter for the different priorities (priorities for the notifications).

All strategy engineers have the option to search for any other parameter that is not presented on their respective page. This is possible through the ad hoc queries button on the welcome page. This button takes the user to a window with, for this project, the relevant tables, notifications, equipment and orders extracted from the backside of SAP. The users find support of how to use the tables to build queries and run them in the user manual developed especially for this project and attached as Appendix B.

3.5.2 BI Reports

When the users are using the model and click on a button to display a graph, they will in all cases, except for when requiring the maintenance service provider graphs, end up in a window with prompts. These prompts are for technical object type, date range, voltage level etcetera depending on which parameter they choose to look at. The graphs displayed have the standard format as shown in Figure 3-4. In this graph the user will see:

- Header: the chosen technical object type, region, voltage level or other useful information.
- Sub-header: date range for the notifications used in the graph
- Title: Name of the graph
- Legends: telling which quantities that are displayed in the graph
- Threshold info box: displays the thresholds for the exceptions

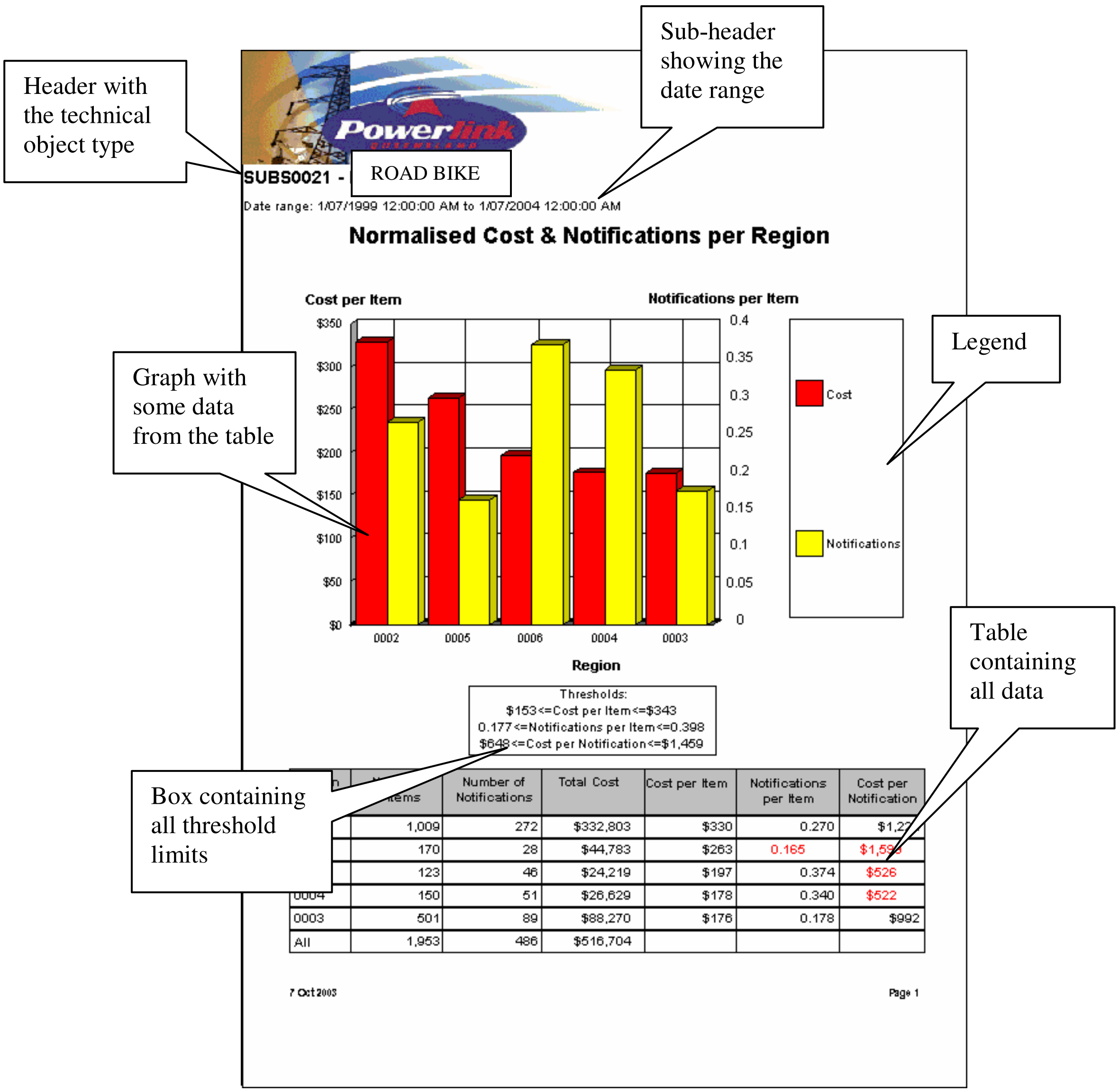


Figure 3-4 The report Normalised Cost & Notifications per Region showing road bike data

The functions within BI Report allow standard deviation to be used for setting threshold limits. The limits can be visualised as horizontal lines in the graphs, but unfortunately they can not be modified, only one standard deviation above and below the mean can be shown. Therefore it can not be used for the purpose in this project since the distribution of the data demands that threshold of two standard deviations above and 0.95 below the mean is used. The thresholds are discussed in the coming section 3.5.3.2, The Exceptions.

The table, which also plays a fundamental part of the report, will have values over and under the thresholds centred and highlighted in red.

When it comes to the last point, there are some limitations with the software used today. The table can be slimmed down to, for instance, show only the top ten values while the graph still shows everything. This is also discussed in section 5.2.1.

3.5.3 The Reports and Their Exceptions

Retrieving all data is one task, another is what to do with the data once you have access to it. This section will explain how the authors have used the data, mainly presented as graphs or tables. The different graphs used to create reports will be examined deeply, why these were chosen and how they are created will be described in section 3.5.3.1. Also some of the rejected graphs will be mentioned, explaining why they were of no use. Then from the graphs used exceptions will be discussed.

By raising exceptions on graphs the user can focus on the most critical parts, and perhaps only look at a slice of the data. Exceptions have been raised for specific values and ratios exceeding a threshold. The different thresholds used have been decided after studies of the graphs and talks with the strategy engineers. The process of how to raise exceptions, on what, why and when will be explained in section 3.5.3.2.

The work with trying to figure out which graphs would provide the strategy engineers with useful information started with bringing forward all possible graphs, then a critical review sorted out which graphs could be useful. An early review was done in cooperation with the strategy engineers as a way to introduce them into the project. All graphs that were not normalised were cleared out since comparison between different data can not be performed easily.

Together with the graph, a table is also presented for the strategy engineer. The table gives an opportunity to look at the exact values if they are not obvious from the graph. The data found in the table is exactly the same as presented in the graphs.

3.5.3.1 Graphs

Cost and Notification by Age (Bathtub curve)

The bathtub curve is a very common graph used all over the world to get information about how the life cycle of different objects looks like, and serves as a good aid for the strategy engineers when it comes decisions on what equipment to replace. The bathtub curve reveals running-in problems which, in general, tend to occur for all different items. This can be seen as a high number of failures in an early stage when an object has recently been put into service, and is often referred to as the *early failure period*. After a certain time in service the failure frequency, in general, tends to drop. The object enters a period of a low number of failures called the *intrinsic failure period*. The period is the operating lifetime for the object. After the hopefully long period of almost failure free operation the objects come into a new period called the *wear out failure period* were the equipment starts getting aged. The age symptoms are shown as increasing number of failures with the age. A high number of failures in the beginning and end of an item's lifetime together with few failures in between, makes it look almost like a bathtub, hence the name bathtub curve, see Figure 3-5 below.

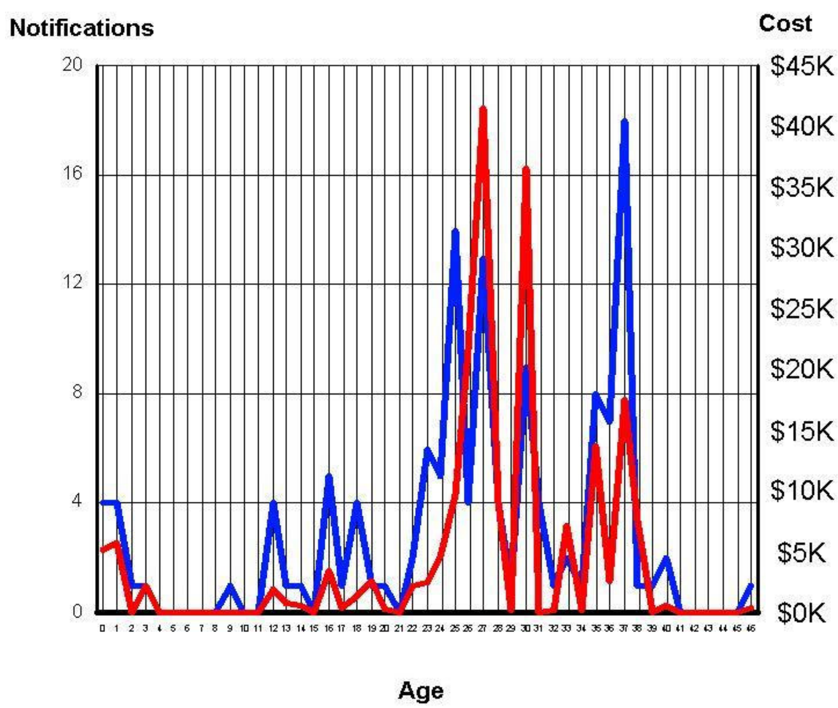


Figure 3-5 Bathtub curve based on both numbers of notifications (blue) and cost (red)

In this project a bathtub curve has been graphed for all manufacturers and models of a certain item of plant. There was not enough data to get decent bathtub curves for the individual manufacturers. Two different curves have been used to present bathtub graphs. First the ordinary graphs representing number of notifications raised for an item at different ages, and then it has also been chosen to view how the cost for items at different ages changes. This was done to see if there was some correlation with the ordinary bathtub curve. If the engineers want to have a look at the bathtub curves per model this can be done by ad hoc queries, but this is not recommended yet since the lack of data gives sparse graphs.

For some equipment it is possible that they can be refurbished, that is when the bathtub curve enters the wear out failure period. If the equipment is brought back to a near new condition the bathtub curve will enter a new intrinsic failure period. The shape will look like a bathtub curve with a peak in the middle that represents the time when the object was refurbished. It might be possible that this can be repeated over and over again for certain objects.

Cost and % of Notifications per Object Part Code/Damage Code/Cause Code

The cost is something the strategy engineers are always interested in. Combined with the number of notifications this is very useful information, which can be used to determine where to put the resources. The cost and number of notifications can be combined in many different ways for looking at the data. In this section the parameters are joined to look at which object part code have had notifications, the damage code and the cause of the notifications. The same total cost and number of notifications are used in three different graphs. The cost used is the average cost for a notification and the number of notifications is a percentage of a specific failure, see Figure 3-6.

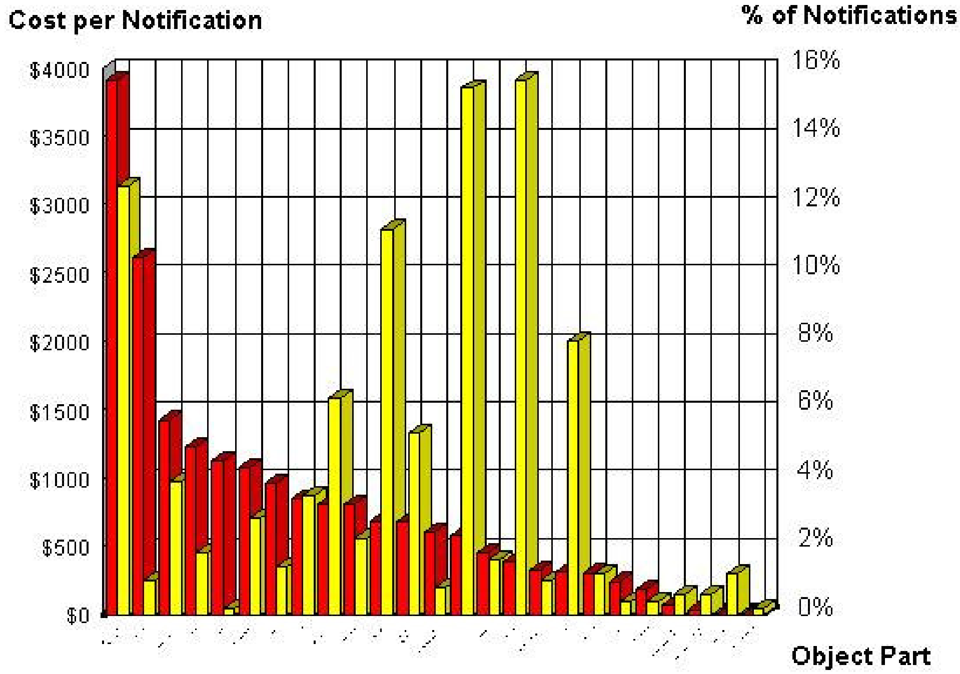


Figure 3-6 Normalised Cost (red) and Notifications per Object Part (yellow)

The useful information found in this graph is that costly failures for a specific item of plant can be identified and focus can be turned to them to try saving money. A costly failure is identified as a cost per notification ratio that is higher than the average ratio. The notification data alone can be used to identify frequently occurring failures which might be possible to avoid in the future.

When looking at notification and cost per object part separately reveals object parts that are expensive or have a lot of failures. Possible solutions might be to change manufacturer if the part is expensive, or to use another design of the part if it fails very frequently. The data for damage code can also give an idea of how to change a design of an object or how to better maintain the object to prevent failures in the future, which in the end would result in money saved for the company. Changing the focus to the cause of failures gives valuable information on how to prevent failures from occurring, but still some of the causes might not be possible to prevent, at least not to a reasonable cost.

<u>Normalised Cost and Notifications per Model</u>

The equipment used by a company for one task might be of many different manufacturers and models due to different cost or performance when buying the equipment. Therefore it is necessary to get an idea of how the cost and number of notifications for all models vary. This graph will be a good help for the strategy engineer when making decisions whether to buy new equipment, when to buy it or how to change the maintenance performed.

The strategy engineers' attention should be focused on the costs and number of notifications, normalised by the number of equipment, that exceeds a predefined threshold. This indicates that something might be wrong and need further investigation. For instance one model can have a design failure or the maintenance has become inefficient. Focus should also be drawn to models where the ratio between cost and number of notifications differing from the normal, this is presented in a separate graph. This could reveal very expensive or cheap maintenance, which could be due to the performance of maintenance.

<u>% of Notifications per Object part/Damage Code/Cause Code per Model</u>

This graph is useful together with the *Normalised Cost and Notifications per Model* and serves as more low-level information compared with the mentioned one. Besides knowing what models that are costly and have notifications raised frequently against them, it is also valuable to know what object part is having a lot of notifications and the damage or the cause of notification for the different models.

By using the information in this graph money can be saved. If one object part has a lot more notifications than the rest it could be due to poor design. Making the strategy engineers aware of the problem could result in change of design of the object part. The damage can be investigated and the cause of notification can be removed, if cost effective. If the cause is due to environmental stresses, the information can be useful to decide which equipment to place where, since the stresses might have different affects on different models.

<u>Mean Time Between Notifications (data table only)</u>

The mean time between failures (MTBF) is a useful parameter telling about the reliability of a model, that is the time between failures occurring. It is defined as the age divided with the total number of failures according to Equation 3-1 below. In the implementation the age is calculated from the assumption that all equipment was placed into service at 1st of July 1999 when SAP was introduced to prevent faulty start up dates to be used, see section 5.1.2 for more information about faulty start up dates. Compared to the bathtub curve this parameter can be viewed on a model level since there is enough data to give valuable information.

$$MTBN = \frac{\#Equipment \cdot \#Years\,in\,service}{\#Notifications}$$

Equation 3-1 MTBN index

The index gives an idea of how long equipment of a certain model will run before it fails, on average. In this project the mean time between notifications (MTBN) has been used instead of the MTBF. The reason for using notifications instead of failure is the data available. When a failure occurs, this is indicated with a notification being raised. However a notification is also raised for non-routine maintenance work to be carried out and so far there is no method for identifying only the time of failures in the SAP data.

The information that is revealed within the MTBN table is very useful for the strategy engineer when planning maintenance for the equipment and setting up the maintenance interval. The maintenance interval is typically never longer than the MTBN value for a model since this means that equipment of that model will always, statistically, fail before maintenance is performed on it unless an operate to failure policy is adopted. To avoid this, the maintenance interval should be adjusted so that it is shorter than the MTBN. However this does not mean that failures will be avoided completely.

Mean Time To Repair (data table, not included in the SAP Notifications)

The mean time to repair (MTTR) is the mean time it takes to repair equipment and is displayed on a model level. The parameter is calculated from the total time for all repairs on the different models divided by the total number of notifications according to Equation 3-1. Since no start up date is included in the equation an exact expression can be used for the calculation.

$$MTTR = \frac{\sum mal\,duration}{\#notifications}$$

Equation 3-2 The MTTR index

For a company struggling with high maintenance cost this could be a very useful tool for discovering models causing a lot of trouble and therefore requiring a great deal of maintenance. There might be a possibility to improve maintenance performance for the models and save some money since the strategy engineers get an idea of what models are taking a long time to repair and require more spares. By further investigations with MTTR as a underlying source, things such as the performance of the maintenance crew in different regions can be analysed. This might show that a service team in one region is slower than the average team.

Average Cost per Notification per Model

As a graphical complement to the previous graph, *Normalised Cost and Notifications per Model*, *Average Cost per Notification per Model* gives further information about the cost-notification relation for the models. The graph shows models that have had high average cost for notifications. This can serve as a help to finding places to save money on, but notice that a high average cost might not tell the whole truth. It could be that a model has high average cost per notification due to few expensive failures. However, if the number of equipment of the specific model is high, the model is cheap to run when considering cost per equipment. So the conclusion is that the parameter has to be examined with care and the meaning of the result depends on what the user is after. The conclusion that the model is expensive to run since it has a high cost per notification can therefore be wrong, if for instance the number of notifications is very low and an average cost is the case.

When using this graph/parameter together with *Normalised Cost and Notifications per Model* an interpretation on the difference between a running and a failing mode can be done. The resulting conclusion can be that the model is cheap to run when no failures occur, but very expensive once the failures show up, which is true when the number of notifications are low and the cost is high.

Normalised Cost and Notifications per Region /
% of Notifications per Object Part Code/Damage Code/Cause Code per Region

The area in which the company operates is divided into different logical geographical regions to ease management and planning. Now the costs and number of notifications can be presented for the different regions, this gives a better idea of how they change in different environments and work conditions.

These graphs are the corresponding ones to the model graphs such as *Cost and Notifications per Model*, but serve as a high-level tool to see how the overall business is working and where a bigger effort has to be put in for reducing costs or notifications.

3.5.3.2 The Exceptions

The graphs presented in the previous paragraphs provide the strategy engineers with a lot of valuable information. Not to overload them with information and risk that some important data is lost it is suitable to raise exceptions wherever it is possible at certain time intervals. The exceptions should link to the report in question, giving the strategy engineers the option only to focus on critical and major problems in the electrical power grid. This will reduce the probability that problems are missed.

First the thought was to raise exceptions based on the average and a scalar factor but since this would not be statistically correct another method had to be used. The other method that will be used is based on mathematical statistics. It is preferable first to identify the distribution of the data. If the data is normal distributed the exception limits can be set up easily with existing functions in the BI software. But this can not be assumed and therefore a significance test has to be performed. After identifying the distribution of the data it has to be decided how the exceptions should be raised and on what causes. (Sharp, 2003)

All parameters in the reports are not suitable to raise exceptions for due to the items in some reports can not be compared. The strategy engineers have had the opportunity to give their comments on what reports to raise exceptions for. Due to limitations in the software used and the nature of the data it was not possible to implement all requested exceptions.

The following text discusses different reasons to raise exceptions. Then the authors will go through in detail how the distribution of a set of data is determined. The paragraph after that will briefly discuss the reports for which exceptions are being raised for followed by an explanation why exceptions have not been implemented in the other reports.

3.5.3.2.1 Possible Causes to Raise Exceptions on

When the data has been collected and processed it is time to think of what else can be done with it. A good suggestion is to look at how exceptions can be implemented. Two interesting possibilities to raise exceptions on exist. The data that deviates from the theoretical probability curve, low and high peaks far from the curve, and data at the top and bottom tail of the curve. Usually 95% of the data is covered and exceptions are raised for the other 5%, that is 2.5% for the top and the bottom. This is often done because of when the data is normal distributed two standard deviations up and down from the mean would give the limits. But these values are to narrow to use if the data is gamma distributed as suggested by Mr. Brian Sharp (2003).

If the distribution is gamma, covering 90% of the data and raising exceptions for the bottom and top 5% of the data will be a better solution. In Figure 3-7 an ordinary case for a gamma distribution is shown. Both the real and the theoretical distribution are shown in the figure. The possible causes to raise exceptions on have been marked in the figure.

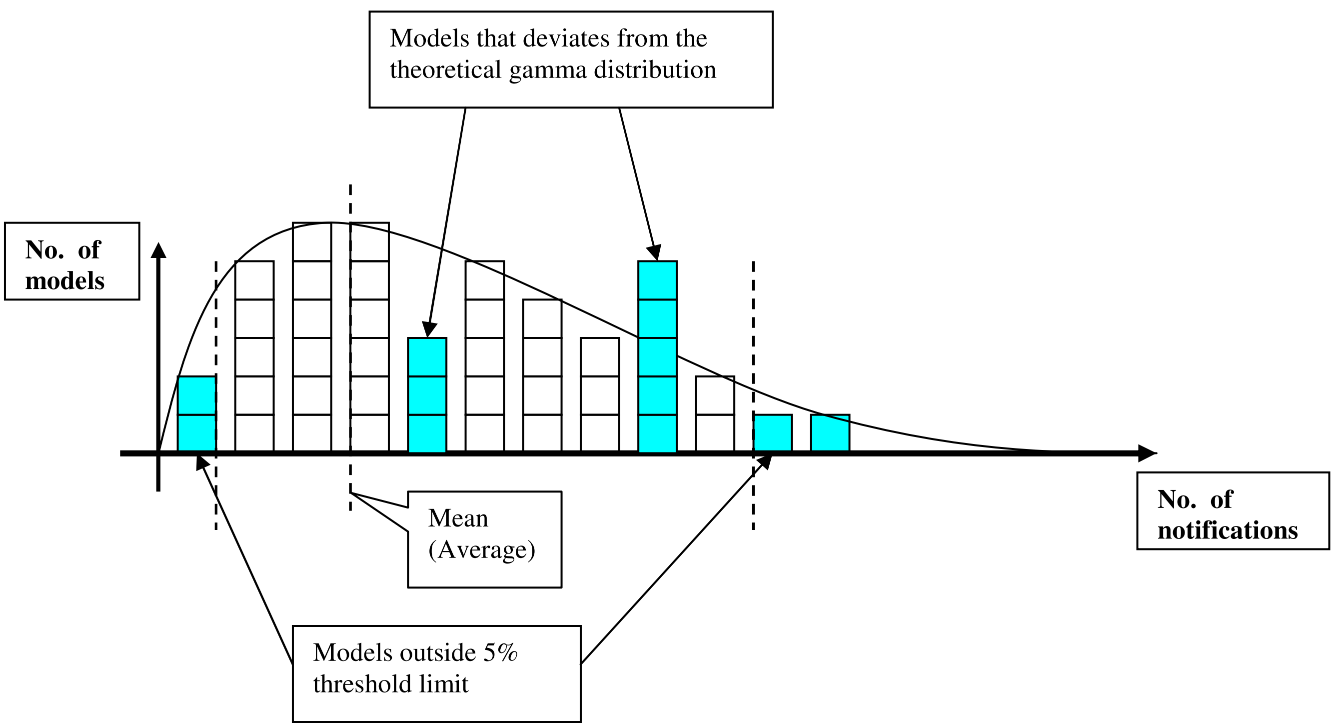


Figure 3-7 The general gamma distribution depicting the different ways of how to raise exceptions

In this project it is more interesting to raise exception for the top and the bottom values rather than the deviating once. One reason is that the users would not easily recognise why exceptions have been raised for models with costs in the middle of the table and a detailed explanation of the exceptions would be needed. Looking at the top and bottom values feels more natural and it is easier to understand how the exceptions are being raised. The next step is to determine the distribution.

3.5.3.2.2 Determine the Distribution of the Data

The reason behind that the distribution has to be determined mentioned in the previous paragraph, if it is normal distributed the limits can easily be calculated with existing functions in BI Report. These limits would work dynamically with every report independently of what the users entered in the prompts, and no other work with how the populations change over time have to be done. But if the data is not normal distributed the calculations are a little bit more complicated and the limit might no longer be possible to calculate within the report. They would have to be calculated with Genio Designer and as a result of this the limits might not be dynamic, which means they would not change depending on the values in the prompts. These limits could still be valid if the population of the data does not change over time, which it is not expected to, but of course this has to be verified first.

Looking at how the distribution is determined, in this example consider the normalised notification data from the report *Normalised Cost and Notifications per Model*. The data is loaded into a spreadsheet in Microsoft Excel from the report. Then a histogram is created to get a picture of how the distribution looks like, this is seen in Figure 3-8.

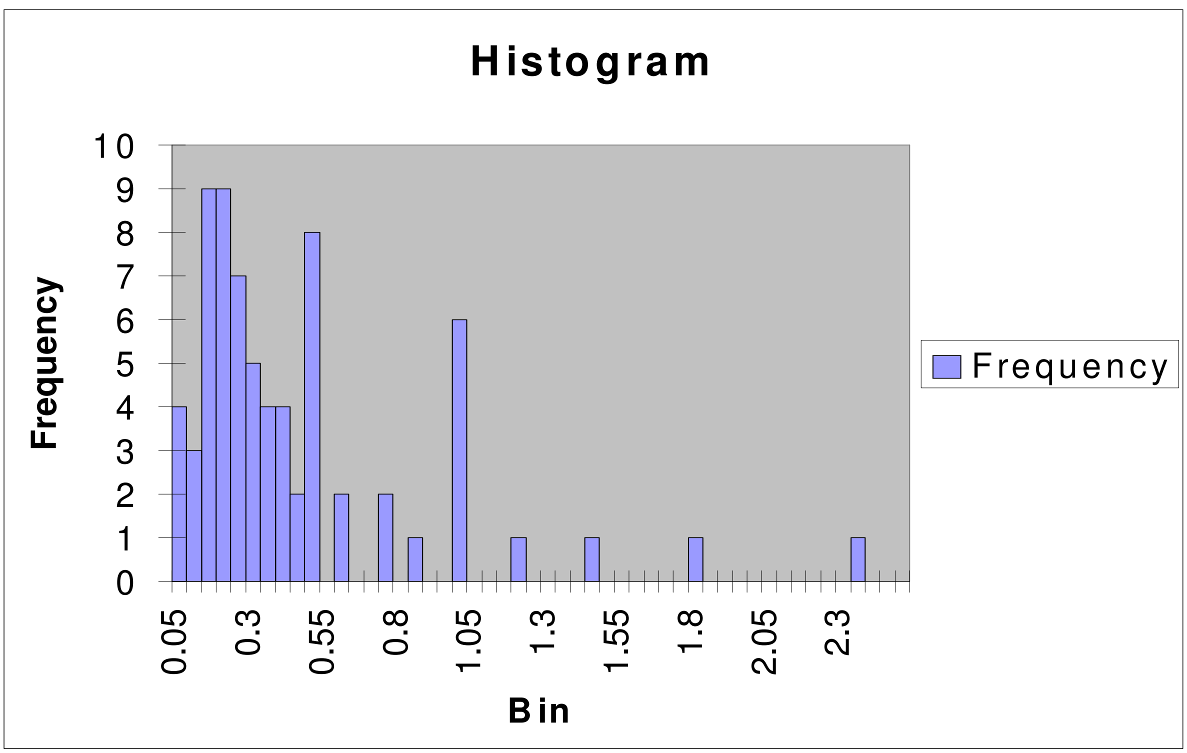


Figure 3-8 Histogram showing the distribution of normalised notification data from the report Normalised Cost and Notifications per Model.

A number of distributions might fit the data, but from earlier text in this report it is known that this kind of data is expected to be gamma distributed. Therefore a test to check this is performed. First the actual probability distribution for the data is calculated and compared with the theoretical.

The actual probability function is calculated in the following way:

$$P(x) = \frac{\textit{Number of Events in Intervall}}{\textit{Total Number of Events}} = \frac{\textit{Frequency}}{\textit{Total Number of Events}}$$

Equation 3-3 The probability function of the actual data

A little bit more work has to be done to calculate the theoretical distribution. First the true mean is calculated:

$$\mu = \frac{\sum_{i=1}^{n} x_i}{n}$$, where x_i is data sample number i, and n the total number of samples.

Equation 3-4 The expression for calculating the true mean

Then two parameter estimations, the frequency λ, Equation 3-5, and η, Equation 3-6, have to be done before the theoretical probability function can be calculated.

$$\lambda = (n-1)\frac{\sum_{i=1}^{n} x_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

Equation 3-5 Estimation of parameter λ used in the gamma probability function

$$\eta = \mu\lambda$$

Equation 3-6 Estimation of parameter η used in the gamma probability function

The probability function is then:

$$f(t) = \frac{\lambda}{\Gamma(n)}(\lambda t)^{n-1} e^{-\lambda t}$$, where t is the middle point in each interval.

Equation 3-7 The theoretical gamma probability function using the estimated parameters

By calculating the theoretical probability distribution it can be compared with the actual distribution and the result is shown in the figure below.

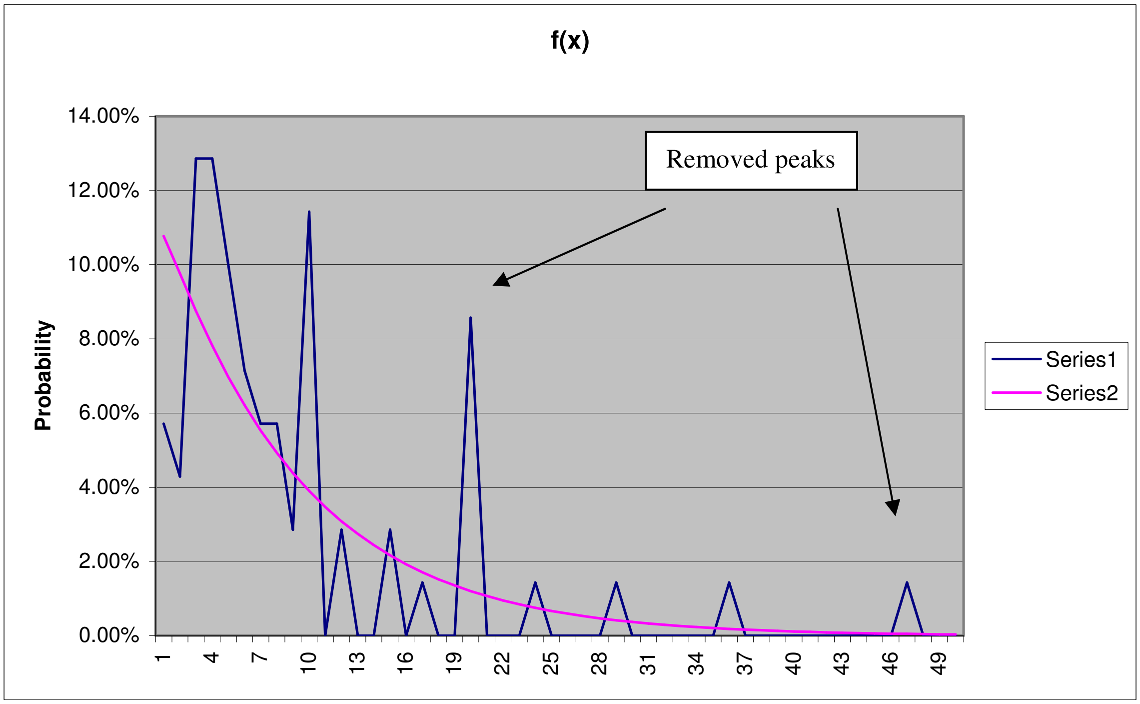


Figure 3-9 A comparison between the true and the theoretical distribution of the data.

The next step is to perform the significance test and that will tell how well the data fits to the theoretical curve. First the null hypothesis and the alternative hypothesis is set up.

Null hypothesis, H_0: The data can be modelled as a gamma distribution

Alternative hypothesis, H_a: The data can not be modelled as a gamma distribution

The expected values E_I, Equation 3-8, in different points is then calculated with help of the theoretical probability function and compared with the actual value in the test statistic, Equation 3-9.

$$E_i = f(i)n \text{ , for i=1..n}$$

Equation 3-8 Expected value at i

$$X^2 = \sum_{i=1}^{n} \frac{(x_i - E_i)^2}{E_i}$$

Equation 3-9 The test statistic used in the significance test to check if the data is gamma distributed

The result from the test statistic is used together with the degrees of freedom of the data, Equation 3-10, to determine the level of fitness with help of the Microsoft Excel function for chi-squared distribution, CHIDIST, Equation 3-11. Instead of using the function in Microsoft Excel the value can also be retrieved using chi-squared distribution table.

$$Degree\,of\;\; freedom = (n-1)$$

Equation 3-10 Calculation of degree of freedom for data

$$Fit\ Level = CHIDIST\left(\chi^2, Deg.of\;\; freedom\right)$$

Equation 3-11 Microsoft Excel function CHIDIST used for chi-squared test

The first result is a fit level of 0.0%, which is not so good since this indicates that the data can not be assumed to be gamma distributed. But after looking at the data in Figure 3-9 the authors realise that there are a few values heavily affecting the result. Removing the two peaks affecting the fitness level the most, these are marked in the figure, the result increase to 66.3% and this means that H_0 is not rejected. But removing those two peaks has to be justified. The peak in the middle is due to a lot of models with only one equipment have had only one notification each raised so far. It is then probable that the normalised value 100% will be more common than other values, therefore this peak is removed. The other peak comes from a single model with relatively few equipment for which the value is rather extreme compared to the other values, therefore this peak is removed. The significance test is performed for one item of plant for the current graphs and it is shown that all data in these reports can be assumed to be gamma distributed. It is assumed that the data for all items of plant is similarly gamma distributed, which it most certain is. (Sharp, 2003)

The distribution of the data might change over time, from gamma distribution to some other distribution. By performing a t test of data from two different time intervals it is possible to verify that the distribution does not change, or do change, over time. The t test is conducted by first gathering the data. In this project the test is done on disconnectors, where the data is the normalised cost per equipment for different models. The data is divided into two time periods, 1999-2001 and 2001-2003. These two populations of data are then compared to see if there are any changes over time.

The data from the two periods is copied into two columns in a Microsoft Excel spreadsheet. The columns are sorted so that same objects from the different periods are matched with each other. The differences, D_i, of the data values from the matched objects, X_i and Y_i, are calculated:

$$D_i = X_i - Y_i$$

Equation 3-12 Difference of the data from two time periods

The next step is to estimate the variance, σ^2, for the differences with Equation 3-13.

$$\sigma^2 = \sum_{i=1}^{n} \frac{(D_i - D)^2}{n-1}$$

Equation 3-13 Estimation of variance

In Equation 3-13 D is a parameter calculated in Equation 3-14 below.

$$D = \sum_{i=1}^{n} \frac{D_i}{n}$$

Equation 3-14 Calculation of parameter D used to estimate variance

As with the significance testing for gamma distribution, a null and alternative hypothesis also have to be set up for the t test. In the null and alternative hypothesis below μ is calculated with Equation 3-4.

Null hypothesis, H_0: $\mu_x = \mu_y$, the two means are equal

Alternative hypothesis, H_a: $\mu_x /= \mu_y$, the two means are not equal

The test statistic is:

$$T = \frac{D}{\left(\frac{\sigma}{\sqrt{n}}\right)}$$

Equation 3-15 Test statistic for t test

By using a level of significance of 0.1, $\alpha=0.1$, H_0 will be rejected if the absolute value of T is greater than 1.697. When using the data in the different equations above in Equation 3-15 the result –1.459 is achieved, which means that H_0 is not rejected at 5% confidence. It has been verified that the data is not expected to change over time.

It is recommended the distribution of the data is checked annually. But since the t test showed that the data is not expected to change this might be done even more seldom. Since all data now is assumed to be gamma distributed, a solution for how to find the limits has to be developed.

3.5.3.2.3 How the Limits are Calculated

To raise exception limits, *Tchebychev's Approximation* may be used. Tchebychev's Approximation says that the probability that a random variable x is greater than k standard deviations from its mean is less than $1/k^2$. The meaning can be expressed as:

$$P\left(\left|x-\mu\right| \geq k\sigma\right) \leq \frac{1}{k^2}$$

Equation 3-16 Tchebychev's Approximation

In Equation 3-16 σ is the symbol for standard deviation and it is calculated in the following way:

$$\sigma = \sqrt{\frac{1}{(n-1)} \Sigma \left(x_i - \mu\right)^2}$$

Equation 3-17 Standard deviation

This means that the probability that x will take a value greater or less than for example three standard deviations from its true mean for any distribution is less than $1/k^2 = 1/3^2 = 11\%$. This is an overestimation for the symmetrical distributions and is in fact much smaller. The Gamma distribution is skewed to the right, so Tchebychev's Approximation will not be suitable for accurate determination of the limits.

However, the limits can easily be calculated when the theoretical curve has been created. The probability function is integrated till 5% is covered and the lower threshold is set there, then the function is integrated again until 95% is covered and the upper threshold is set there. However this method can only be used in Genio Designer and not in BI Reports. This means that the thresholds would be static and the change of data over time would have to be checked now and then.

For the gamma distribution different values for the low and high threshold have to be used because the distribution is unsymmetrical. For an exponential distribution the following applies:

$$\begin{aligned} &\textit{Lower threshold}: \mu - 0.95\sigma = 5\% \\ &\textit{Upper threhold}: \mu + 2\sigma = 95\% \end{aligned}$$

Equation 3-18 Threshold estimation for exponential distributions

The exponential distribution is a special case of a gamma distribution when the quotient of the mean divided by the standard deviation is equal to one. These rules for exponential distributions, Equation 3-18, can be applied since the gamma distributions in this project are almost exponential. This is indicated by the fact that in general in this project the quotient is almost equal to one. In Table 3-1 the values of μ and σ for the normalised notification data are shown together with the ratio between them.

Table 3-1 Exponential rule of thumb used on notification data for disconnectors

Report	Mean, μ	Standard deviation, σ	Ratio, μ/σ
Normalised Cost and Percentage of Notifications per Object Part Code	0.040	0.048	0.833
Normalised Cost and Percentage of Notifications per Damage Code	0.050	0.065	0.769
Normalised Cost and Percentage of Notifications per Cause Code	0.083	0.108	0.769
Normalised Cost and Notifications per Model	0.428	0.424	1.009
Percentage of Notifications per Object Part Code per Model	0.314	0.294	1.068
Percentage of Notifications per Damage Code per Model	0.343	0.300	1.143
Percentage of Notifications per Cause Code per Model	0.294	0.203	1.448

From that it can bee seen it is possible to calculate dynamic exceptions. The next section will give a brief overview of how the exception raising is implemented in the reports.

3.5.3.2.4 *<u>How the Limits are Implemented in the BI Software</u>*

In each and every report that exceptions are raised for the thresholds are calculated the same way. And once they have been calculated they are also used in a similar way. The way they are calculated and raised is:

1. Calculate the mean
2. Calculate the standard deviation
3. Calculate the low/high threshold
4. Raise exceptions for values exceeding the thresholds

The values for which exceptions are raised will change colour to red and be centred to make it easier to find them. In the reports the specific thresholds are display to inform the user of where the limits are situated.

The next two sections deal with the reports for which exceptions have been raised and for the ones exceptions have not been raised.

3.5.3.2.5 *<u>Reports where Exceptions have been Implemented</u>*

Only a number of the reports are used for exceptions raising. The following seven reports are the ones for which exceptions are being raised, ordered after appearance in the data model:

- Normalised Cost and Percentage of Notifications per Object Part Code
- Normalised Cost and Percentage of Notifications per Damage Code
- Normalised Cost and Percentage of Notifications per Cause Code
- Normalised Cost and Notifications per Model
- Percentage of Notifications per Object Part Code per Model
- Percentage of Notifications per Damage Code per Model
- Percentage of Notifications per Cause Code per Model
- Average Cost per Notification per Model

In these reports the exceptions are being raised for parameters that are normalised, otherwise it would not be necessary to have exceptions. Without normalised values comparing different objects would not be of any value. The different parameters used to raise exceptions on are:

- Cost per Item
- Cost per Notification
- Notifications per Item
- Notifications per Fault Code

3.5.3.2.6 *Reports where Exceptions have not been Implemented*

Briefly the cause of why the reports in this section are not considered for exceptions will be explained. In general all of the reports mentioned here are used to get high level information.

Cost and Notifications by Age (Bathtub curve)

The main purpose with this report is to provide an overview for the strategy engineers since it contains all equipment for a specific item of plant. Therefore it is not suitable to raise exceptions because the equipment might behave different. But it would be interesting to raise an exception when specific equipment is approaching the end of the bathtub, that is when the number of notifications and total cost starts to increase. A way to implement raising of exceptions could be to track a change in the average number of notifications calculated for the last couple of years. When the average exceeds a certain threshold, an exception should be raised. The threshold needs to be chosen big enough to prevent exceptions are being raised for small deviations from the intrinsic failure period (section 3.5.3.1).

Mean Time Between Notifications (MTBN)

As with the bathtub curve the purpose with this report is also to give the user high level information. It would have been very suitable to raise exceptions for MTBN values falling below the time between service set up in the maintenance strategies, otherwise the equipment for sure would have a notification before maintenance. The strategy engineers did not show interest in exceptions for this data and therefore it was not implemented.

Normalised Cost and Notifications per Region

The report gives an overview about different regions and it would be interesting to compare the regions and see if some region falls out of the pattern. To be able to compare the regions it first has to be confirmed that the regions belong to the same population. If they did not come from the same population comparing the regions would be as meaningful as comparing pears and apples. Then the values presented in the report are averages from the different regions that in most cases would fall well inside limits set up. This together with limitations in the software is the reason that exceptions were left out of this report.

Percentage of Object/Damage/Cause Code per Region

The data in this report is used to give the users, the strategy engineers, an overview of the type of notifications in different regions. There has not been a wish from the strategy engineers for raising exceptions for this one.

3.5.4 The HyperCube

The HyperCube is, as mentioned before, an efficient tool for storing and viewing all possible data that may be of interest for the strategy engineer. Inclusion of a cube in the work means that the model should be complete, that the data that can not be found in the reports can be found in the cube by drilling to the suitable parameter of interest. Ideally, notifications, equipment, object parts/damage/cause, cost, etcetera should be included. The cube becomes harder to work with and structure up when the data consists of a large number of big dimensions, which is the case in this project if all data is used in one cube. Therefore the cube in this case has been limited considering the number of dimensions included, since a small and fully functional cube rather than a big non-functional cube has been the goal.

3.5.5 BI Web - what the Users will see

To make the data model, reports and HyperCube visible to the user, a user interface is needed. It is here BI Web is used, a program which allows users to log on to a system using a simple web browser. The data model, reports and HyperCube must be published to the repository on the server before the different items can be used. In the web, there is limited ability for a user to make any changes, except for in the cube, so the visual appearance of the model and the reports will be the same who ever uses it. The AMT engineers who create and maintain the model have a different view of the model than the user, due to the fact that they have to be able to implement changes if needed. But this is done with the other software presented; BI Query, BI Reports and Genio Designer.

When entering the BI Web, the user comes to the repository where all web-based material is stored, all reports, models and cubes. For this project the catalogue SAP Notifications has been created to gather all reports, the model and the HyperCubes to ease the navigation in the repository. The user has also the possibility to look at the most recent scheduled jobs, where scheduled reports and HyperCubes are displayed. (BI Web™, 2002)

3.5.6 Genio Projects

The Genio part of the model is a part that the user will not see, it is only visible for the creator and staff maintaining it. However, if no extra parameters are to be added or the thresholds to be changed, no further work with this part will have to be done.

As mentioned above in section 3.4.2, the work done in Genio Designer can be split into two blocks. One part dealing with the scripts written for updating tables, code for data manipulation and calculation of exceptions, and one part dealing with the scheduling of the processes. The Genio Designer part can be viewed in the following hierarchical way:

- the projects collects all processes, modules, variables etcetera
- the processes runs a number of modules and scripts
- the modules consists of a script to be run
- the data sets set up the data source for some of the modules

The data sets respective modules created are located under Genio Objects - Data Sets/Modules and the processes under Genio Objects - Processes. For the AMT staff to be able to understand how to update and work with this part of the model, the different processes within the projects are shortly described below, see Appendix C for further details. (Genio Designer, 2002)

3.5.6.1 SAP Notifications Data Tables

This project is created for daily updating of the main tables with data from SAP. The tables updated are Notifications, Equipment, Orders and some other necessary ones used in the data model, together with some of the different data tables concerning substations, communications and secondary systems used for the reporting. The data tables for *Maintenance Service Providers* and *Transmission Lines* are also updated on a daily basis, but are still kept under their individual projects to retain a good structure of the work performed in Genio.

For the main tables, the process *SAP Table Scripts* runs a number of SQL scripts to create and populate data tables used by some reports and to raise exceptions. When all scripts have been run, the process ends up with triggering an event called *SAP Notif*. This event serves as a triggering for the other process in this project, SAP Notifications Data Tables, to run, which, as described above, updates substation, communication and secondary system tables. As seen in Figure 3-10 below, this process includes:

- Execution of *Notif_mod*, a module that filters the real Notifications table, so that only relevant notification dates are included.
- Running of *Update_mod*, which updates all the data tables that is used in the reports.
- Execution of module UpdateAvAge_mod that updates the table containing the average age for all different models. It is only scheduled to run once every quarter, which is expressed in the condition.

- *"Send Mail to…"* sends an e-mail to the administrator to confirm that the tables have been updated.
- Triggering of event SAP Notifications Data Tables, which triggers the start of other processes in other projects every day.

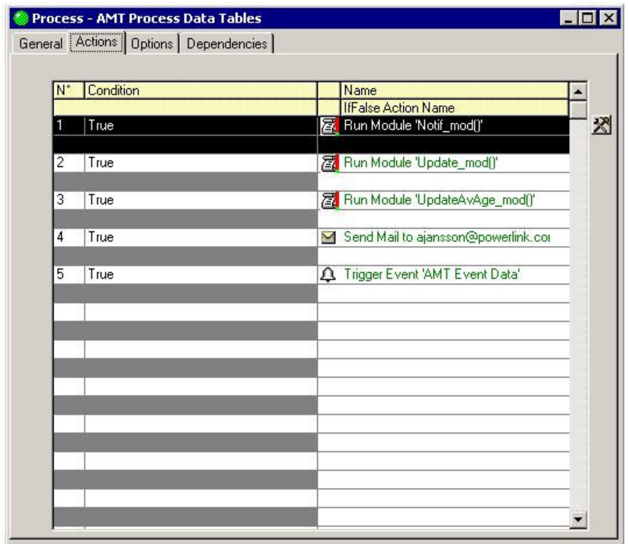


Figure 3-10 SAP Notifications Data Tables

3.5.6.2 SAP Notifications Service and Lines

These projects create, as SAP Notifications Data Tables, data for the reports considering Transmission Lines and Maintenance Service Providers. Each project contains two processes, one to update relevant data calculated and another to send e-mail to the user with a predefined interval. Since there is no span of technical object types needed for either of the projects, the updating processes are very simple and only consist of the one time running of modules.

3.5.6.3 SAP Notifications Data Substation/Comsys/Secsys

These projects are created to raise exceptions for their individual technical object types. Since it is only the object type that differs the processes from each other, the structure of modules, data sets and processes are the same, which makes it fast to implement the structure for all three areas. The one and only process that exists within each of these projects is triggered once every quarter by the process AMT Project Service and contains the following parts, also shown in Figure 3-11:

- Running of *TechObj_mod*, a module that creates a table called *Tech_Objects*, with all different technical object types for each project.
- Running of module *DeleteRow*, which takes the first value in the table Tech_Objects, saves it into a variable and then deletes the chosen row.
- Execution of *TotalModule* that collects and runs all the relevant modules, raising of exceptions etcetera for all technical object types.

- *"Send Mail to…"* sends an e-mail for every technical type to the user with the number of exceptions raised for each graph.
- *"Goto line #…"* as long as there is still objects in the technical object table, all modules from the specific line number are run again, for the next technical object type.

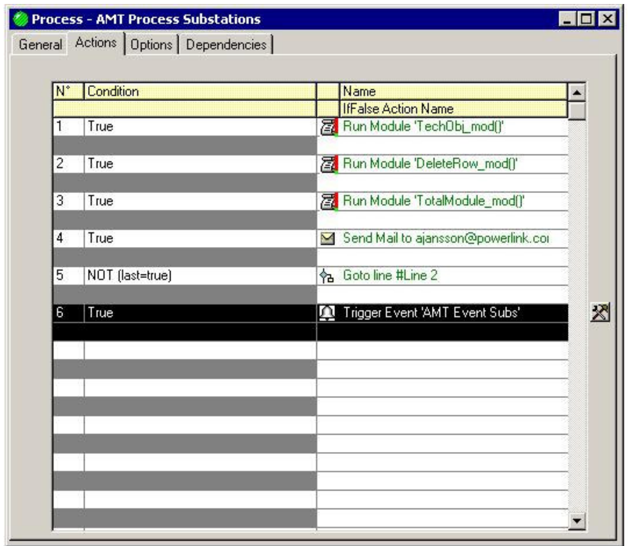


Figure 3-11 Example of process within Genio Designer, in this case SAP Notifications Substations

3.5.7 Genio Scheduler

As described earlier, one of the goals with the project is to make the raising of exceptions and sending of mails automatic with a specific time interval. In the previous section, the way the exceptions are generated by running processes were presented for the reader. However, as is obvious, this is not an automatic process since for generation of exceptions once every quarter would require a manual triggering of the processes. Therefore, a program that collects all processes within the different projects and runs them on a regular basis is needed, a task that has been assigned to *Hummingbird Scheduler*.

The order of execution of the processes has been chosen according to Figure 3-12 where the bell indicates the special event that triggers the execution of the next process. The execution is set up to follow a chain for both the daily execution and also the quarterly execution, to minimise the risk that to much data is being processed each time and also to avoid sharing violations between projects that use the same tables. In Figure 3-12, the triggering marked with star (*) indicates that the processes involved only are executed when the triggering is done on a specific day, that is the first Monday in every quarter.

As seen, by using Genio Scheduler, the only thing that has to be scheduled is the time for the SAP Table Scripts to be executed. The program allows execution to be done at a specific time every day, which is suitable since the data tables are to be refreshed every day, and this is preferably done during nights to minimise load on the data network.

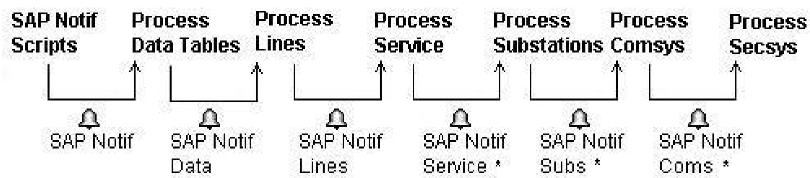


Figure 3-12 Order of execution of the different Processes existing in Genio Designer

4 Case Study

In the previous chapter, the reader was introduced to the main goal of this project, the generic process, whereby a high level description of the model is outlined. To further increase the understanding of the use of the model this chapter will cover two different case examples on which the model is applied.

The two different cases presented covers one with sufficient data available and one with minimal and bad quality data available. This will first show how valuable it is to make use of all gathered data instead of just storing and not processing it, which is the case at many companies today. Data that can be used to keep track of what is happening with the assets, and show possibilities where to cut costs. The second example then highlights the importance of first gathering the data, because without data no analysis is possible.

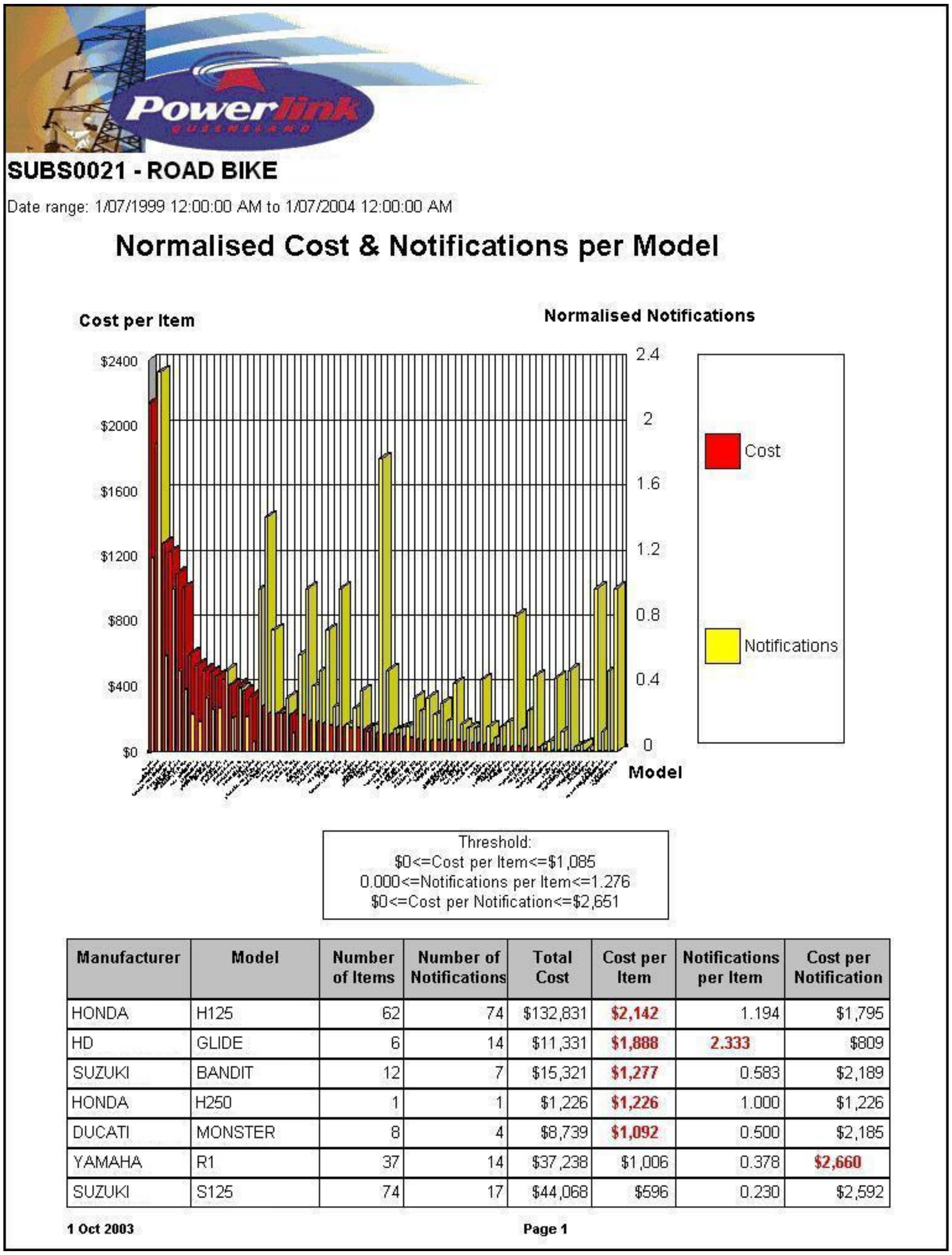
The two different cases are represented by two different electrical equipments. Due to security issues the technical object types, the manufacturer names and model names have all been changed and are now represented with fictitious names.

Road bikes have been chosen to be studied for the case with sufficient amounts of data. The reason is that there are many road bikes in service and they have had a number of notifications. In total there are 1,953 road bikes and a total of 610 notifications raised against them since 1st of July 1999. The *dirt bike* will cover the case with few data and, as the case with road bikes, it has a high number of equipment operating, a total of 1,749. But the amount of notifications is lower compared to road bikes, only 39 notifications. In this case the lack of data is due to high reliability of the asset which of course is good, but creates problems when using the data model.

4.1 Sufficient Data

In this part the authors will demonstrate the usefulness of the data model, how the predefined reports can be used to discover deviations from the normal. This is important since deviations might be or become costly. Then other reports will be used to further investigate the cause of the deviation and possibly find a solution to the problem. The example will demonstrate how an investigation about a specific model can be done, of course the user can also chose to look at different regions or for the whole group of one item of plant for instance.

The first report to look at, at model level, is most logically the *Normalised Cost and Notifications per Model* report, which gives a good overview. This report lists data for all different models for, in this case, road bikes in a table, and in a graph the normalised costs and numbers of notifications are displayed. In Figure 4-1 the first page of the mentioned report is shown.



SUBS0021 - ROAD BIKE

Date range: 1/07/1999 12:00:00 AM to 1/07/2004 12:00:00 AM

Normalised Cost & Notifications per Model

Threshold:
$0<=Cost per Item<=$1,085
0.000<=Notifications per Item<=1.276
$0<=Cost per Notification<=$2,651

Manufacturer	Model	Number of Items	Number of Notifications	Total Cost	Cost per Item	Notifications per Item	Cost per Notification
HONDA	H125	62	74	$132,831	$2,142	1.194	$1,795
HD	GLIDE	6	14	$11,331	$1,888	2.333	$809
SUZUKI	BANDIT	12	7	$15,321	$1,277	0.583	$2,189
HONDA	H250	1	1	$1,226	$1,226	1.000	$1,226
DUCATI	MONSTER	8	4	$8,739	$1,092	0.500	$2,185
YAMAHA	R1	37	14	$37,238	$1,006	0.378	$2,660
SUZUKI	S125	74	17	$44,068	$596	0.230	$2,592

1 Oct 2003 Page 1

Figure 4-1 First step in performing an analysis by looking at the report Normalised Cost and Notifications per Model

From the graph on the first page of the report it is easy to identify models with high values on normalised cost and number of notifications. They are all indicated with high bars, and approximate values can be identified via the axis. However the actual values and other info can be found in the table. Looking at notifications there are a few models differing from the rest, and *HD Glide* has the highest normalised value of them all, 2.3 notifications per piece of equipment. Now questions like why it has many more notifications per equipment than the rest are being raised.

To further investigate the reason for the high normalised notification value other reports will be used. The report to look at next is *Object Part per Model*, which will give information from the notification of what object part has been raised against for different models. In Figure 4-2 a page from the report is shown, and here the data for *HD Glide* is found. The object part with the most notifications for that model is *Main drive/gearbox* with a total of 78.6% of all notifications raised against it.

Powerlink
QUEENSLAND

SUBS0021 - ROAD BIKE

Date range: 1999/07/01 to 2003/10/01

Manufacturer	Model	Object Part	Number of Notifications	Total Number of Notifications	% of Total
KAWAZAKI	NINJA 5	CONTACT ARMS	3	3	100.0%
HD	SOFT	STATION POST	6	11	54.5%
		BRAIDS/FLEXIBLES	4	11	36.4%
		MAIN CONTACT	1	11	9.1%
	GLIDE	MAIN DRIVE/GEARBOX	11	14	78.6%
		CONTACTOR	2	14	14.3%
		MAIN CONTACT	1	14	7.1%

2 Oct 2003 **Page 9**

Figure 4-2 Further investigation performed with help of the report Object Part Code per Model

Since there have been quite a few notifications in total for *HD Glide* it is strange that a majority have been raised against a particular part. A logical step in the investigation is to continue to check information about what the causes have been. The report *Cause Code per Model*, shown in Figure 4-3, will give all information about the cause of notifications raised against different model. This report tells the user that a majority of the notifications, 57.1%, have been raised due to *Design or manufacturer cause*.

SUBS0021 - ROAD BIKE

Date range: 1999/07/01 to 2003/10/01

Manufacturer	Model	Cause Code	Number of Notifications	Total Number of Notifications	% of Total
HONDA	SUPERSPORT A	<null>	1	15	6.7%
	SUPERSPORT B	NONE OF THE LISTED	1	1	100.0%
	TOUR500	<null>	15	23	65.2%
		ABNORMAL	3	23	13.0%
		NORMAL	3	23	13.0%
		FLORA / FAUNA	1	23	4.3%
		UNINTENTIONAL	1	23	4.3%
	TOUR750	NORMAL	1	2	50.0%
		DESIGN OR	1	2	50.0%
	SPORT B	NORMAL	1	1	100.0%
	SPORT A	NORMAL	1	1	100.0%
KAWAZAKI	NINJA 5	NORMAL	3	3	100.0%
HD	SOFT	NORMAL	5	11	45.5%
		DESIGN OR	2	11	18.2%
		NONE OF THE LISTED	2	11	18.2%
		ABNORMAL	1	11	9.1%
		IMPROPER	1	11	9.1%
	GLIDE	DESIGN OR	8	14	57.1%
		ABNORMAL	4	14	28.6%
		NORMAL	2	14	14.3%

2 Oct 2003 Page 7

Figure 4-3 The report Cause Code per Model enables the analyst to draw some conclusions

Conclusions that now can be drawn are that this particular model might be faulty from production or have poor design. This information can be used in negotiations with the manufacturer for example to improve design, replace equipment, warranty matters, possibilities to get discount or to support a decision not to buy more equipment of that specific model.

The investigation above has focused on the number of notifications, a further step is to look at the whole population of that model, how often they are failing and the average age. This is first done with the report *Mean Time Between Notifications* where the MTBN and average age for *HD Glide* are found, see Figure 4-4. The MTBN for the model is only 1.8, hence every equipment is expected to have a notification more than every second year, which is not good. Also notice that the average age of the model is high, 27.5 years.

SUBS0021 - ROAD BIKE

Date range: 1999/07/01 to 2003/10/01

Manufacturer	Model	Number of Notifications	Number of Items	MTBN [year]	Average Age
HONDA	SUPERSPORT C	0	3	999.0	24.2
HONDA	SUPERSPORT D	0	1	999.0	30.2
HONDA	BURN	0	3	999.0	24.2
HONDA	TOUR500	23	16	3.0	7.8
HONDA	TOUR750	2	2	4.3	4.2
HONDA	SPORT C	0	1	999.0	4.2
HONDA	SPORTD	0	4	999.0	19.7
HONDA	SPORT B	1	4	17.0	4.2
HONDA	SPORT A	1	7	29.7	40.2
KAWAZAKI	NINJA 6	0	2	999.0	25.2
KAWAZAKI	NINJA 9	0	2	999.0	19.7
KAWAZAKI	NINJA 5	3	5	7.1	40.2
KAWAZAKI	NINJA 11	0	8	999.0	35.7
BUELL	SPORT	0	1	999.0	35.2
HD	SPORT	0	7	999.0	4.2
HD	SOFT	11	40	15.4	4.2
HD	GLIDE	14	6	1.8	27.5
HD	CRUISE	0	4	999.0	4.2
All		486	1,953		

2 Oct 2003 **Page 5**

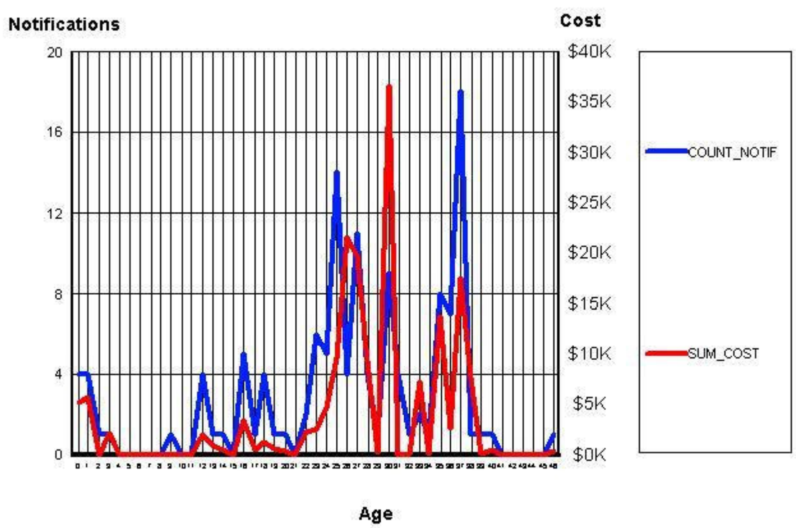
Figure 4-4 The MTBN report also contains average age data used in this case study

The high average age motivates the user to have a look at the bathtub curve for disconnectors, which is found in the report *Cost & Notification by Age*. The bathtub curve gives an idea of how the life cycle looks like for a specific type of equipment, in this case the road bikes. It is possible to identify when the equipment is getting old, that is, when the number of notifications increases and it may be time to change it. The report, Figure 4-5, shows an increasing number of notifications around the age of 25. This is a good indication of that road bikes should be replaced at 25 years unless it is possible to refurbish them to a near new condition, which then would allow the model to restart the bathtub curve.

SUBS0021 - ROAD BIKE

Date range: 1999/07/01 to 2003/10/01

Cost & Notification by Age



Total Notifications	Used Notifications	% Used
486	130	26.7%

Age of Item	Number of Notifications	Total Cost
0	4	$5,124
1	4	$5,734
2	1	$0
3	1	$2,221
4	0	$0
5	0	$0
6	0	$0
7	0	$0
8	0	$0

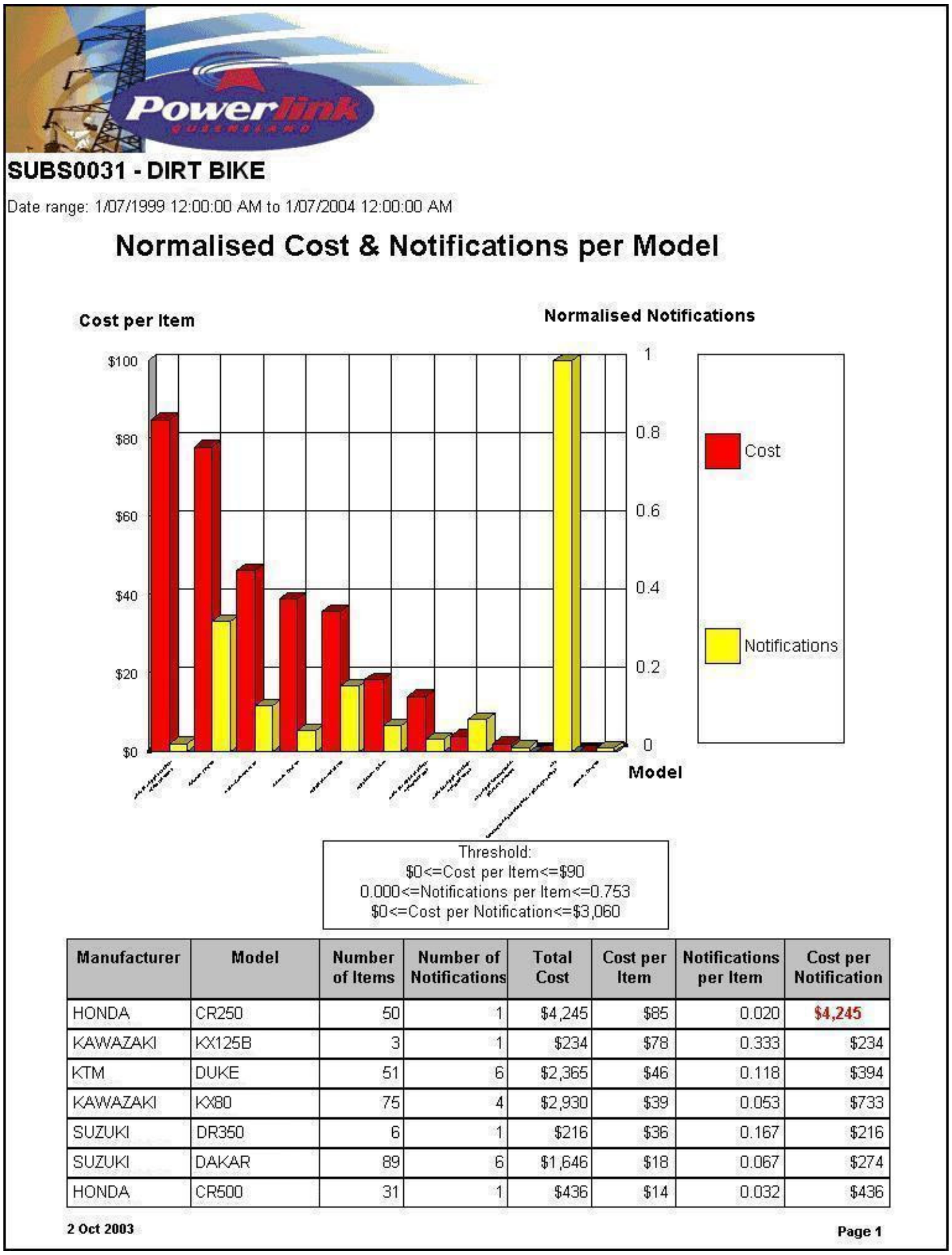
2 Oct 2003 Page 1

Figure 4-5 Bathtub report shows the life cycle of disconnectors in this case

4.2 Lack of Data

This section will, unlike the previous, deal with a case when there is not only lack of data but also poor data quality. It is necessary to understand how to handle the situation with this kind of data and how the data model reacts. The same reports as for the previous case with sufficient data will be used in this section, and the same path will be followed.

First the report *Normalised Cost and Notifications per Model* is used to identify an interesting model to focus on. In Figure 4-6 one of the yellow bars deviate very much from the rest. This bar represents the normalised number of notifications, one notification per item, for *Husaberg B FC250e*.



Manufacturer	Model	Number of Items	Number of Notifications	Total Cost	Cost per Item	Notifications per Item	Cost per Notification
HONDA	CR250	50	1	$4,245	$85	0.020	$4,245
KAWAZAKI	KX125B	3	1	$234	$78	0.333	$234
KTM	DUKE	51	6	$2,365	$46	0.118	$394
KAWAZAKI	KX80	75	4	$2,930	$39	0.053	$733
SUZUKI	DR350	6	1	$216	$36	0.167	$216
SUZUKI	DAKAR	89	6	$1,646	$18	0.067	$274
HONDA	CR500	31	1	$436	$14	0.032	$436

Figure 4-6 Identification of interesting model with lack of data

The investigation continues to the object part report shown in Figure 4-7, where it is discovered that all notifications, six in total, have been raised for an object part called <*null*>. This is of course not a name for an object part but only shows the fact that the object parts for which the notifications were raised against have not been entered. Since there are a few numbers of notifications and the information has not been entered correctly for this model it makes it hard to interpret the results.

SUBS0031 - DIRT BIKE

Date range: 1999/07/01 to 2003/10/02

Object Part per Model

Threshold:
0.0%<=% of Total<=126.6%

Manufacturer	Model	Object Part	Number of Notifications	Total Number of Notifications	% of Total
HUSABERG A	FC200	PORCELAIN	1	1	100.0%
HUSABERG B	FC250e	<null>	6	6	100.0%
KTM	DUKE	<null>	2	6	33.3%
		METAL OXIDE BLOCKS	1	6	16.7%
		OTHER MECHANICAL	1	6	16.7%
		PORCELAIN	1	6	16.7%
		SURGE COUNTER	1	6	16.7%
HONDA	CR250	PORCELAIN	1	1	100.0%
	CR250R	OTHER ELECTRICAL	1	2	50.0%
		SPARK GAPS	1	2	50.0%
	CR500	EARTHING	1	1	100.0%
KAWAZAKI	KX80	OTHER ELECTRICAL	2	4	50.0%
		METAL OXIDE BLOCKS	1	4	25.0%
		PORCELAIN	1	4	25.0%
	KX125	PAINTWORK/	1	1	100.0%
	KX125B	EARTHING	1	1	100.0%
SUZUKI	DR350	OTHER ELECTRICAL	1	1	100.0%
	DAKAR	EARTHING	3	6	50.0%
		OTHER ELECTRICAL	2	6	33.3%
		FIXTURE/MOUNTING	1	6	16.7%

2 Oct 2003 Page 1

Figure 4-7 Object Part per Model with very few notifications

Looking at the cause of notifications does not give much more help. Also here the problem with null data appears, all six notifications have been raised without entering the correct cause of the notification. This means that no conclusions can be drawn and the reason for the high value of the normalised number of notifications for that model will not be revealed.

SUBS0031 - DIRT BIKE

Date range: 1999/07/01 to 2003/10/02

Cause Code per Model

Threshold:
% of Total < 50%

Manufacturer	Model	Cause Code	Number of Notifications	Total Number of Notifications	% of Total
HUSABERG A	FC400	NORMAL	1	1	**100.0%**
HUSABERG B	FC250e	<null>	6	6	**100.0%**
KTM	DUKE	NORMAL	3	6	**50.0%**
		<null>	2	6	33.3%
		NONE OF THE LISTED	1	6	16.7%
HONDA	CR250	UNINTENTIONAL	1	1	**100.0%**
	CR250R	UNINTENTIONAL	1	2	**50.0%**
		<null>	1	2	**50.0%**
	CR500	UNINTENTIONAL	1	1	**100.0%**
KAWAZAKI	KX80	<null>	3	4	**75.0%**
		NONE OF THE LISTED	1	4	25.0%
	KX125B	NORMAL	1	1	**100.0%**
	KX125	NORMAL	1	1	**100.0%**
SUZUKI	DR350	NORMAL	1	1	**100.0%**
	DAKAR	NORMAL	5	6	**83.3%**
		ABNORMAL	1	6	16.7%

2 Oct 2003 **Page 1**

Figure 4-8 Cause Code per Model, also with few notifications

But still some useful information could be retrieved about the whole population of the model from other reports. A first step is to study the MTBN report that is shown in Figure 4-9. *Husaberg B FC250e* is expected to operate 4.3 years without notifications being raised, and the average age is 103.7 years. Because the number of notifications are so few the MTBN value might not be true. The value for the average age seems to be high compared to other models, the reason for this high value is an error with the start up date for the equipment, more about this can be found in section 5.1.

SUBS0031 - DIRT BIKE

Date range: 1999/07/01 to 2003/10/02

Mean Time Between Notifications (MTBN) per Model

Manufacturer	Model	Number of Notifications	Number of Items	MTBN [year]	Average Age
		0	6	999.0	103.7
HUSABERG B	FC250	0	6	999.0	3.0
HUSABERG B	FC350	0	6	999.0	53.4
HUSABERG B	FC500	0	6	999.0	53.4
HUSABERG B	SWE	0	12	999.0	103.7
HUSABERG B	FC50e	0	24	999.0	91.0
HUSABERG B	THU	0	3	999.0	4.2
HUSABERG B	THU B	0	1	999.0	4.2
HUSABERG A	FC30	0	3	999.0	4.2
HUSABERG A	FC50	0	3	999.0	4.2
HUSABERG A	FC60	0	28	999.0	4.2
HUSABERG A	FC60B	0	9	999.0	70.6
HUSABERG A	FC80	0	9	999.0	4.2
HUSABERG A	FC80B	0	6	999.0	4.2
HUSABERG A	FC125	0	9	999.0	4.2
HUSABERG A	FC125B	0	1	999.0	4.2
HUSABERG A	FC200	1	84	356.7	4.2
HUSABERG A	FC400	0	3	999.0	4.2
HUSABERG C	FC125e	0	3	999.0	103.7
HUSABERG C	FC125ee	0	36	999.0	43.8
HUSABERG C	FC500e	0	3	999.0	103.7
HUSABERG B	FC250ee	0	42	999.0	4.0
HUSABERG B	FC250e	6	6	4.3	103.7
HUSABERG B	FC60e	0	45	999.0	9.8
HUSABERG B	FC80e	0	3	999.0	103.7
YAMAHA	YZ80	0	24	999.0	26.5
KTM	125SX	0	3	999.0	4.2
KTM	525MXC	0	12	999.0	4.2
KTM	400EXC	0	6	999.0	4.2

2 Oct 2003 Page 1

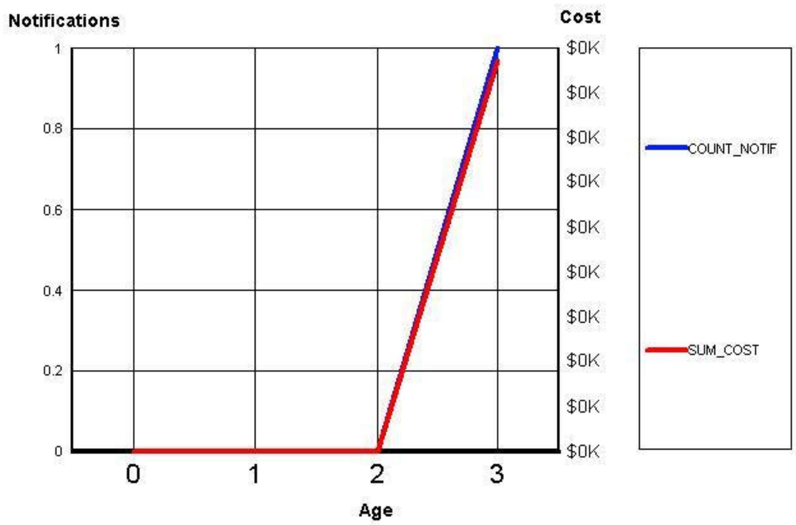
Figure 4-9 The MTBN report shown gives some data that can be used

If the real average age had been available it would have been of value to study the bathtub curve to receive knowledge about the expected lifetime for this particular item of plant. Via this it would have been possible to see at what stage the device is. That is if it was in the early, intrinsic or wear out failure period, and decisions could have been made about replacement or refurbishment of the equipment. Even though it is not useful, the bathtub curve for dirt bikes will be studied to show how it looks like with a small amount of data. The report in Figure 4-10 shows a worthless graph, only one out of 30 notifications has been used and also it only covers the age from zero to three years. The reason is of course the already mentioned problem with start up dates, see the coming section 5.1.

SUBS0031 - DIRT BIKE

Date range: 1999/07/01 to 2003/10/02

Cost & Notification by Age



Total Notifications	Used Notifications	% Used
30	1	3.3%

Age of Item	Number of Notifications	Total Cost
0	0	$0
1	0	$0
2	0	$0
3	1	$436
All	1	$436

2 Oct 2003 Page 1

Figure 4-10 Bathtub curve with extremely low number of notifications

5 Limitations

During the work with this project a great amount of problems were encountered. By far the largest source of problems has been the softwares used, especially the Hummingbird BI Suite, which includes the programs BI Analyze, BI Cube Creator, BI Query and BI Reports and has been used to create the automatic process. The data has also caused some problems due to difficulties to get it and bad quality. Another big problem has been to find information about this subject, which indicates the novelty of the work performed. These problems are described more in detail in the following section.

5.1 Data – Quality and Availability

The quality of the data plays a very important role in a great number of applications. There might be a very good and useful model but if you insert bad data you will, in most cases, for sure have a bad result or no result at all if you do not insert anything. Poor quality of data can be of many different types; the data can be completely wrong, it can be missing or the same data might be represented in a great number of ways.

The data used in this project is typical data collected from any electrical power grid, which has been measured, gathered and stored on a common database. From the database the data in this project has via SAP been extracted and put into the ACMS database, see Figure 1-1, a process that has involved manipulation of the original data. This has caused some problems, most of them concerning dates and manufacturer and model names. In most cases a solution has been found, but there are however problems still remaining and requires a great deal of work with, for instance, tracking of original extraction.

The different problems experienced are explained below depending on their type and some examples that highlight the issues are also shown. Then in section 6.1 the solutions and recommendations are presented.

5.1.1 Lack of Data

Lack of data means that there are no data to process, which makes it hard to test the model and see if it works. The reasons for lack of data may be many, one is that the data have never been collected at all, if this is the case then there is very little to do unless the data can be collected afterwards. Other reasons are for instance that the data could have been lost on the way from the actual extraction to the insertion into the ACMS database or it may have been entered in the wrong field when entering it.

One area where lack of data occurred was when looking at costs associated with functional locations. Then the specifications, for example voltage level, route/circuit length, for the functional locations were not available except at the built section level, which is the highest level of the functional location. The specifications needed to be qualified since the strategy engineers wanted to be able to look at data for different voltage levels.

In Table 5-1 there is an example of a result where some of the data is not available, in this case looking at a couple of different functional locations.

Table 5-1 Extract from Functional Location table

Functional Location	Feeder	Rated Voltage [kV]	Route Length [km]
1000	721/722	110.0000	12.5000
1000-SPN	-	-	-
1000-SPN-050A	-	-	-
1000-SPN-050A-COND	-	-	-
…	-	-	-
1300	-	-	-
1300-UGC	-	-	-
1300-UGC-1_BT	-	-	-
1300-UGC-1_BT-CORE_1	-	132.0000	-

5.1.2 Poor Quality of Data

Poor quality of data does not mean the data is useless but the data will, in many cases, not give a good result. The results retrieved might be hard to interpret due to wrong or bad quality of the data. Hence it will make it hard to test the function of the model, but also limit use to strategy engineers.

Bad quality of data was first encountered when data for different manufacturer and models of certain equipment were retrieved. It appeared as if some of the manufacturers and some of the models had a couple of different names associated with them, also certain models had many different names. When the equipment was put into the database, shortenings and spelling mistakes caused some of the problems, also the fact that a manufacturer was called different names depending where the equipment came from or when the equipment was bought exacerbated the problem. The same holds for the models as well, and the manufacturer sometimes also changed the model name from one year to another. This means that the results were hard to overview, it was not possible to see how many notifications one specific model has had or how much that model had costed.

To give the reader a better idea of how the problem appears in the data, one example for manufacturer name and one for model names have been extracted from SAP and are shown in Table 5-2 and
Table 5-3 below.

Table 5-2 Example of the quality of the manufacturer names

Manufacturer	Model
SWITCHGEAR	DBR4
SWITCHGEAR/AEM MOD	DBR4
SWITCHGEAR/AEM MOD.	DBR4
SWITCHGEAR/AEM MODIFIED	DBR4

Table 5-3 Example of the quality of the model names

Manufacturer	Model
TAPLIN	300RCE/B200/10
TAPLIN	300RCE/B200/11
TAPLIN	300RCE/B200/2
TAPLIN	300RCE/B200/5
TAPLIN	300RCE/B200/6
TAPLIN	300RCE/B200/7
TAPLIN	300RCE/B200/8
TAPLIN	300RCE/B200/9

From these tables it is obvious how the many different names of the manufacturers and models will complicate interpretation of reports.

The start up date of the equipment was another parameter that was wrong. For a lot of equipment the date was the 1st of January 1900, which meant the equipment would have been 103 years old when this thesis was written. This is, even though some equipment can be really old, of course not correct. The reason for this rather strange date was the fact that some dates were stored as *null* in SAP and would therefore not be included in the extraction of data. These null values were therefore assigned a date to make it able to use the data connected to them, and to make them prominent they were put to 1900.

Another great number of equipment had 5th of July 1999 as their start up date, the amount of equipment starting up at the same day is a little bit suspicious. The reason for this was the transition from the old system, *Engarde*, to SAP. In Engarde very few start up dates were filled in which were, when moving the data, displayed as 5th of July 1999 in SAP and therefore had as a consequence that all data could not be used in the bathtub curve. This then would have resulted in a span of years that would not have any notifications at all together with a small gathering of notifications at age 103 for all type of equipment. Also the data for MTBN were not fully correct for some models.

Table 5-4 Number of Equipment divided into different start up dates

Start up date	Number of Equipment
1st of January 1900	8,482
5th of July 1999	27,382
All other dates	6,361
Total	42,225

The dates seemed to be a big source of the poor quality of the data. For the notification dates only dates greater then or equal to the 1st of July 1999 could be used, simply because there can not be found any more reliable data in the SAP database. This had impacts on, for instance, the bathtub curve, which uses very few data and does not get the chance to get a proper bathtub look.

Data for the object part, damage and cause of the notification for equipment gives one example where poor quality is due to laziness. When investigating notifications for equipment the service provider enters data for what object part that was affected, the damage and the cause of the notification. To avoid problems such as with the manufacturer and model names a system with codes representing all possible object parts, damages and causes is used. But still it is possible for the service provider to mess the data up and that is if they leave the field empty or consider data to be the unspecified null value. It is obvious that for a great number of notifications this has been the case since the cost and number of notifications for null values are a lot higher than for each and every of the other values.

Table 5-5 Distribution of Notifications per Cause Code

Cause Code	Number of Notifications
<null>	20,042
All other Cause Codes	16,536
Total	36,578

5.2 Software

In the start up phase of this project there was no access to the actual software to be used. This was dealt with later on, but still access was limited, and no administrator access was possible. Better access would have helped from a time consuming perspective.

BI Analyze was used a great deal in the start of the project which was good to get knowledge about how the HyperCube works, but unfortunately the results from BI Analyze can not be used on the Web. This means that some unnecessary time was spent on the calculation problems that occurred in BI Analyse. Instead Genio Designer was used to conduct complicated calculations and BI Report used for the reporting on the Web.

Problems during the work with the software turned up mainly within BI Report and Genio Designer. Some problems were solved while others at the time of writing still remains to be solved. The biggest problems within each of the software applications are described in the following sections.

5.2.1 BI Report

The programs within Hummingbird BI Suite worked unsatisfactory since they did not cooperate well, did not solve the tasks they were assigned and confusion was a fact when it was possible to solve one task in many different ways, with different results. The Hummingbird Support in this case did not give too much help, but they did however solve some problems that occurred with BI Suite. At the time of writing this thesis there are two bigger software problems still to be solved which have affected the end result of this project:

- Since many reports consist of different objects, a top 10 representation would be a good way to make the data easier to handle. The problem is that this only is possible in the tables, not in the graphs.
- The way exceptions are raised within the reports will cause an error when there is only one object in the report. The reason for this is that standard deviation is used, the calculation for σ, Equation 3-17, use involves a division by n-1 which is equal to zero when n is equal to one.

5.2.2 Genio Designer

Another issue causing more work than necessary has been the lack of specifications for the data model. As time has passed the model has become more complex considering what should be reported on, which has led to that the tables within the AMT Database having to be changed. This caused problems with the modules in Genio Designer, since the new tables had to be inserted and therefore changes the scripts in the modules if the structure of the tables have been changed.

If all desired parameters had been known from the beginning, the usage of tables and data sets within Genio Designer could have been more efficient since many models roughly use the same scripts. However, when a new parameter has come up, the old tables and modules already created has been kept in order not to mess up work already done. This might be an issue for future work, to clean up the project created within Genio Designer in order to slim down the amount of tables and variables used.

The efficiency issue does however not only depend on the dynamic size of the model. More tables than ideally necessary has been used because of certain weaknesses within Designer to join together tables in queries. Instead of making one big data set combining a number of tables, which had been the natural way in BI Query, the set has been broken down in to several steps requiring more tables and modules to get the proper result. At the time of writing, a limitation on how much code that can be written within one module and certain expressions has also caused some efficiency issues. The solution has been to split up the work in several modules, and might make the projects look a bit unstructured. The limited amount of code has also resulted in some sacrifice of the result. For instance certain modules will raise exceptions in the logging part of Genio Scheduler after have being run, the end result is however correct, which often is due to a division by zero. This can be avoided but would require additional code, which can not be fitted in the modules without having to create new ones.

5.3 Data

As mentioned earlier in this thesis the collection of data in the beginning was a big problem since the right location was hard to find due to that it was spread in many different tables within SAP and the names were hard to interpret. As with the software problems this was solved after a couple of weeks when the new AMT database had been built inside the ACMS database.

6 Future Work

During the work with the project, a number of problems and inefficient parts have been discovered, some of them were solved while others were left for the future. Hopefully they can be solved by an upgrading of software or spending an increased length of time on the project. However, no critical task was left undone, which means the future work does not have the highest priority. In this chapter some of the problems dealt with in chapter 5 are considered out of a future work perspective. The last part of the chapter also describes some of the new work within the area of automatic reporting that is being considered.

6.1 Solutions and Recommendations for Improvement of Data Quality

It is important to identify and correct problems with the data quality. For the problems found, which are listed above, some recommendations of solutions have been brought forward.

The problem with lack of data for specifications of functional locations (see section 5.1.1) could almost be solved completely with the help of GENIO and an extra Functional Location table. A module was created in GENIO that searched for the built section of the functional location, saves all data to be used and then populates all empty fields for functional locations under that built section in the new table. One drawback with this solution (so far) is that it does not work for functional locations that lack the data at the built section level, even though the data exists at some other levels. The recommendation is that the module for populating the new functional location table should be improved so that all empty fields for all functional locations would be filled. Another way of solving this could be to change how the data is represented in SAP; this is something that has to be discussed with the strategy engineers.

For the manufacturer and model names the problem is well known and work is for the moment being performed to clean the data within SAP. It has been reflected before, which is shown in (Gabb, 2001) where the problem is mentioned for *circuit breakers*. As a result of this further investigation was performed and the outcome of this was a report by Jamie Gabb (2003), which gives recommendations of how the names for all Powerlink Queensland assets should be changed. Recommendations are that the data should be changed according to the report and also include eventual additional manufacturers that have been added after the internal memorandum. To preserve the consistent data a procedure for how manufacturer names should be entered into SAP should be brought forth. These recommendations also apply to the model names, but here no investigation of all Powerlink Queensland assets concerning different model names has yet been performed. (Gabb, 2001; Gabb, 2003)

Measures have been taken to try to correct the start up date for all equipment, that is trying to change the start up date so that the date is transferred to the right field within SAP. This is not completed since there are still a number of dates that are the 1st of January 1900, and an even higher number of dates that are the

5th of July 1999. Recommendations are to continue this process of changing the dates to the correct field, and if the problem remains investigate the reason why.

The problem with maintenance service providers not filling in the notification codes still exists and older data has not been cleaned. Recommendations are that a procedure, explaining how and why the correct code should be entered, should be written and handed out to all maintenance service providers. Then after they have started to adopt the procedure the data should be cleaned, if this is done before handing out the procedure the data will most likely be distorted again.

6.2 Data Source

As discussed in section 5.2.2, the uncertain path followed at the beginning of the work with the project led to some amount of double work being performed. The result of this is that data for the generated reports is taken from both BI Query and Genio. From a maintenance point of view this is not ideal, instead one software package should be used for generation of the data for the reports, preferably Genio. However, the amount of code/queries written can then tend to increase, which is the price to pay for more efficient maintenance of the model.

6.3 Visual Aspect

According to section 0, there are some visual issues with the final data model. These include a top ten representation in the graphs of the reports and a limited amount of body text in the modules.

These issues seem to be due to limitations within the software applications, and hope is put into a solution via upgrading of the versions of the programs used. At the time of writing, system tests are being performed with newer versions and hopefully the problems should be solved soon.

6.4 Work Within Other Areas

During the work with SAP Notifications new areas for automatic reporting were explored. For example, similar data models have been created for FODs and diesel generator reporting. These are already being used extensively for asset monitoring and the goal is to include all data that is used for reporting in a data model. For instance reporting of data stored within the Outage System- Transmission Reporting and Co-ordination Database (OSTRAC) would be suitable to automatize.

7 Conclusions

This final chapter aims to provide the reader with conclusions about what has been achieved in the previous chapters and how the purposes of the thesis have been fulfilled. It will also discuss the final results together with advantages and disadvantages of the new technology.

The need for more efficient maintenance and asset monitoring is a result of deregulation and the demand to be more cost efficient. The right way to achieve this is not specified but a general approach is to make sure that data stored within the corporate database is being used for feedback on maintenance performed, both routine and breakdown. This was the basic goal for this thesis work, to make the data easily available for the strategy engineers in the form of automatically generated reports for SAP notification data. The reporting should also include raising exceptions for different parameters such as *Normalised Cost and Notifications per Model*. Preferably, all data of interest should also be gathered and stored in a multi-dimensional HyperCube. This allows the users to view all possible data by drilling up and down through the parameters.

One of the risks with creating an automatic model to perform analysis is that the reliance on peoples memory and natural feeling for the equipment might be reduced, instead trust is put into the raw data behind the figures in the model. The gut feeling that has been used before, and still is used, is often great for decision-making and a source of historical trends; people know how things used to be like, which the raw data might not reveal.

The software application Hummingbird BI suite was used for the creation of the data model for automatic retrieving of information from SAP. One of many possible solutions found allowed the strategy engineers to modify the reports to suite their own purposes, which was not an acceptable solution since the strategy engineers view the same reports and one modification performed by one user might not suite the other users. Therefore a data model with a number of standardised reports was created and published to a corporate repository. This allows the strategy engineers to access on-line reports via the web. The layout of the reports is fixed and the strategy engineers can only choose what data to view.

The data in the reports and hypercubes are updated on a daily basis which means that the strategy engineers at any time can check the reports with up to date data. The reports contain either a graph and a data table or only a data table while the HyperCube in all cases contains both a data table and a graph.

The idea of one single HyperCube had to be abandoned. This was due to two problems:

- The amount of data became too large, making the cube an inefficient tool
- The software application did not allow some of the data to be displayed correctly

Instead one cube with a limited number of dimensions was created. It allows users to view the data in a raw state and manipulate it to suit their purposes at the time. It is a tool to use in conjunction with the reports, allowing a clearer picture to be established.

In the reports, exceptions are raised for values lying outside a predefined threshold and these are highlighted with red in the data tables. The thresholds are dynamic, which means that they are adapted to the specific data loaded into the reports. This is fundamental since the reports can be used to view different types and ranges of data.

To set the thresholds, the distribution of the data has to be decided. This is fundamental for finding the appropriate average and standard deviation, parameters used for the calculation of the thresholds. If the average and standard deviation was calculated and used without knowing the distribution, the number of raised exceptions would deviate from the truth. This deviation would mean that the feedback on the maintenance to the strategy engineers would be incorrect and in worst case lead to even more notifications being generated.

The above mentioned issue is partly addressed to the AMT staff for future work. Checking of the distribution has to be performed once every two years, and is the only manual part in the automatic process for generating reports and raising of exceptions. For the moment it seems hard to make this step automatic, since in would require an intelligent software that can recognise different distributions and set thresholds depending on the distribution.

To ease the work for the strategy engineers, e-mails generated by the software application Hummingbird Genio Design are sent out on a regular basis for each area and technical object type. The e-mails contain, where applicable, the number of exceptions raised for each standardised report and a hyperlink pointing direct to the report in question. This serves as high level information compared with the reports. The e-mails give a good overview of the plant, and where special attention has to be focussed, instead of searching through every standardised report for every technical object type etcetera.

To be able to provide the strategy engineers with fresh data every day, processes gathering data from the ACMS and putting into the data tables has to be run every day. For this purpose Hummingbird Genio Scheduler has been used. The software application schedules and executes certain scripts within Designer.

SAP has only been in use for four years to collect and register data, which makes it a relatively new technology. It can show some of the expected infant problems occurring when new techniques are used. This means poor quality of data to some extent and a limit to what can be achieved with it. For instance all notifications raised before 1999 when SAP was commissioned can not be used since there is no reliable data in the SAP database. This causes problems, for example the bathtub curve, and when data for years before SAP is trended over time. The problem will be solved in the long term since more and more data naturally will be added to the database. When the strategy engineers start using the data model, the quality of data will also increase due to feedback to modify and correct the data in SAP. However, problems like this could appear in the future if, for example, Powerlink chooses to change the corporate database system.

The data model will be a suitable tool for Powerlinks maintenance and asset monitoring in the future. The question is however whether the data model is an effective tool compared to other software applications available on the market. As seen from the investigation performed by the authors in section 2.2, no other leading products on the market today seem to have the same possibilities as SAP Notifications to automatically generate reports and send reminders to the strategy engineers. However, the model lacks the feature of giving advice about maintenance procedures and trend analysis that many other typical maintenance software applications have. This might become a disadvantage in the future, when other applications catch up with SAP Notifications within the realm of automatic reports and reminders. When new powerful software applications are presented on the market, comparisons with the data model have to be performed to check whether SAP Notifications is the most suitable tool available for Powerlink.

Bibliography

Bannister A., Powerlink Queensland, “Private Communications”, Brisbane, October 2003

BI Cube Creator™ - User’s Guide [User manual], Hummingbird Ltd, Toronto, Canada, 2002

BI Query™ Data Models – User’s Guide [User manual], Hummingbird Ltd, Toronto, Canada, 2002

BI Query™ Reports - User’s Guide [User manual], Hummingbird Ltd, Toronto, Canada, 2002

BI Query™ Queries - User’s Guide [User manual], Hummingbird Ltd, Toronto, Canada, 2002

BI Web™ - User’s Guide, Hummingbird Ltd [User manual], Toronto, Canada, 2002

Childs Mark, “Maintenance management trends”, Maintenance Journal, vol.14:4, pp. 4-11, October 2001

Datastream, 2004-02-09
http://www.datastream.net/industry/automotive_F796CE072C5D4D18A5088472E5147D19.asp

Gabb J., “Evaluation of Circuit Breaker Maintenance”, Document Ref. AMTREP003, Brisbane, 3rd of April 2001.

Gabb J., “Manufacturer Naming convention for Powerlink Assets”, Document Ref. AMTMEM002, Brisbane, 11th of April 2003.

GE Industrial Systems, 2004-02-09
http://www.gehealthcaresolutions.com/eam.htm

Gellings, C.W, “Power delivery system of the future”,
Power Engineering Review, IEEE , Volume: 22 Issue: 12 , Dec. 2002

Genio Designer – User’s Guide [User manual], Hummingbird Ltd, Toronto, Canada, 2002

Hastings Nicholas A J, *Asset Management and Maintenance*, Queensland University of Technology (School of Mechanical, Manufacturing and Medical Engineering), Brisbane, 2000

Kelly Anthony, *Maintenance Planning and Control*, Butterworths, p.36, 1984

Kelly, Anthony, *Maintenance And Its Management,* Ashford Press Ltd, 1989

Krieg Terry, "Transformer monitoring: Key to cost-effective asset management", Electrical World T&D, 2001

Levitt Joel, *Complete Guide to Preventive and Predictive Maintenance*, Industrial Press, New York, 2003

Mainman, 2004-02-09 http://www.kciamericas.com/mainman.htm

Maximo 4i, 2004-02-09 http://www.gocfi.com/facility-management-technology/maximo4i_features.htm

Moller I., Becconsall B., "Results of Infrastructure Development Benchmarking U.M.S Group - 1998", Internal Report, Powerlink Queensland, 21st of October 1998.

The SAP fan club, 2003-11-14, http://www.sapfans.com/

Sharp, Brian, "EEP222 Maintenance of Electricity Supply Systems"

Course notes and presentation, Queensland University of Technology, July 2002

Thurston Michael, Lebold Mitchell, "Standards developments for condition-based maintenance systems", 55th Meeting of the Society for Machinery Failure Prevention Technology, April, 2001

Transmission & Distribution World article, "Economic Maintenance Strategies for the Future- The program considers condition based rather than time-based maintenance", 2004-02-09, Internet: http://tdworld.com/ar/power_economic_maintenance_strategies/, 1 Feb 1997.

Trendmaster Pro, 2003-10-13 http://www.bently.com/articles/2003ORBIT/Trendmaster_Pro.pdf

Plant Monitoring (course notes), 2003-10-13 http://www.eng.man.ac.uk/mech/courses/em352/pm3-3.html

Wharf, Chris, Powerlink Queensland, "Private Communications", Brisbane, October 2003

Appendix

A Glossary of Terms

Ad Hoc Query	Allows the user to set up his own query for retrieving certain data
Cause Code	Code that refers to the cause involved with the notification raised
Circuit Length	Total length of transmission line, all phases, from point A to point B within one functional location
Damage Code	Code that refers to the damage/problems involved with the notification raised
Equipment	Individual, physical object that is to be maintained as an autonomous unit
Easement Maintenance	Maintenance involved with keeping transmission lines and structures free from vegetation
Feeder	Transmission line or underground cable carrying power from a substation
Functional Location	Organisational unit that structures the maintenance objects of a company according to functional, process-oriented or spatial criteria
HyperCube	A tool for storing and viewing multi-dimensional data
Notification	Data record, with which a user informs the Plant Maintenance department about a particular occurrence on an item of plant
Priority	Criterion for urgency of work items
Technical Object	A component within a technical system, on which a maintenance task is to be performed
Plant Maintenance	Measures taken to maintain operational systems in working order (for example machines, production installations)
Planner Group	Group that manages the maintenance within a specific region and technical area
Script	A piece of code to be executed. Performs certain, well defined tasks
Route Length	Length of transmission line from point A to point B within one functional location

B User Manual - Data Model

The user manual *AMTPRO066 Procedure for using SAP Notifications Data Model and the HyperCube* is attached to give the reader more understanding of how the data model works.

Asset Monitoring Team - Procedure

Procedure Name: **AMTPRO066 Procedure for using SAP Notifications Data Model and the HyperCube**	Procedure No.: **AMTPRO066**	
	Revision Date **20** / **11** / **2003**	Revision No. **1**

AMTPRO066 Procedure

for using SAP Notifications Data Model and the HyperCube

5					
4					
3					
2					
1					
0	20/11/03	ORIGINAL DOCUMENT	J.A/A.J		
REV	DATE	REVISION DESCRIPTION	AUTHOR	CHECKED	APPROVED

Asset Monitoring Team - Procedure

Procedure Name: **AMTPRO066 Procedure for using SAP Notifications Data Model and the HyperCube**	Procedure No.: **AMTPRO066**	
	Revision Date **20 / 11 / 2003**	Revision No. **1**

Table of Contents

Asset Monitoring Team - Procedure

Procedure Name: **AMTPRO066 Procedure for using SAP Notifications Data Model and the HyperCube**	Procedure No.: **AMTPRO066**	
	Revision Date **20 / 11 / 2003**	Revision No. **1**

Open the data model

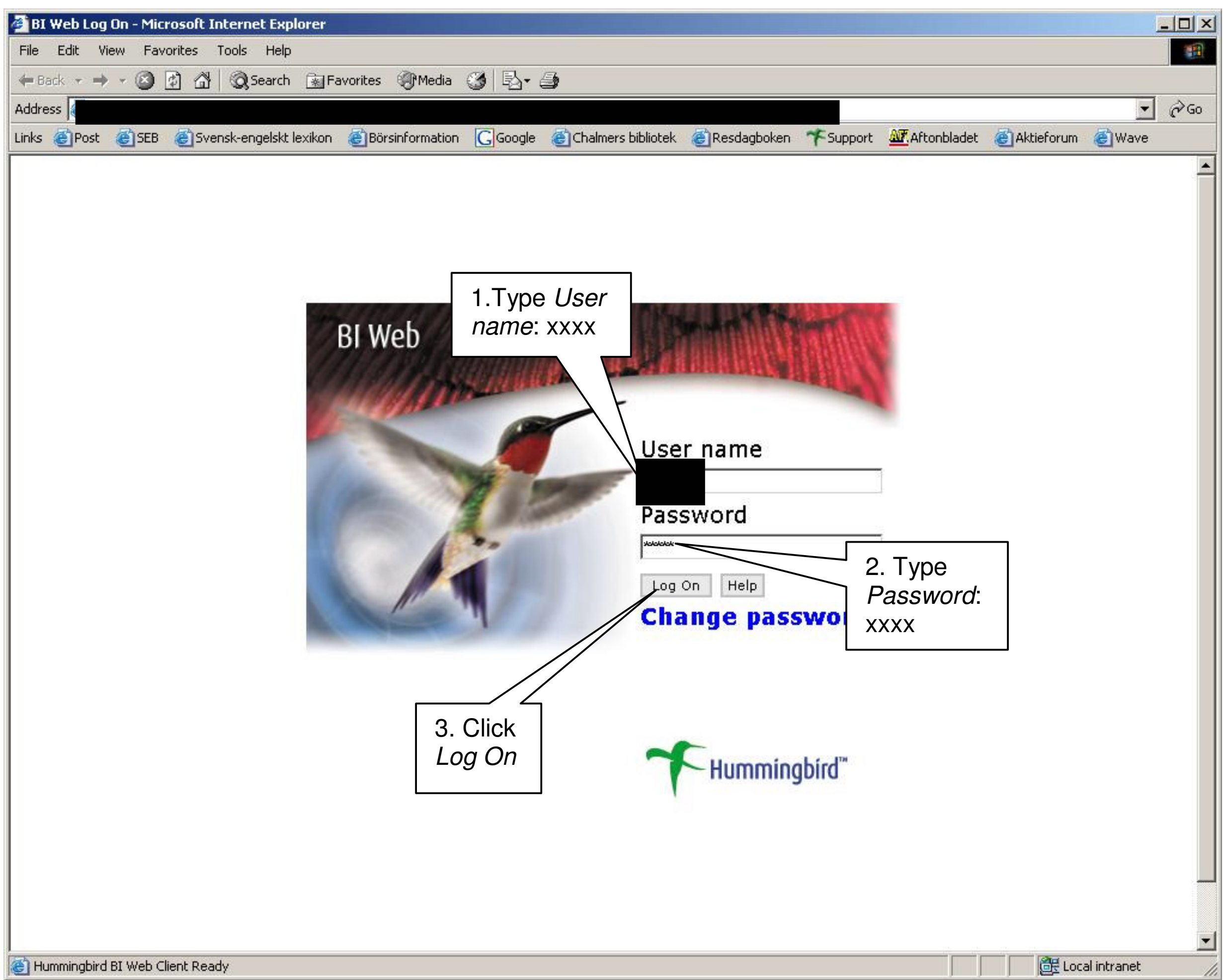
The *SAP Notifications* data model is accessed by opening up *BI Web*. Open *BI Web* by typing the following command or click the shortcut.

http://xxx.xxxxxx.xxx /

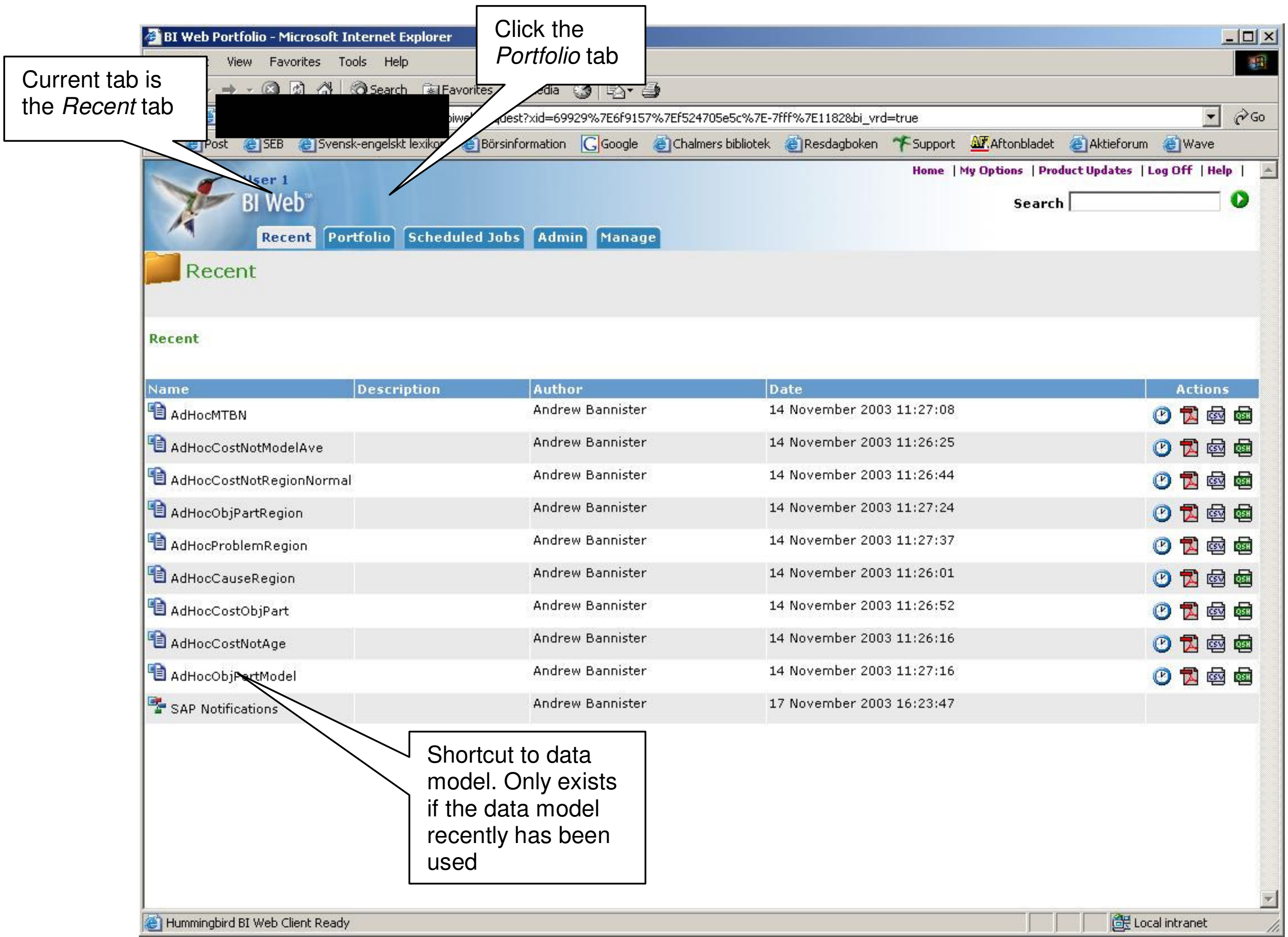
Use the following login data:

User Name: xxxx (to xxxx)

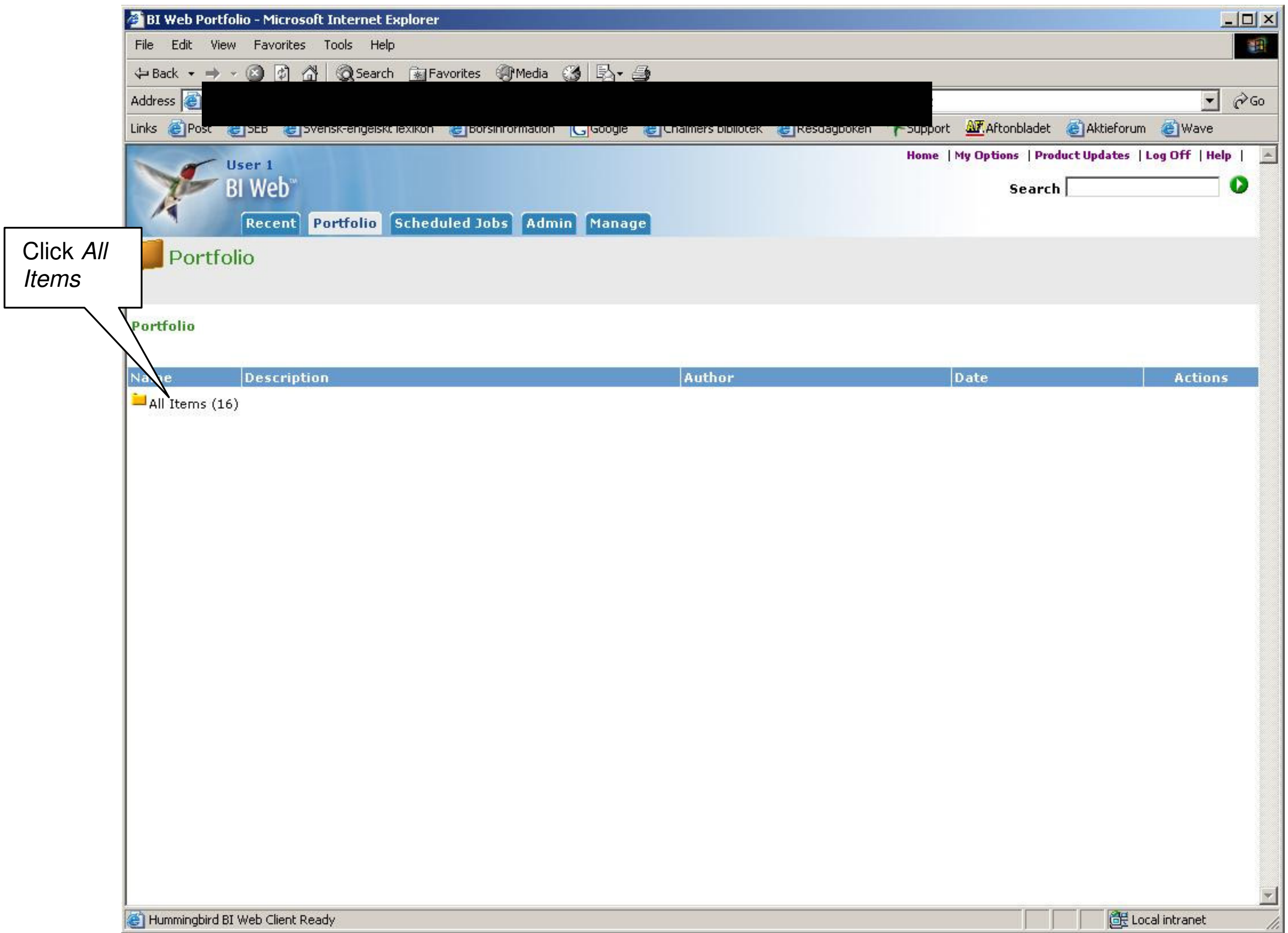
Password: xxxx (to xxxx)



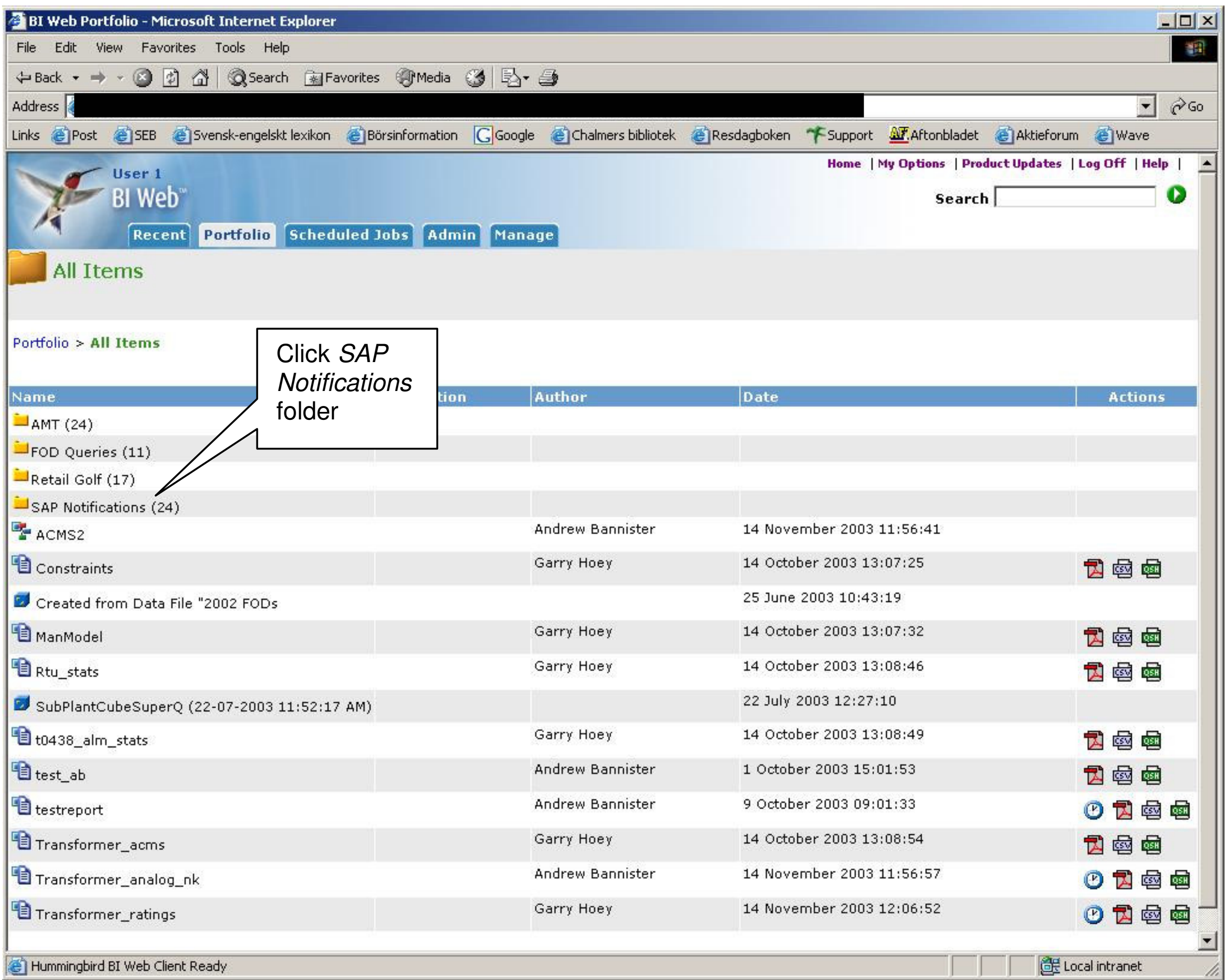
This will take the user to the *Repository* window and the *Recent* tab. The *Recent* tab contains the most recent used objects. Click the tab named *Portfolio*.



Click the folder named *All Items* to open up a window with all objects in the repository.

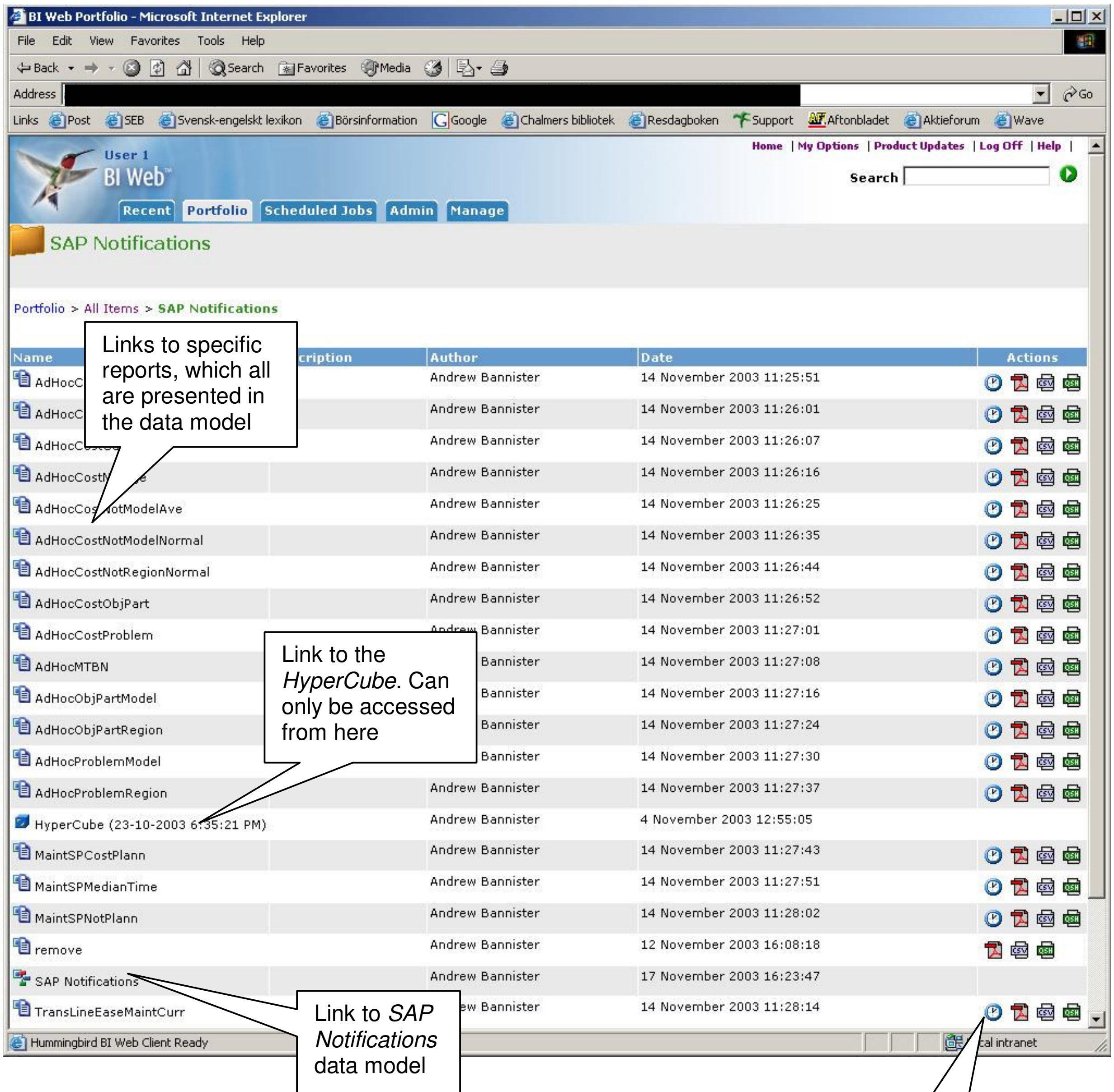


Click on the *SAP Notifications* folder to open up a window with all relevant objects for the data model.



Procedure Name: **AMTPRO066 Procedure for using SAP Notifications Data Model and the HyperCube**	Procedure No.: **AMTPRO066**	
	Revision Date **20 / 11 / 2003**	Revision No. **1**

In this window all objects relevant to the *SAP Notifications* data model are shown. Click the row with *SAP Notifications* to open up the data model.



The -symbol indicates that the report will be refreshed on load

- Symbol for report

- Symbol for *HyperCube*

- Symbol for data model

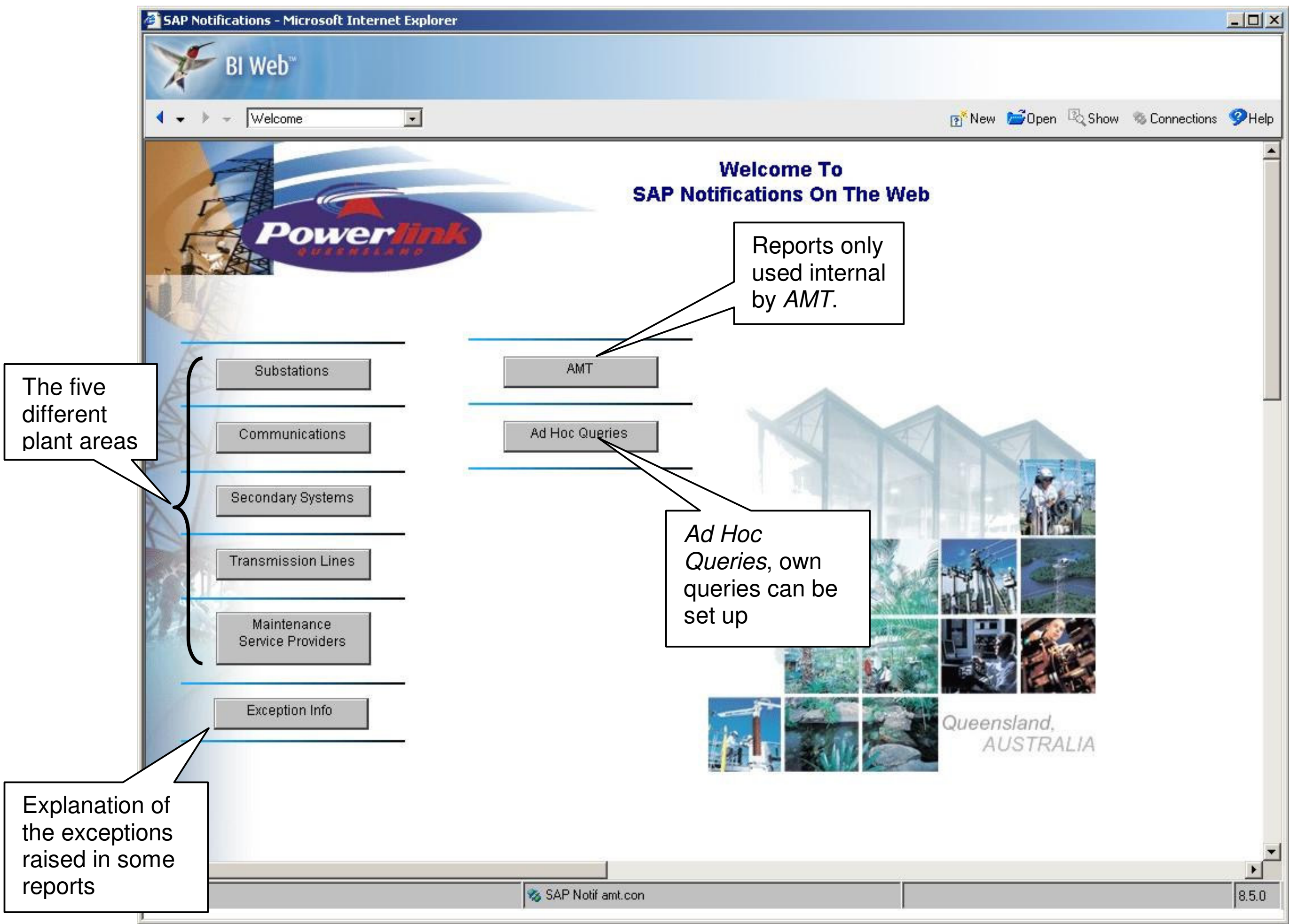
Procedure Name: **AMTPRO066 Procedure for using SAP Notifications Data Model and the HyperCube**	Procedure No.: **AMTPRO066**	
	Revision Date **20 / 11 / 2003**	Revision No. **1**

Work with the data model

The data model can be used in two ways, either to use the predefined reports for different areas or to do your own ad hoc queries. The two different ways will be described in the following sections.

Reports

The welcome window of the data model is shown below. This is the first window that opens up when the data model is opened.

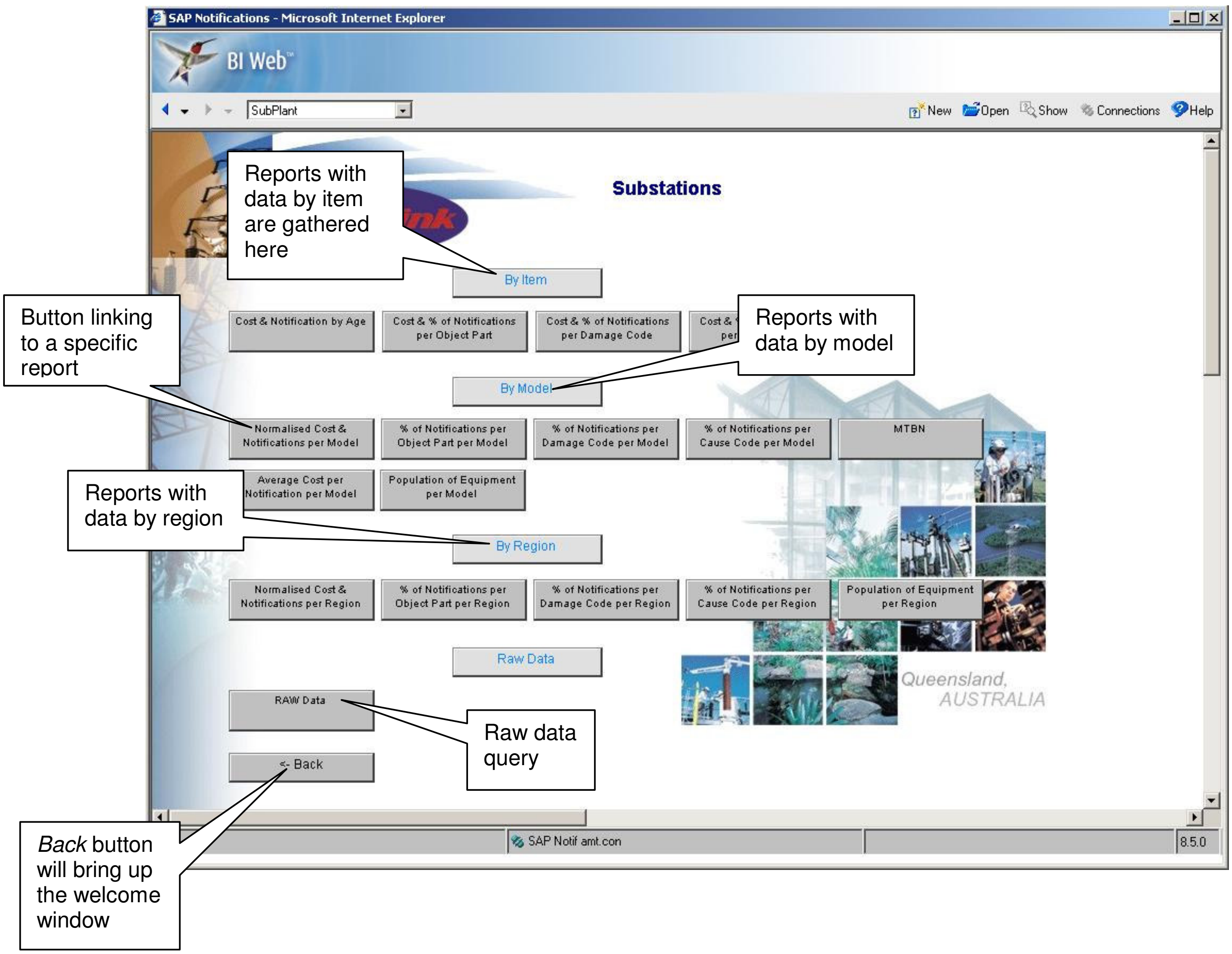


From the welcome window the user can choose to look at reports for five different areas (*Substations, Communications, Secondary Systems, Transmission Lines* and *Maintenance Service Providers*) by just a couple of mouse clicks, or create their own ad hoc queries. To enter an area click the corresponding button.

Asset Monitoring Team - Procedure

Procedure Name: **AMTPRO066 Procedure for using SAP Notifications Data Model and the HyperCube**	Procedure No.: **AMTPRO066**	
	Revision Date **20 / 11 / 2003**	Revision No. **1**

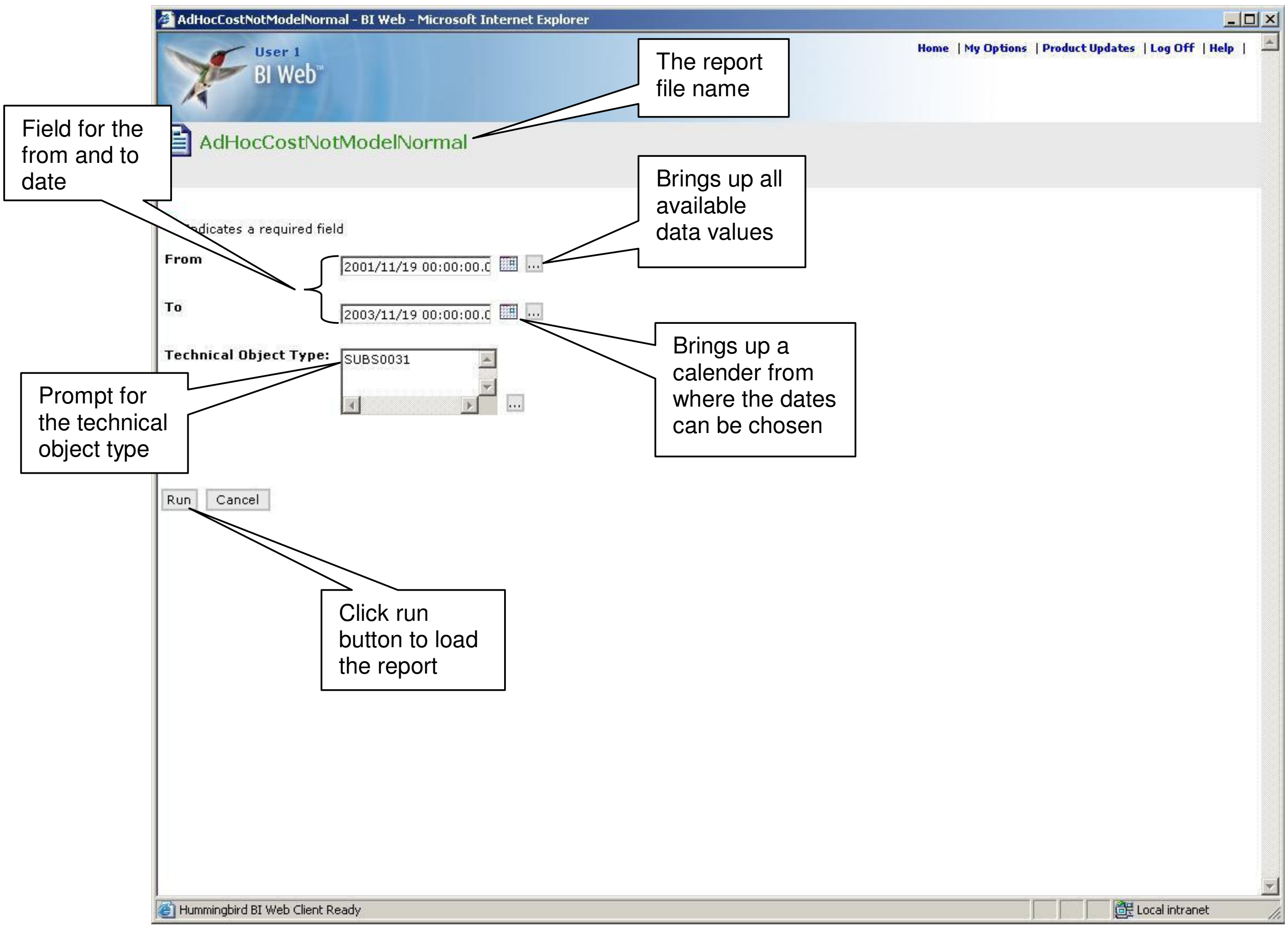
The report window lists all different reports concerning the chosen area, in this case the user entered the *Substations* reports window. By clicking a button the user will be prompted to enter some values.



Asset Monitoring Team - Procedure

Procedure Name: **AMTPRO066 Procedure for using SAP Notifications Data Model and the HyperCube**	Procedure No.: **AMTPRO066**	
	Revision Date **20** / **11** / **2003**	Revision No. **1**

This prompt will appear when the user presses the button called *Normalised Cost & Notifications per Model*. Enter the preferred value or leave the prompts empty, then click run to create the report. Leaving a prompt empty will bring up all values, for example for regions this would bring up data for all regions.



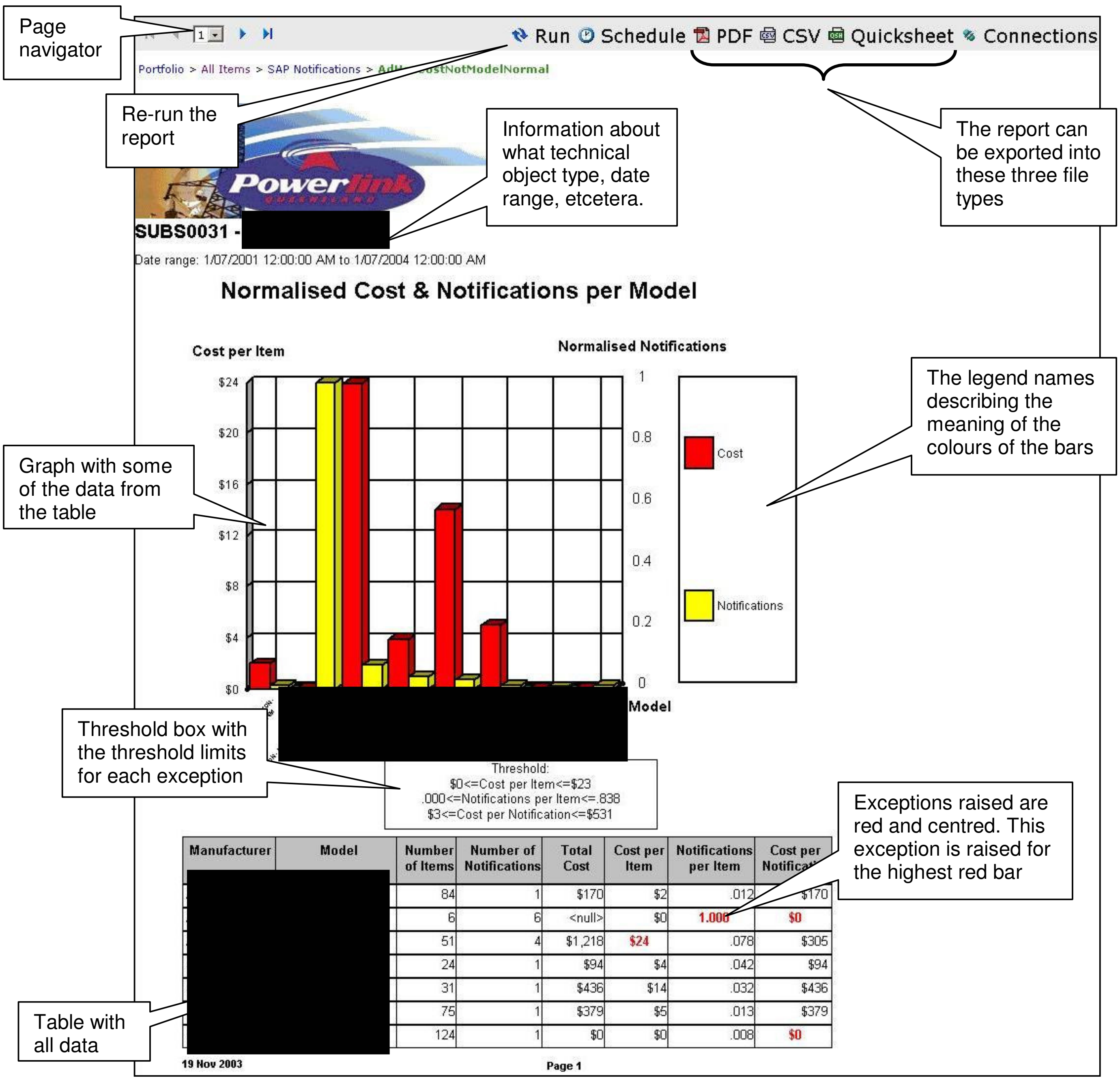
 - Symbol for calendar

 - Symbol for data values

Asset Monitoring Team - Procedure

Procedure Name: **AMTPRO066 Procedure for using SAP Notifications Data Model and the HyperCube**	Procedure No.: **AMTPRO066**	
	Revision Date **20** / **11** / **2003**	Revision No. **1**

The report will take a few seconds to load, normally around ten seconds. But this depends on the overall load of the system. An example a report is shown below, the report shown is *Normalised Cost & Notifications per Model*. If the report consists of more than one page the user can chose to look at other pages using the page navigation in the upper left corner. The report can be sent to a PDF, CSV or Quicksheet file by clicking the corresponding button.



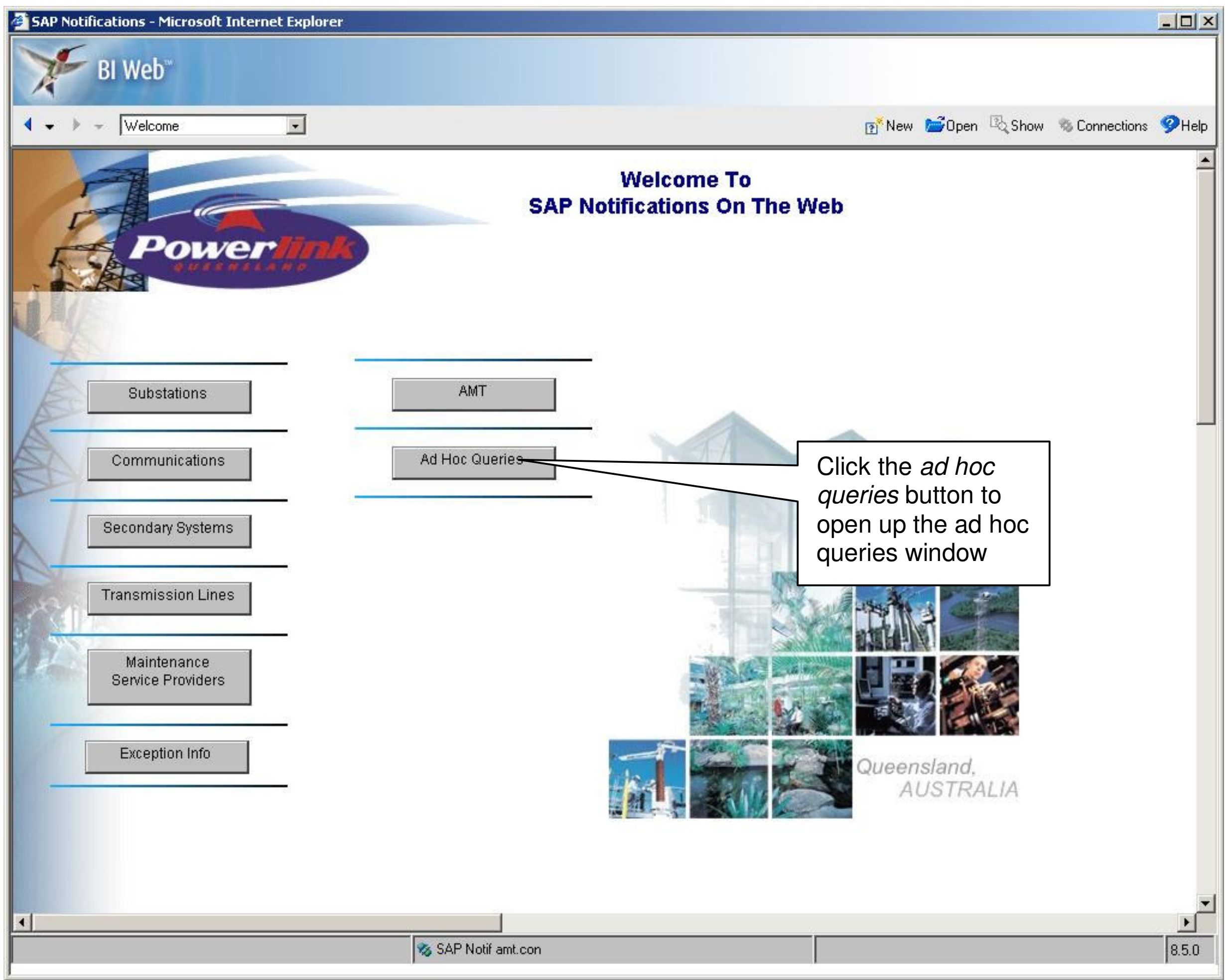
Manufacturer	Model	Number of Items	Number of Notifications	Total Cost	Cost per Item	Notifications per Item	Cost per Notifica
[redacted]		84	1	$170	$2	.012	$170
		6	6	<null>	$0	1.006	$0
		51	4	$1,218	$24	.078	$305
		24	1	$94	$4	.042	$94
		31	1	$436	$14	.032	$436
		75	1	$379	$5	.013	$379
		124	1	$0	$0	.008	$0

Asset Monitoring Team - Procedure

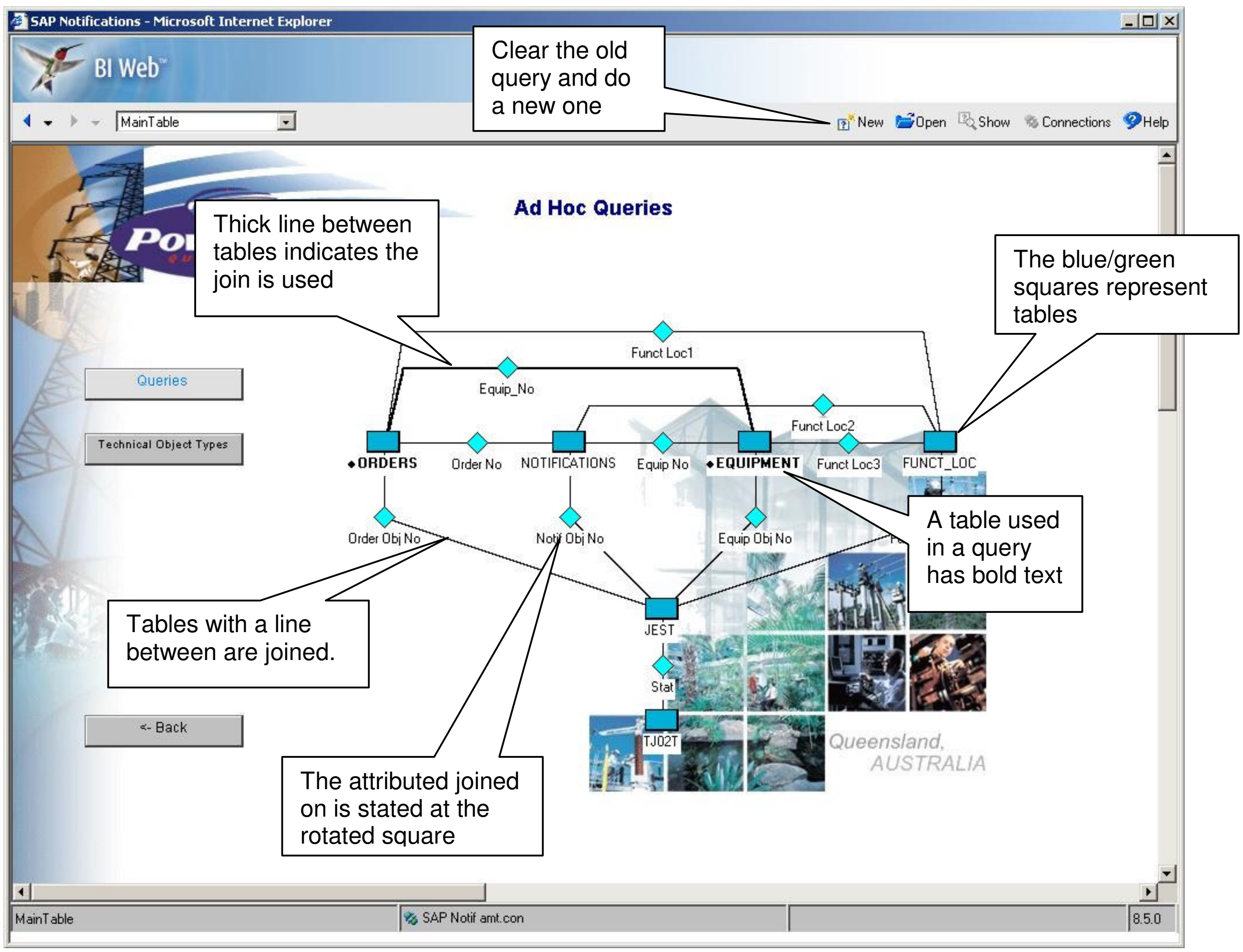
Procedure Name: **AMTPRO066 Procedure for using SAP Notifications Data Model and the HyperCube**	Procedure No.: **AMTPRO066**	
	Revision Date **20 / 11 / 2003**	Revision No. **1**

Ad Hoc Queries

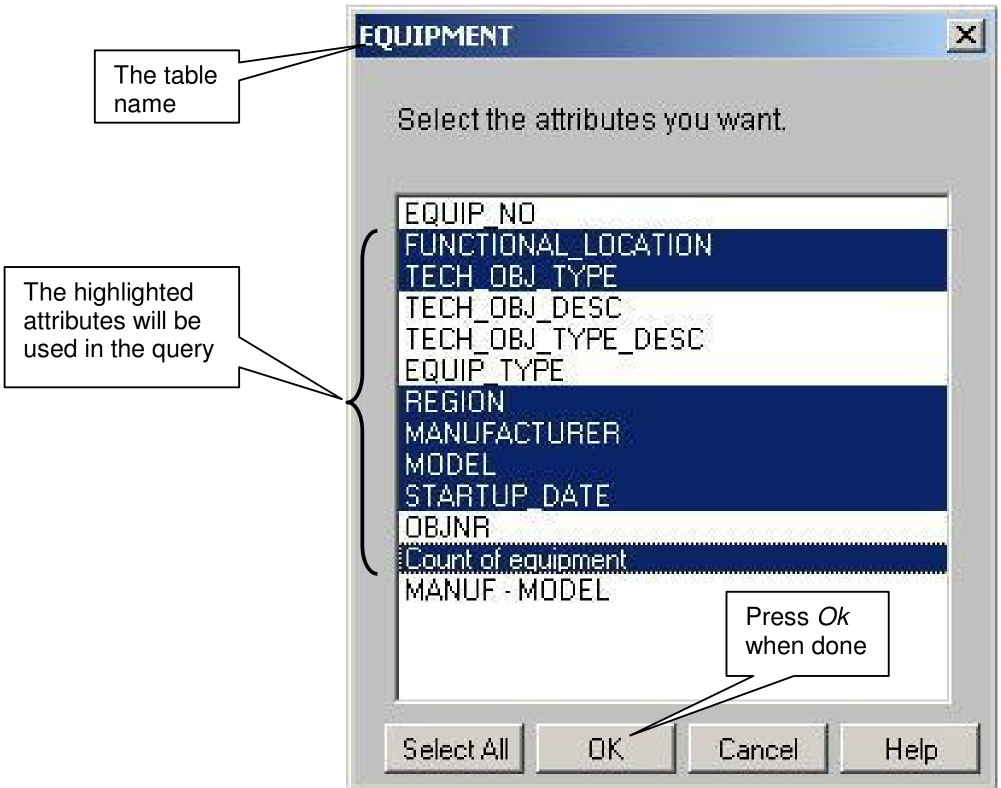
Instead of looking at reports in different areas, the user can create their own queries. This is useful when the data can not be found in the reports. The ad hoc window is accessible from the *SAP Notifications* welcome window.



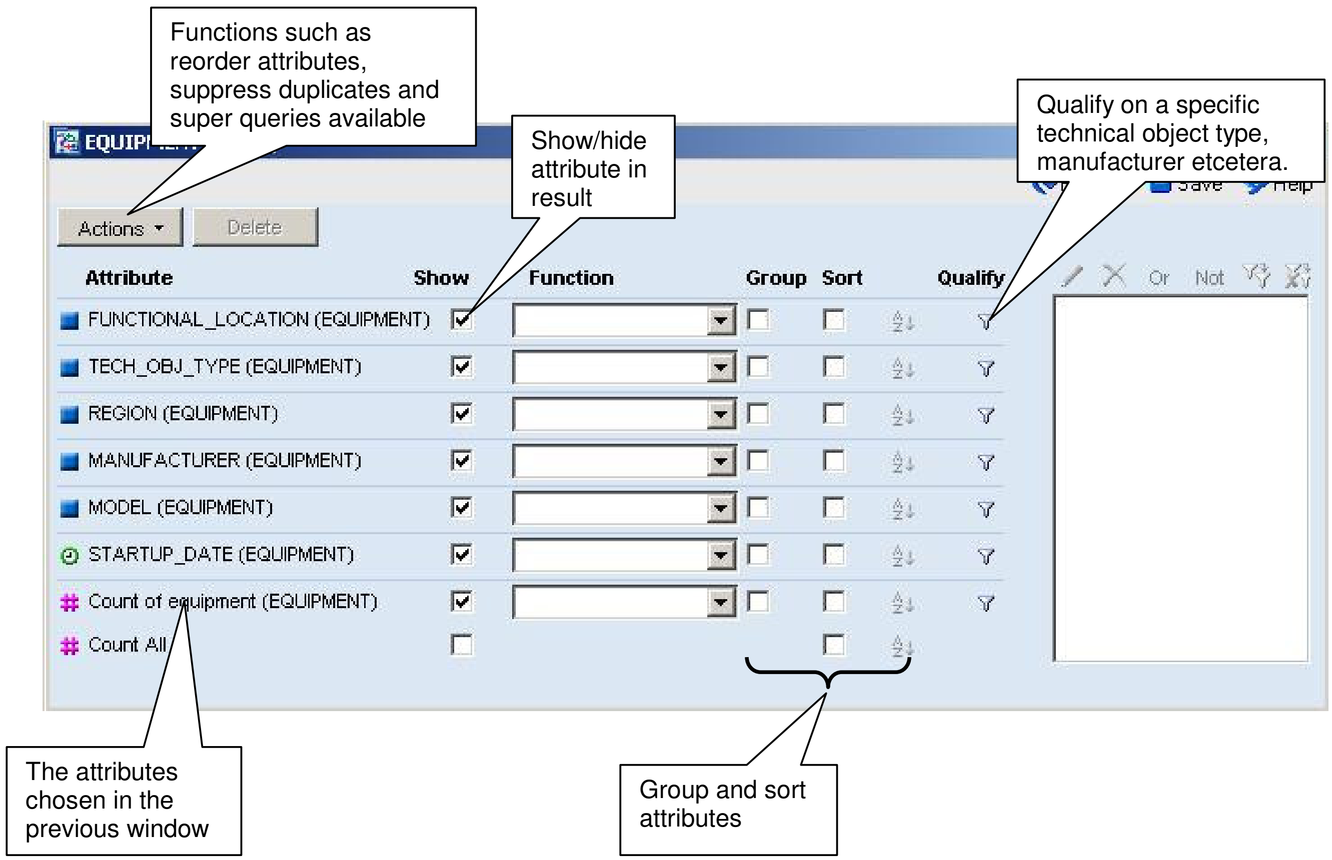
Pressing the *ad hoc queries* button will open up the window below. From here different queries can be created to achieve the necessary data. Double clicking a table (blue/green square) brings up all attributes available in that table that the user can choose to look at. To all these attributes functions can be applied, they can be grouped, sorted or qualified. The different tables are joined together on different attributes, this is seen as thin lines between the boxes. The data from one table can be joined with other tables simply by choosing a new table and adding some attributes from that table. The joined tables will be marked with a thicker black line in between.



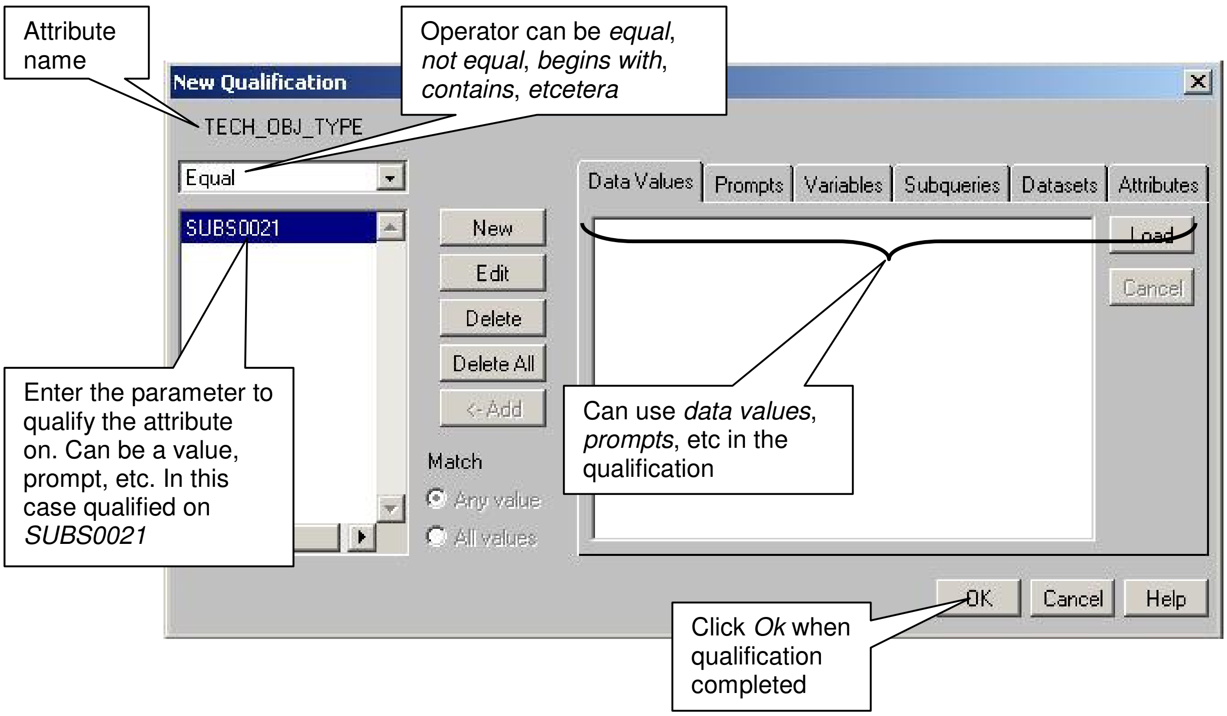
By double clicking a table the possibility to choose attributes from the selected table appears. The attributes shown below are available in the equipment table.



When the attributes have been chosen the following window will appear. This window allows some processing of the attributes and also the attributes can be qualified here.

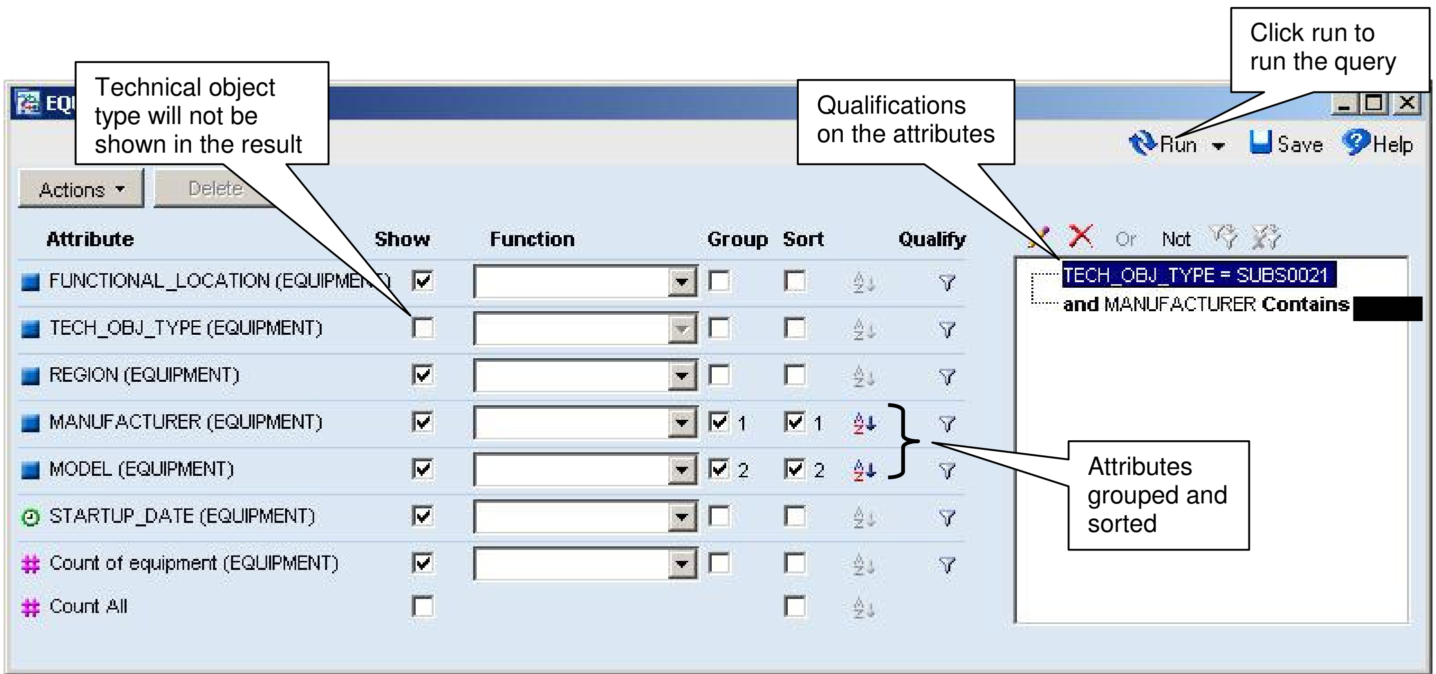


By choosing to qualify on a specific attribute the qualification window opens up. In this window a specific attribute can be qualified in a great number of ways.



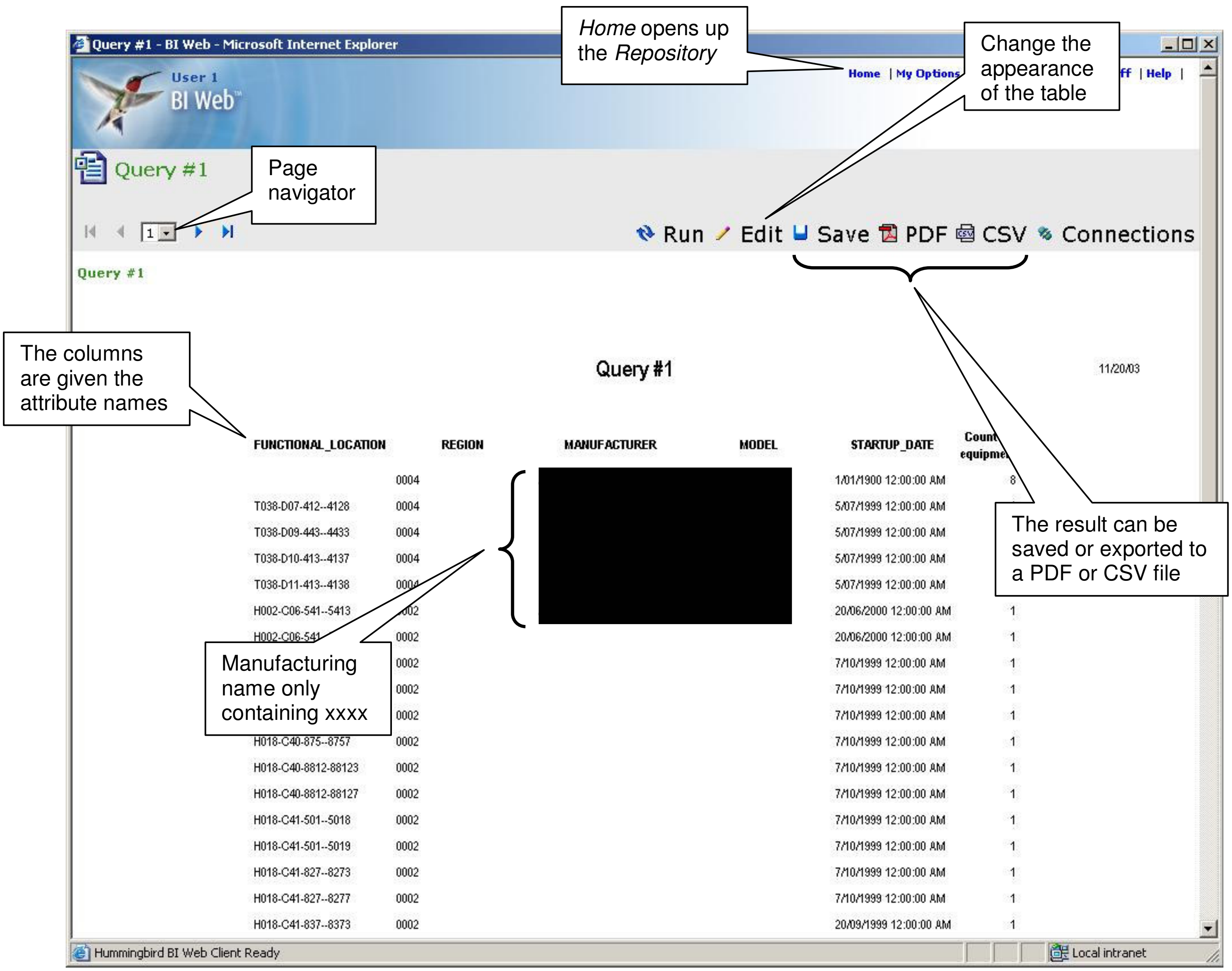
The attribute *TECH_OBJ_TYPE* is qualified so that the result when running the query will contain only items where the technical object type is equal to *SUBS0021*. Another more dynamic way to do this would have been to use a prompt where the technical object type is to be typed in every time the query is run. This would give the opportunity to use the same query for many different technical object types without much work.

After the qualification has been set up the user will come back to the previous window. The qualification set up will be visible in the text box to the right. In this case a qualification on manufacturer has also been done.



This query will only bring back data where the technical object type is equal to *SUBS0021* and the manufacturer name contains xxxx. The technical object type will not be visible in the result. The data will be grouped and sorted on manufacturer and model names

The result from the above query set up is shown below. The time it takes to retrieve a result varies a lot depending on how the attributes are qualified and if many tables are joined.



The appearance of the result table can be changed in an *edit* window accessed by pressing the *edit* button. The style of the text can be changed, colours, etc can be added. The data values can be further processed by loading the values into Microsoft Excel. All values are not printed in the grouped columns, only the first row contains the value. This is seen in the manufacturer and model columns. But the values will of course appear when loading the data to Microsoft Excel.

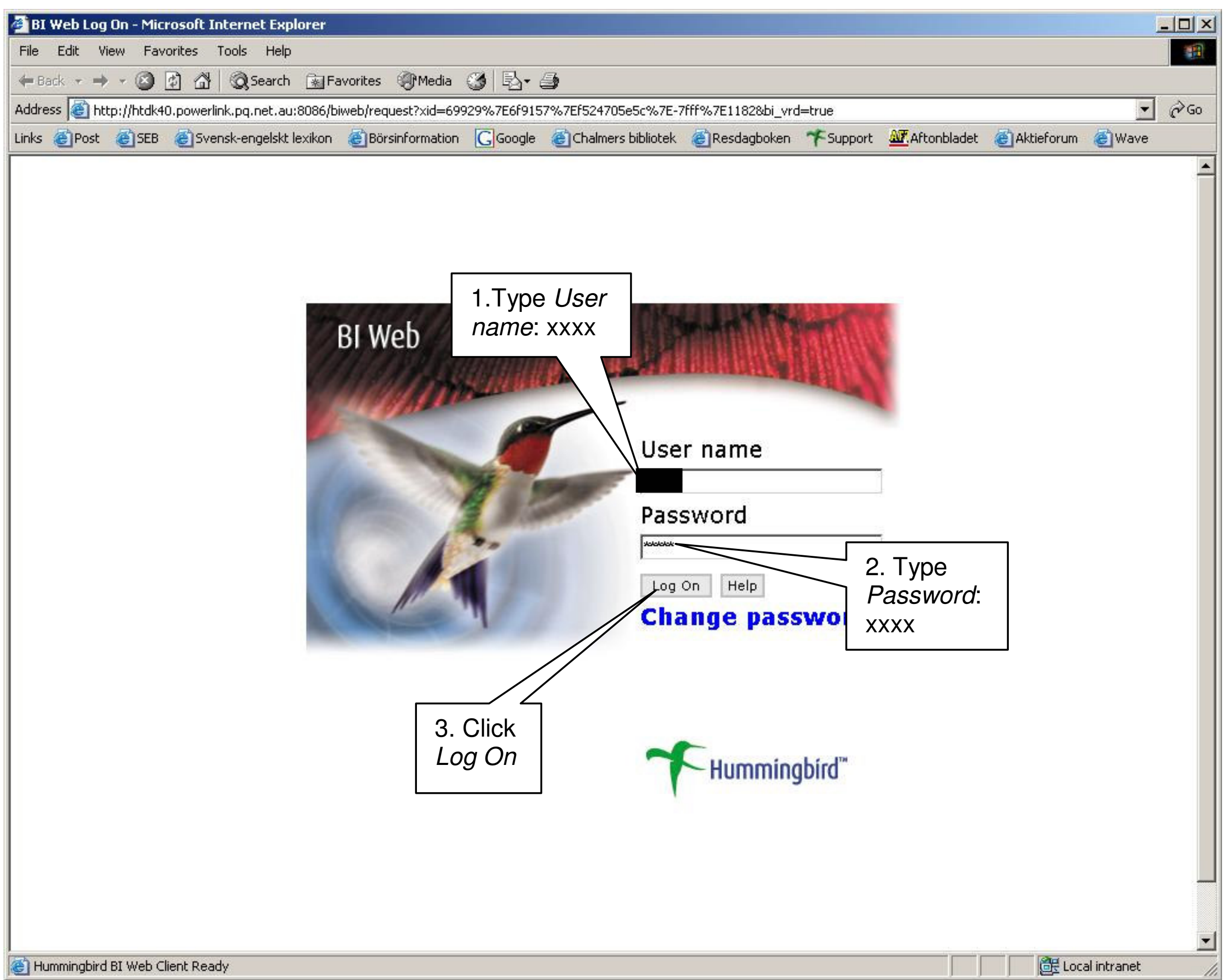
Open the HyperCube

The *HyperCube* is accessed by opening up *BI Web*. Open *BI Web* by typing the following command or click the shortcut.

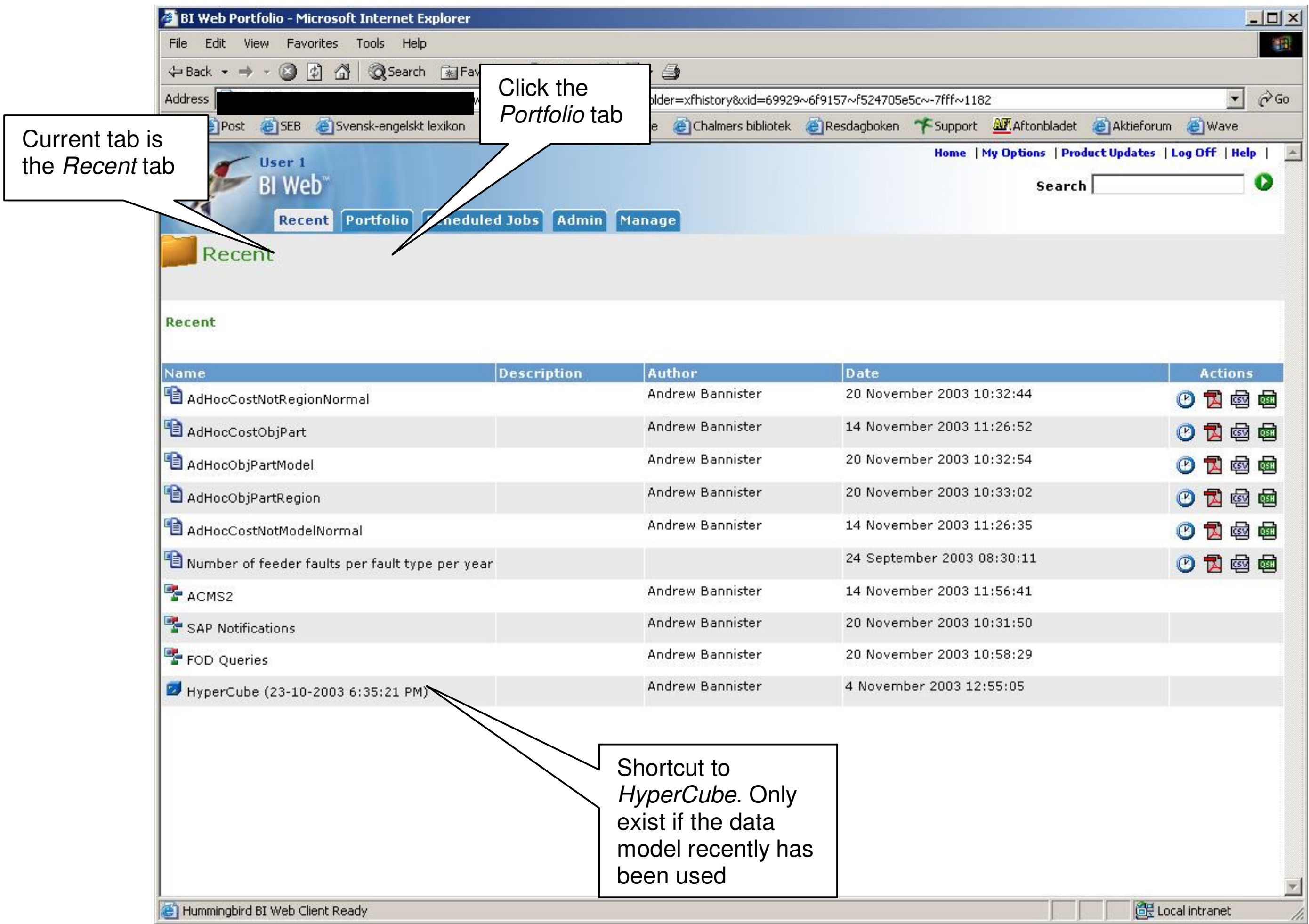
http://xxx.xxxxxx.xxx/

Use the following login data:

User Name: xxxx (to xxxx)
Password: xxxx (to xxxx)

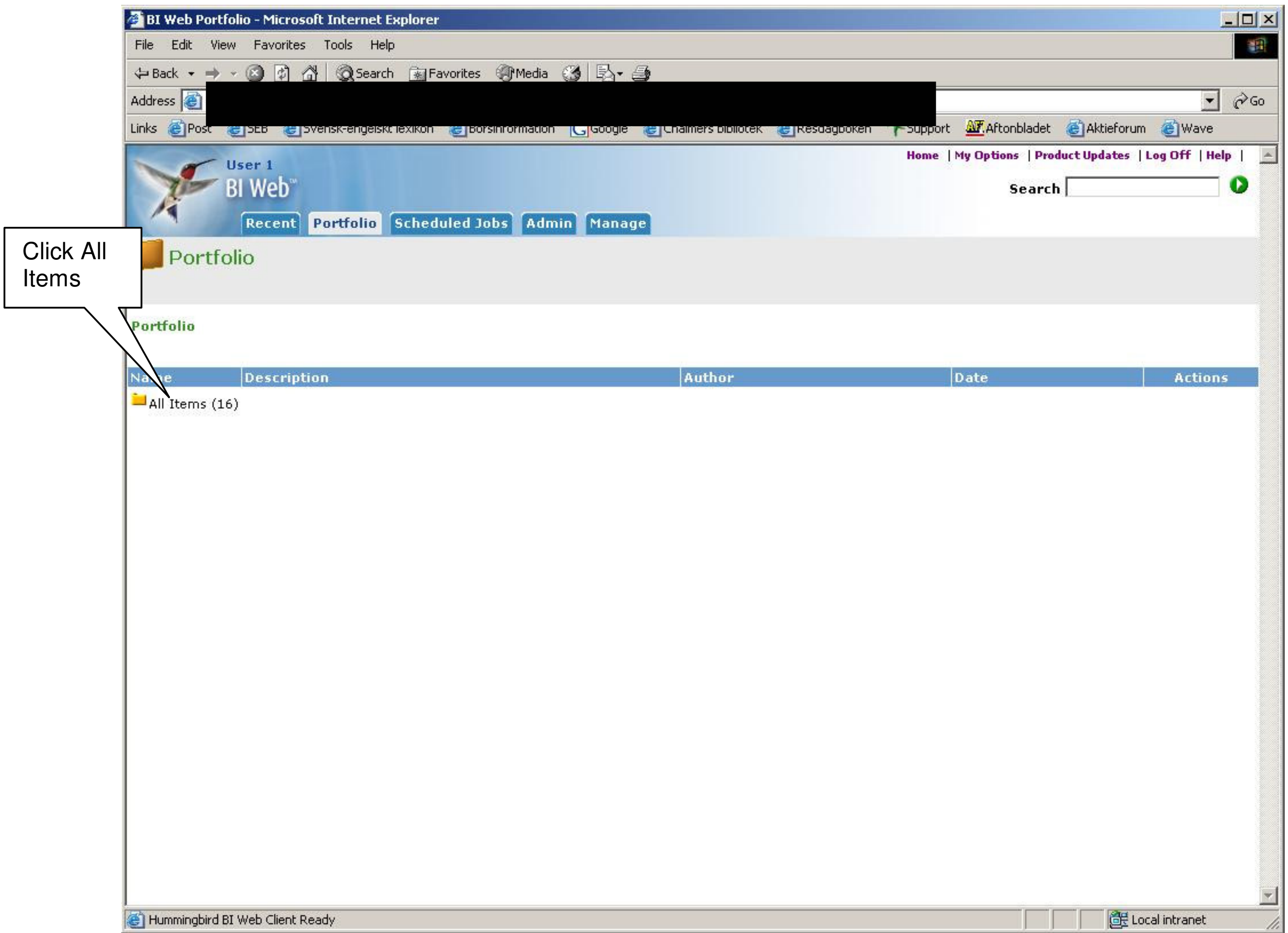


This will take the user to the *Repository* window and the *Recent* tab. The *Recent* tab contains the most recent used objects. Click the tab named *Portfolio*.

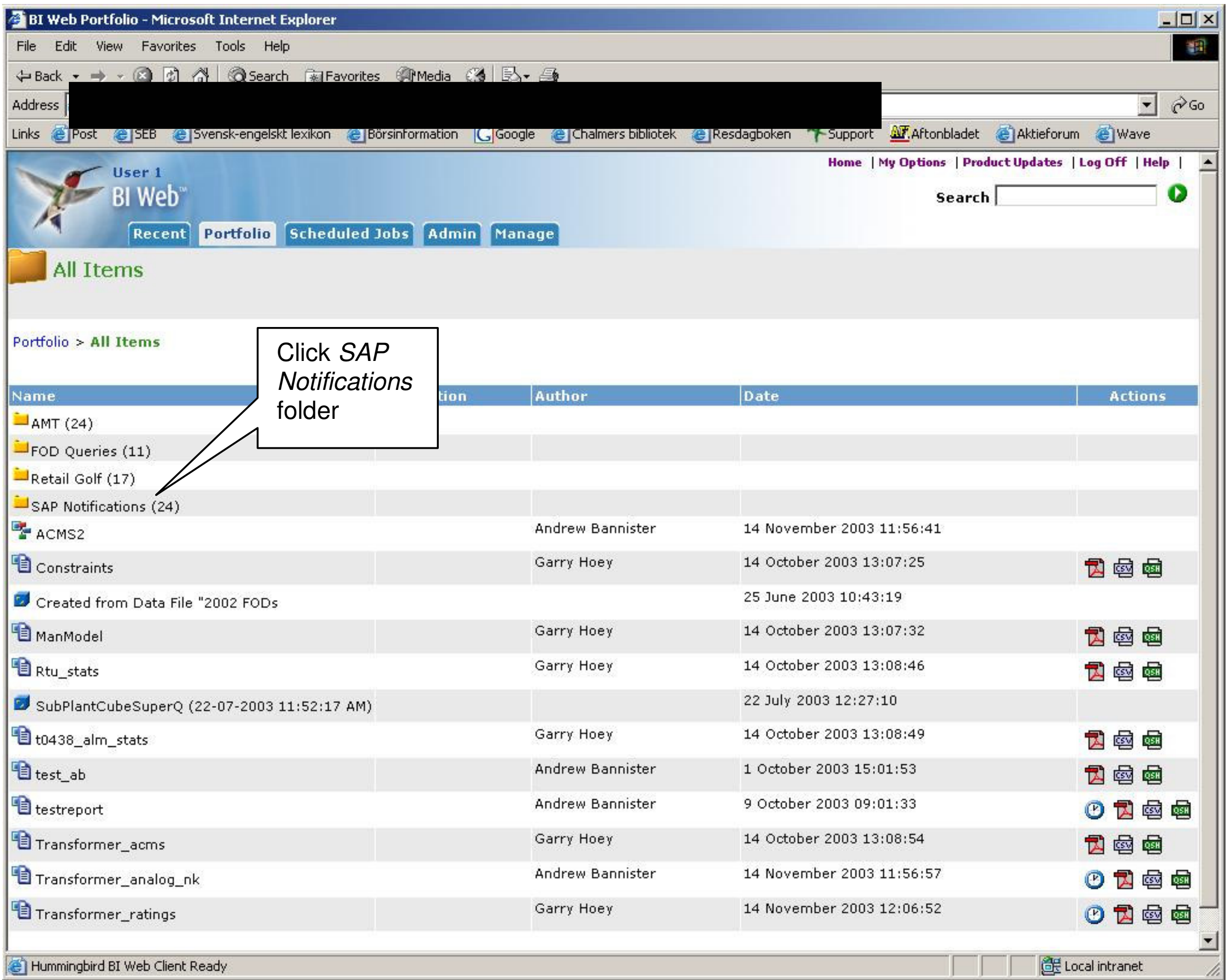


The experienced user can use the shortcut to the *HyperCube* in this window, however notice that this shortcut will only exist if the *HyperCube* recently has been used.

Click the folder named *All Items* to open up a window with all objects in the repository.

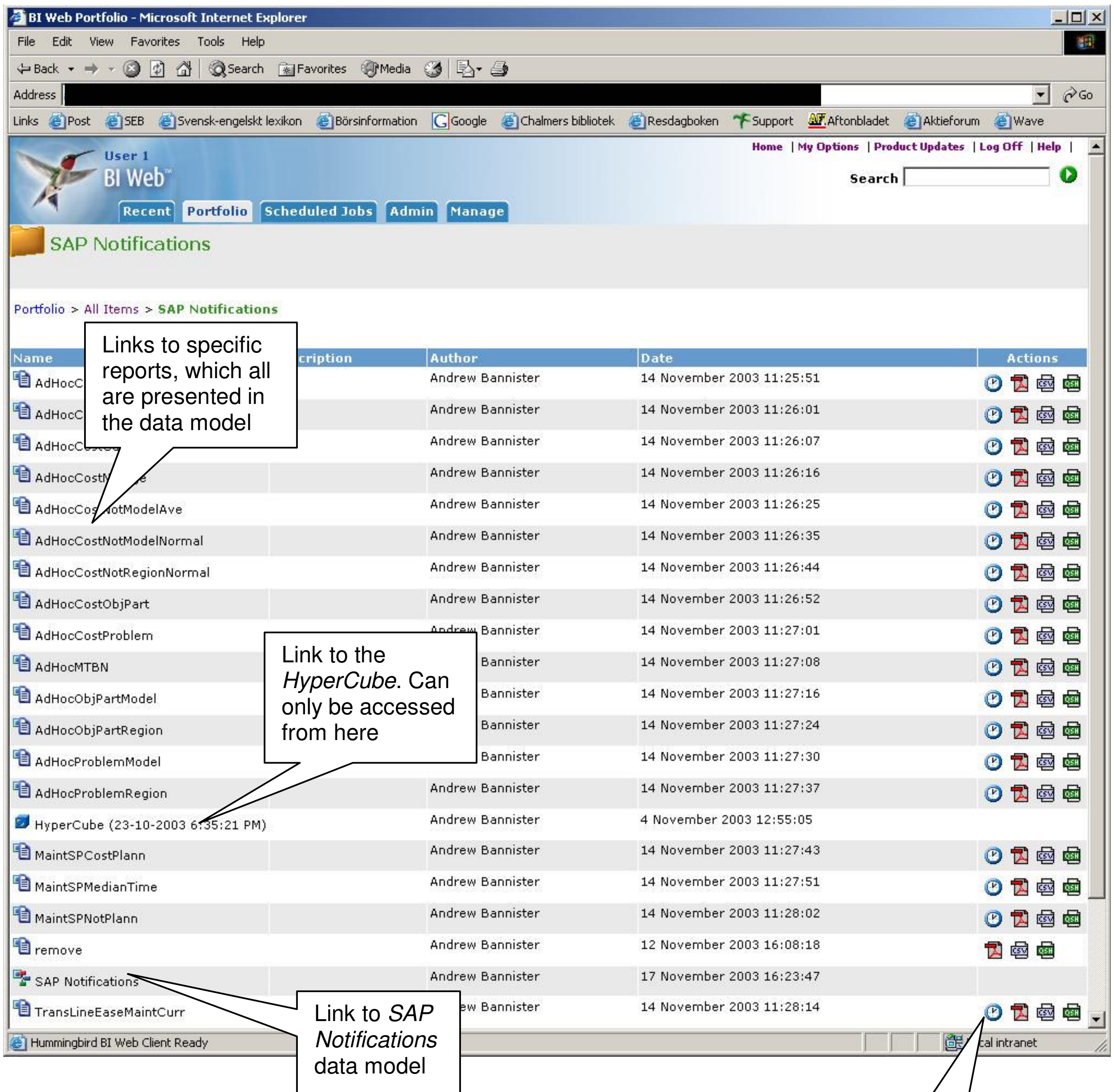


Click on the *SAP Notifications* folder to open up a window with all relevant objects for the data model. The relevant objects are the actual data model, reports and the *HyperCube*.



Procedure Name: **AMTPRO066 Procedure for using SAP Notifications Data Model and the HyperCube**	Procedure No.: **AMTPRO066**	
	Revision Date **20** / **11** / **2003**	Revision No. **1**

In this window all objects relevant to *SAP Notifications* data model are shown. Click the row with *HyperCube* to open up the *HyperCube*.



- Symbol for report

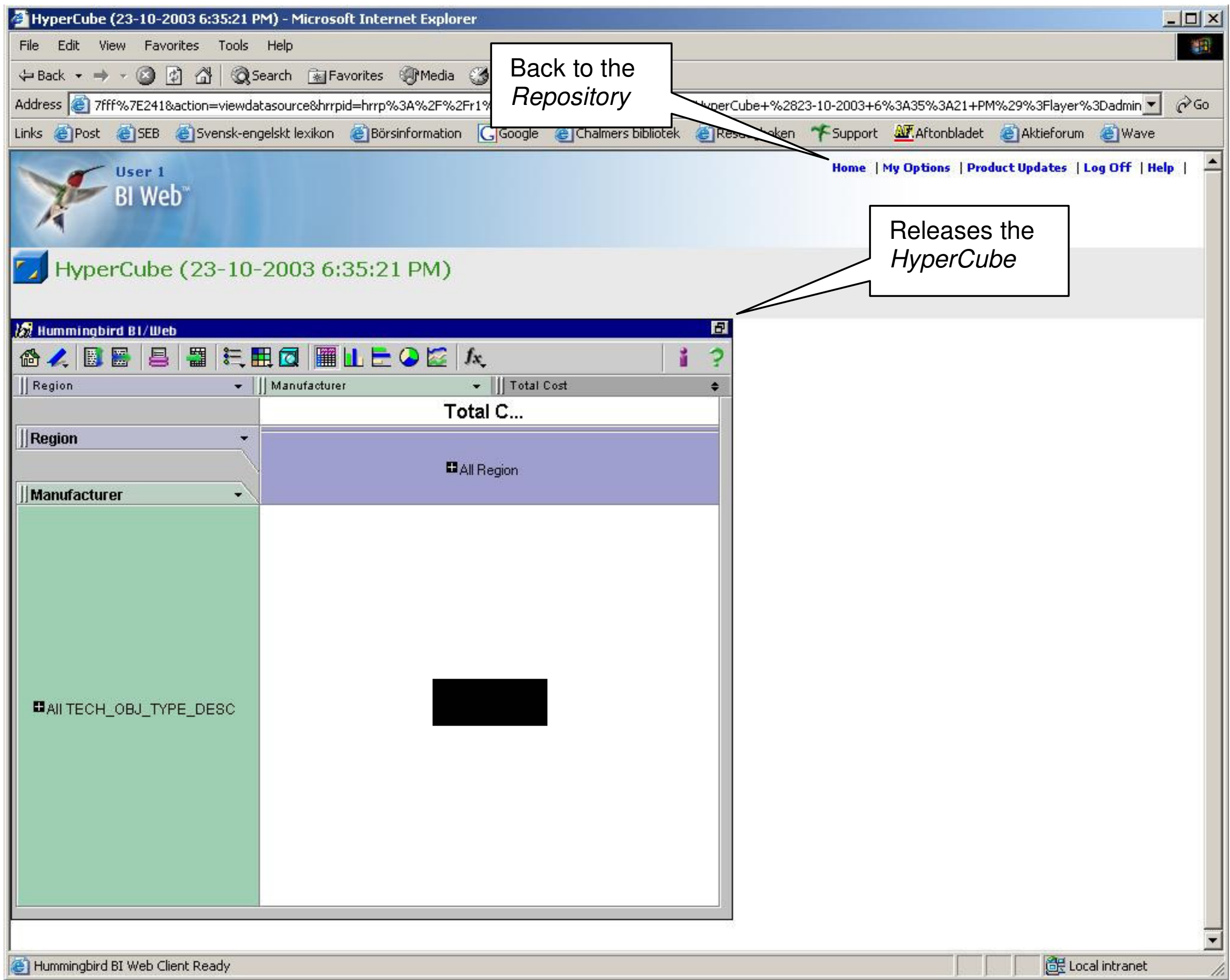
- Symbol for *HyperCube*

- Symbol for data model

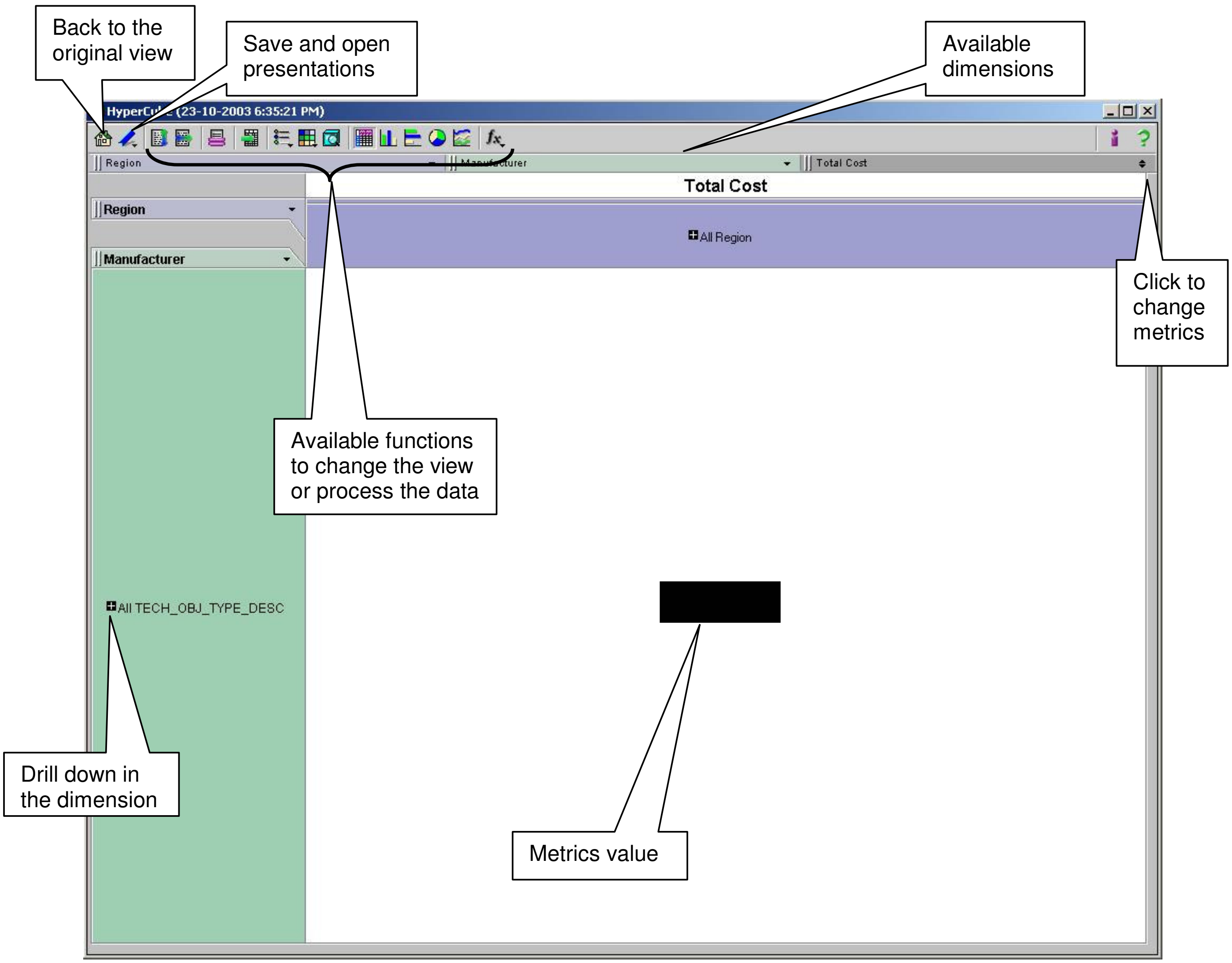
The  -symbol indicates that the report will be refreshed on load

Work with the HyperCube

The window that the user will see when opening up the *HyperCube* looks as below. The *HyperCube* can be released from this window and then re-sized.



There are a lot of ways to use the cube. The data can be drilled to many different levels and different dimensions can be used to view the data. It is possible to choose whether to view the data as figures or tabs, and if there are many metrics it is possible to choose one of them to look at.



Only two dimensions can be used at the same time. When many dimensions are available the dimension used can be changed by just dragging the wanted dimension from the dimension row to the y-axis or the x-axis.

C Maintenance Report - Genio Designer

The maintenance report *AMTREP116 SAP Notifications - End to end process for GENIO* is attached to give the reader a better picture of how the work in Genio Designer is performed.

SAP Notifications - End to end processes

AMTREP116

Controlled Document – do not distribute

Version	Author	Checked by	Signature	Date
1.A	JA/AJ			

Document History

Document Status	Version	Date	Author
Draft	1.0	2/12/03	JA/AJ
Final Report			

Distribution Notice

Warning. A paper copy of this document may not be the current version. Do not copy and distribute before checking the electronic version of the document.

Document Information

Document Owner Asset Monitoring Team
Powerlink Queensland
33 Harold Street
Virginia
Queensland 4014

Telephone 07 3860 2628

Facsimile 07 3860 2522

Email Janders1@Powerlink.com.au

Ajansson@Powerlink.com.au

Copyright

Table of Contents

1. INTRODUCTION

This is report is created as a support for maintenance of the software developed for the project SAP Notifications. It describes the end to end process for all work done by GENIO.

2. BACKGROUND

SAP Notifications is a Project developed for automatic analysis of SAP data for all transmission plant. The goal is to automatically produce frequently used reports highlighting exceptions and also to send out scheduled emails with exception reporting. The reports are available via BI Web, see “AMTPRO066 Procedure for using SAP Notifications Data Model” for information on how to use the data model.

3. GENIO PROJECTS

In the end to end process for the generation of exceptions and data for certain graphs the following projects within Genio Design are to be used (Table 1):

Project	Status	Version
SAP Notifications Comsys	Development	V4 Final
SAP Notifications Data Tables	Development	V2 Final
SAP Notifications Lines	Development	V1 Final
SAP Notifications Secsys	Development	V4 Final
SAP Notifications Service	Development	V1 Final
SAP Notifications Substations	Development	V5 Final

Table 1 Project and respective status and version number

To enable the correct creation of data tables the processes within the different projects need to be executed in correct order, as shown in Figure 3-1 beneath.

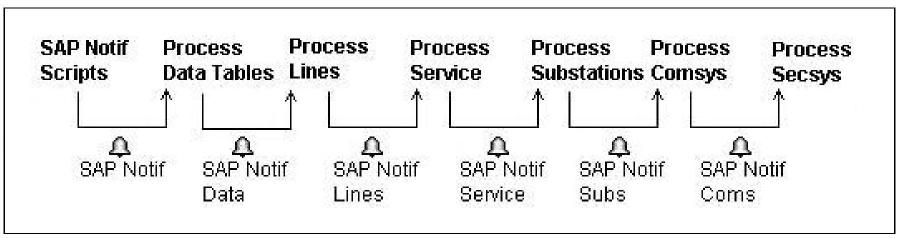


Figure 3-1 Execution order

Processes *SAP Table Scripts* and *Process Data Tables* are in Project *SAP Notification Data Tables* and need to be run first. SAP Table Scripts creates and updates the tables *JEST, Funct_Loc*, *Notifications*, *Orders* and *Equipment*. All of these tables contain raw data for use in the Genio manipulation code and reporting tools. The second process, Process Data Tables, updates certain data tables for Substations, Comsys and Secondary Systems, together with the *Notif table*, which is used extensively by most of the Genio manipulation modules. After these two processes have been run, the others can be triggered to execute in parallel or in series. We have executed the processes in series as shown in Figure 3-1.

In the above scheduling, the process SAP Table Scripts includes a triggering of an event (SAP Notif), which is used for executing Process Data Tables. This in turns triggers *Process Lines* etc, i.e. all processes are executed in an order, like a chain reaction. The first four processes are triggered every day, while the three last ones are to be executed once every quarter. The reason for the “chain reaction” way of executing all process is that if parallel execution would be implemented for the daily respective quarterly processes, the program would be to heavily loaded (implemented and tested).

3.1. Modules within the projects

The six different projects are all built up by a number of variables, data sets, modules and processes. The processes are merely executions of chosen modules and some other tasks like sending E-mails and running of SQL scripts. The data sets are built up by joining of tables and serves as the data source for certain modules. The different projects are connected via the ACMS, which means that tables within the AMT database are being accessed. These are the only tables needed for the project SAP Notifications.

Because of some limitations within Genio, for instance with sorting, filtering and limited length of the scripts that can be written, a number of extra modules have been created to break down certain tasks into “smaller pieces”. This has created a hierarchy of modules were the “main module” calls the second- highest one etcetera.

Worth of notice is that the three projects SAP Notification Substations, Secsys and Comsys for the moment are almost identical since the same exceptions are being created. The only difference is the set of variables used and some smaller parts of certain modules. However, the reason for making three identical ones is for future work; the parameters raising exceptions might differ in the future and it is therefore easier to keep them separated for ease of maintenance. The maintenance was also considered when the project *SAP Notifications Lines and SAP Notifications Service* were created to store the modules and processes related to these areas. Since the processes only populate data tables, it would have been natural to store them in the project SAP Notifications Data Tables, which updates all other data tables. However, to slim down the size of the projects and keep a good structure and overview of the work in Genio, the processes were decided to be stored within its own project.

The following text describes all the different “main” modules (even the ones not used for exception raising so far), what their main task is, which sub modules they call and what exceptions are being raised (where applicable). A short description of the modules is also included together with the scripts in Genio Design, and the raising of exceptions are explained in more detail in chapter 4, GENERATION OF EXCEPTIONS.

3.1.1. SAP Notifications Data Tables

AvAgePerMod_mod	Calculates the average age for every model. Uses the result from AvAgePerModTmp_mod.
AvAgePerModTmp_mod	Calculates the age for all models of a certain technical object type. The result is used in AvAgePerMod_mod.
BathTubIndex_mod	Module used for creating a continuous age axis and the final table for the bathtub curve (notifications versus age). Uses the result from BathTubSorted_mod.
BathTubSorted_mod	Module used for sorting the age column resulting from BathTub_mod.
BathTub_mod	Generates table with number of notifications versus age for all technical object types.
BathTubCostIndex_mod	Creates a continuous age axis and the final table for the bathtub curve (cost versus age). Takes the result from BathTubCostSorted_mod.
BathTubCostSorted_mod	Takes the result form BathTubCost_mod and sorts the age axis.
BathTubCost_mod	Generates table with cost versus age for all technical object types.
CostFailPerCause_mod	Calculates the percentage per cause of failure together with the average cost per notification.
CostFailPerDamage_mod	Calculates the percentage per damage code together with the average cost per notification.
CostFailPerObjPart_mod	Calculates the percentage of failure per object part together with the average cost per notification.
CostPerMod_mod	Calculates the cost per model, normalized on number of equipment. Uses the result form CostPerModTmp_mod.

CostPerModTmp_mod — Used for calculating the cost per model, unnormalised.

MTBN_mod — Creates the final table used for the reporting of the MTBN index. Uses the result from MTBNTmp_mod and table *avage_per_model* as source.

MTBNTmp_mod — Generates the table containing the MTBN index. Uses the result from NotPerMod_mod as source.

NotPerMod_mod — Calculates the number of notifications per model.

Notif_mod — Used for filtering the Notification table for the notification date and status, since direct filtering of the notification table in the main modules gives wrong result.

TechObj_mod — Generates a table with technical object types plus their description. Used in Update_mod and UpdateAvAge_mod

UpdateAvAge_mod — Generates the average age per model data, used in a certain query. Uses TechObj_mod as source to run the module AvAgePerMod_mod for several technical object types.

Update_mod — Populates the data tables used for creating certain reports. Uses TechObj_mod as source to run different modules for several technical object types.

3.1.2. SAP Notifications Substations

CauseMod_mod — Uses the result from CauseModTmp_mod, calculates and raises exceptions for the percentage of cause of failure per model.

CauseModTmp_mod — Calculates the number of notifications per cause of failure per model.

CostFailPerCause_mod — Calculates the percentage per cause of failure together with the average cost. Raises exceptions for normalized number of notifications and normalized cost.

CostFailPerDamage_mod	Calculates the percentage per damage code together with the average cost. Raises exception for normalized number of notifications and normalized cost.
CostFailPerObjPart_mod	Calculates the percentage of failure per object part together with the average cost. Raises exception for normalized number of notifications and normalized cost.
CostPerMod_mod	Described above.
CostPerModTmp_mod	Described above.
CostNotMod_mod	Joins together normalized cost and normalized number of notifications per model (uses the result from CostPerMod_mod and NotPerMod_mod). Raises exceptions for normalized number of notifications, normalized cost and for a ratio between cost and notifications.
DamageMod_mod	Calculates and raises exceptions for the percentage of damage code per model. Uses the result from DamageModTmp_mod.
DamageModTmp_mod	Calculates the number of notifications per damage code per model.
DeleteRow_mod	Reads first row from table tech_objects, saves the value into a variable and thereafter removes the actual row from the table.
NotPerMod_mod	Described above.
ObjPartMod_mod	Calculates and raises exceptions for the percentage of object part failure per model. Uses the result from ObjPartTmp_mod.
ObjPartModTmp_mod	Calculates the number of object part failure per model.
RaiseExc_mod	Module used for raising exceptions for notifications, cost and ratio cost/notification (when applicable). Uses the the average and variance calculated within different modules.

TechObj_mod Creates a table containing all “subs” items.

TotalModule_mod Calls the different modules for the technical object type extracted in module DeleteRow_mod.

3.1.3. SAP Notifications Comsys/Secsys

Uses the same modules as “SAP Notifications Substations” above.

3.1.4. SAP Notifications Lines

Func_Loc_mod Updates certain fields (voltage, length, number of towers etcetera) in a copy of the functional location table named FL. Uses the module Func_LocAdd_mod to add rows.

Func_LocAdd_mod Adds rows in the FL table.

Trans_Line_mod Generates (summarises) data for easement and project cost per functional location for previous and current financial year and different functional locations. Uses the module Trans_LineAdd_mod to add rows.

Trans_LineAdd_mod Adds rows in the Trans_Line table.

3.1.5. SAP Notifications Service

ClTimeMedian_mod Calculates the median closure time for different planning groups and priorities. Uses the result from ClTimeMedianComb_mod and also MedianAddrow_even_mod and MedianAddrow_odd_mod to add specific rows to the target table.

ClTimeMedianComb_mod Combines the result from ClTimeMedianTmp2_mod with the tables ORDERS and NOTIF to get the individual notification numbers and closure time durations for all planning groups and priorities combined with the total number of notifications for each set of planning group/priority.

ClTimeMedianTmp2_mod Calculates the number of notifications for all planning groups and priorities. Uses the result from ClTimeMedianTmp_mod.

ClTimeMedianTmp_mod Creates a table with planning group, priority and all notification numbers.

MedianAddrow_even_mod Adds a specific row in the target table.

MedianAddrow_odd_mod Adds a specific row in the target table.

3.2. Processes

Within the different projects, there are a number of processes created, see Figure 3-2. The processes run certain modules, send E-mails and trigger events etcetera.

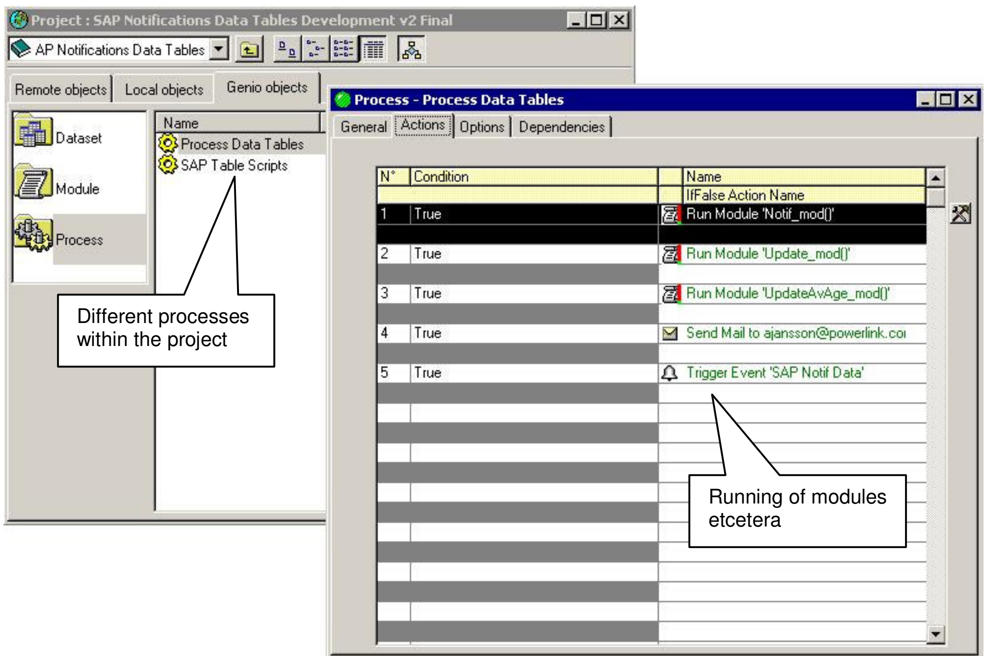


Figure 3-2 Overview of a process with different tasks in Genio.

The processes for the different projects and their main tasks are described in the following text.

3.2.1. SAP Notifications Data Tables

This project includes two processes (see also Figure 3-2); *SAP Table Scripts*, process that runs SQL scripts to manipulate SAP data into four simple tables for the SAP Notifications project. *SAP Notifications Data Tables* updates, as mentioned earlier, the Notif table and also runs Update_mod and UpdateAvAge_mod for updating of data tables used for Substations, Communications and Secondary Systems. After the updating, an E-mail is sent to the creator of the table (or appropriate person) to indicate that the tables have been successfully updated. The event *SAP Notif Data Tables* is also triggered for the execution of the next process (see Figure 3-1)

3.2.2. SAP Notifications Lines

Process Lines

Process that runs the modules Trans_Line_mod and Func_Loc to update the Trans_Line respective FL tables. Sends an E-mail to the user once every quarter when the modules have been run (note, the process is however run <u>every</u> day).

When the modules have been run through, the process triggers the event “SAP Notif Lines”.

3.2.3. SAP Notifications Service

Process Service

Runs the module ClTimeMedian_mod for the updating of the ClTime_Median table. Sends an E-mail to the user once every quarter when the module has been run (note, the process is however run <u>every</u> day).

When the modules have been run through, the process triggers the event “SAP Notif Service”.

3.2.4. SAP Notifications Substation

Process Substations

(Figure 3-3) Process that runs the module TotalModule_mod and sends an E-mail to the user containing number of exceptions raised for each report and item of plant. Modules TechObj_mod and DeleteRow_mod are also executed in the process, together with a simple "go to" command (goes to the DeleteRow module, runs TotalModule_mod and sends E-mail for the next technical object type as long as the tech_objects table is not empty).

When the whole table tech_objects has been run through, the process triggers the event “SAP Notif Subs”.

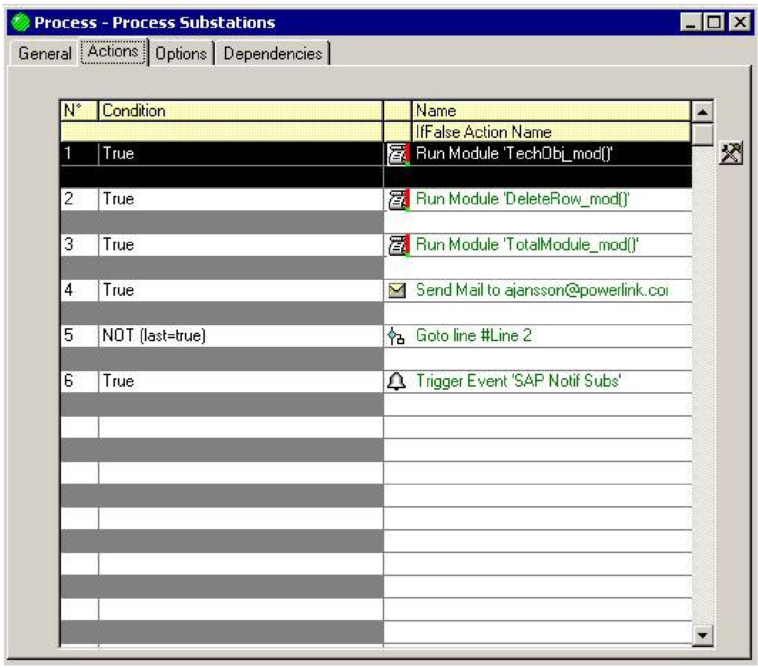


Figure 3-3 Process Substations

3.2.5. SAP Notifications Comsys

Process Comsys

The modules executed are the same as for *Process Substations,* the only difference is that the module TechObj_mod only contains Communication system objects.

When the whole table tech_objects has been run through, the process triggers the event “SAP Notif Coms”.

3.2.6. SAP Notifications Secsys

Process Secsys

The modules executed are the same as for *Process Substations* and *Process Comsys,* the only difference is that the module TechObj_mod only contains Secondary System objects.

4. GENERATION OF EXCEPTIONS

This chapter aims to provide some more low level detail about the way exceptions are being raised within the projects SAP Notifications Substations, Comsys and Secsys. The way exceptions are raised are not only identical in the three different projects, but also more or less identical within each project for the different reports. The reason for this is that standard deviations are used, which are calculated in the same way for different data sets, and the same percentage of deviation form the average is used for all the reports. Due to this, simplifications can be made, i.e much of the work can be collected in one module, *RaiseExc_mod*, that use average and standard deviation for each data value to decide whether exception is to be raised or not. If an exception is raised, a counter is increased one step and when all data has been processed, the value of the counter is included in the E-mail sent to the asset manager.

The thresholds for the exceptions were set after investigation of the distribution of the data had been performed, see procedure for exception raising AMTPRO 0073. In this document, it is found out that the data is gamma distributed and almost exponential. The values of the thresholds used are therefore based on this outcome.

4.1. The generation

In Figure 4-1 and Figure 4-2 below, the way exceptions are raised for *Object Part Per Model* is presented. In the first figure, it is shown how three different variables are being calculated and thereafter used for exceptions raising, according Figure 4-1.

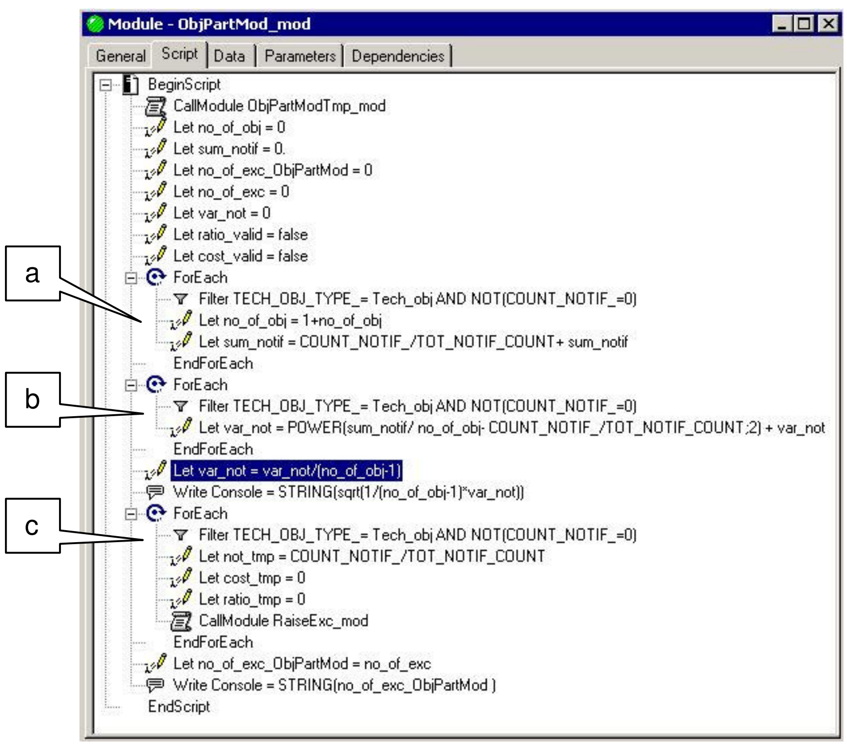


Figure 4-1 Raising of exception for *Object Part Per Model*

In Figure 4-1 the three different “loops” are used for

a. Calculating number of objects (models in this case) and notifications. For this particular report (Object Part per Model), exceptions are not raised for cost or ratio between cost and notifications, so no calculation for these parameters is being done here.

b. A term containing the variance, σ^2, for the notifications is calculated according to the formula

$$\sigma^2 \cdot (n-1) = \sum (x - \bar{x})^2$$

where n is the number of objects and $\bar{x}$ is the average of the number of notifications, calculated with the result from a.,

$$\bar{x} = \frac{\sum x}{n}$$

c. Module RaiseExc_mod (Figure 4-2) is called upon for each notification value. In this case the normalized number of notifications is used together with the total number of objects (a.), total number of notifications (a.) and the variance (b.).

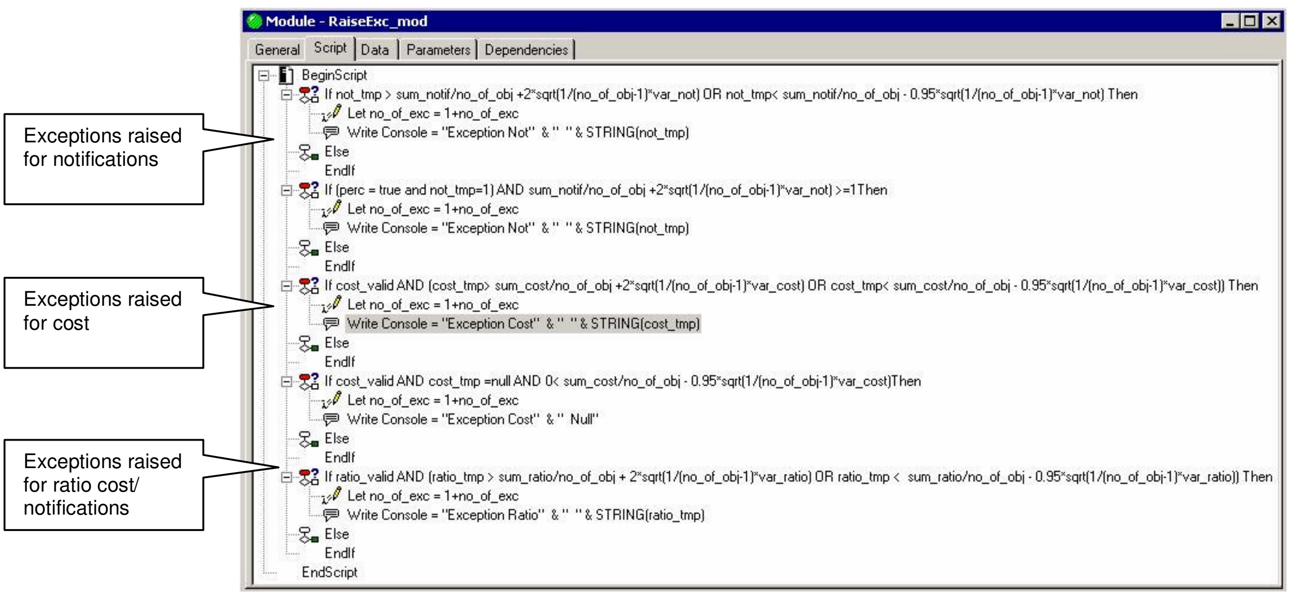


Figure 4-2 Module *RaiseExc_mod*, used for counting number of exceptions raised for all different reports.

As seen in Figure 4-2, an exception is raised if the normalized number of notifications exceeds $\bar{x} + 2 \cdot \sigma$ or is below $\bar{x} - 0.95 \cdot \sigma$, which corresponds to values outside a 90% interval around the average.

4.2. The E-mail feature

When a technical object has been run for the all the exception raising modules, the E-mail for the same object is sent out to the user, the asset manager, according to the process in Figure 3-3. As seen in Figure 4-3, the E-mail includes a header, telling which technical object type has been processed and who the receiver of the mail is. The body includes which date the E-mail is valid for, the reports with their respective number of exceptions raised and a link to both the web based data model and the HyperCube (where applicable). For each report there is also a hyperlink, taking the user direct to the Web report.

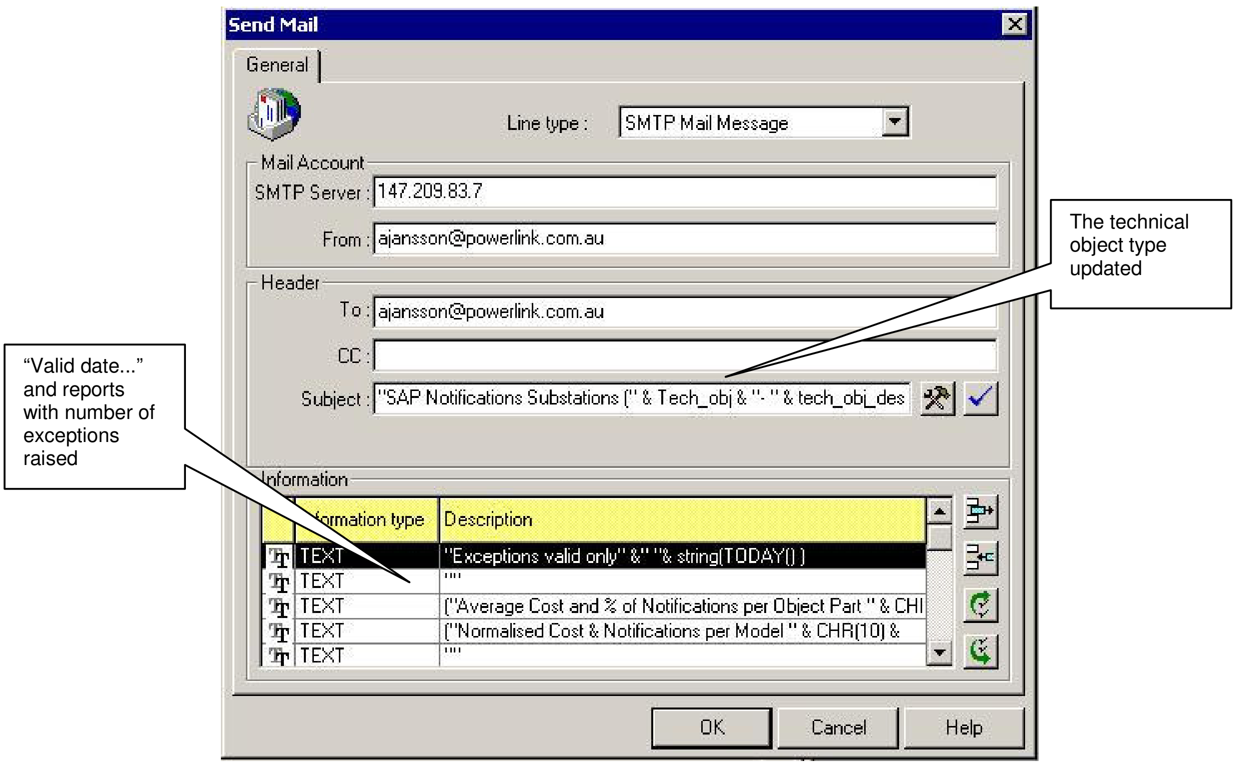


Figure 4-3 The E-mail function

5. VERIFICATION OF DATA IN THE WEB BASED REPORTS

The data used in the *SAP Notifications model* today has been filtered in different ways to only include relevant data. Only notification dates greater than or equal to 01/07/1999 have been included. Notifications where the description contains *engarde* (data from the old corporate software application) and/or the status is *DLFL* (deletion flag) have been excluded. All *INAK* (inactive) items are included to maintain the history. In the bathtub report equipment with start up dates 01/01/1900 or 05/07/1999 has been excluded to ensure good quality of the data.

When comparing data from the reports with SAP data, due to the restrictions above, there are a number of things that have to be taken into consideration. The way the validation of the data has been performed depends on the area investigated. The data validation process will be described for each area and each report. When there have been similarities between areas or reports these have been discussed in a group, for example the areas of *substations*, *communications* and *secondary systems* all use the same reports and are therefore described together.

Notice that the figures presented in this section will change over time due to new costs, notifications and other parameters changing. However, when a verification has been done it does not have to be repeated if it was shown that the data in SAP and the data model matched. But remember that data in new reports has to be verified.

5.1. Substations, Communications and Secondary systems

Data in the reports for substations, communications and secondary systems has been checked for disconnectors (*SUBS0021*) including INAK. The general method has been to control that the total number of equipment, notifications or cost has been equal to the data displayed in SAP. If that has been the case then no further investigations have been performed. The reports are controlled in the order in the order they appear in the data model, and they have also been grouped.

5.1.1. By Item

-Cost and Notifications per Age (bathtube)

Regarding the number of equipment and notifications, the result in SAP is 1954 equipment and 130 notifications. In BI Query the result is 1953, the difference in number of equipment is due to the equipment with number 20044693. It has a valid date from 14.10.2003, which was outside the range when Genio Designer created the data for the report. The number of number of notifications is correct in the report.

The total cost retrieved from SAP is $181 136.51 (with 113 notifications, SUBS0021), to be compared with $179 246.51 from the curve (with 113 notifications). The difference is for notification number 10051056 where it seems to be the right total cost in SAP while there is a smaller cost within BI Query. Probably the difference is due to the fact that in the beginning of the project the tables were not updated daily (the case when this investigation was performed) and therefore the order table do not have the correct cost.

-Average cost and % of Failure per Object Part

Transaction IW29 is used for achieving data for the notifications. All startup dates, only notification status NOCO (completed notifications), no engarde data and notification dates greater than or equal to 01.07.1999 were used when querying both SAP and the data model. This gives the result of 486 notifications for both data sources.

The total actual cost is retrieved from SAP by using transaction IW39. Chose multiple selection in the notifications field and then multiple selection in the equipment field that appears in the new screen. In the equipment screen just enter the appropriate technical object type. With that, the 486 notifications mentioned above results in the following costs:

SAP: $518 775.76

BI Query: $516 704.40

The reason for the difference is that there are two notification numbers that represent a cost in BI Query. These are not connected to any order within SAP, hence they do not have a cost associated with them in SAP. When running a query in the data model, the notification and order tables are joined together via order number, which results in a cost connected to two different notification number, one from the notification table and one from the order table. The result is:

Notifications.notif_no	Equip_no	Total_cost	Order_no	Orders.Notif_no
10013519	20011954	$353.10	5078091	10013520
10021384	20002839	$318.50	5093580	10021383

When performing a search in SAP with the given order number, it is obvious that the correct notification number for the order are the ones within the order table above. This is an existing problem that should be fixed as soon as possible.

There is one notification number that is not filled in within the order in SAP, causing its corresponding cost not to be displayed when performing a search in SAP. The reason is that the orders associated with disconnectors are retrieved with help of all notifications for disconnectors. However, when setting up a join between the notification and order tables in the data model, the cost is displayed due to the join is on order number and the notification mentioned has an order number in SAP.

A number of deviations in costs between SAP and the data model are present for the chosen technical object type and these are displayed in the table below. The costs are higher in SAP and the notification numbers connected to the orders are quite new, so one possibility is that the costs have been changed after the order table within the data model has been updated. This is a problem that no longer will appear due to the daily update of the tables.

Notif nr	Notif date	Basic start	Cost- BI	Cost-SAP
10050385	10.07.2003	01.09.2003	$528.6	$694.6
10050386	10.07.2003	02.09.2003	$529.69	$695.69
10050387	10.07.2003	02.09.2003	$458	$624
10051056	11.08.2003	11.08.2003	$649	$2,539.00
10051537	06.09.2003	19.08.2003	$39	$179
10052019	16.09.2003	16.09.2003	$0	$270

With these differences taken into consideration the difference between SAP and the data model is only $55.04. Should check what the cost is for the notification number which was missing in the order in SAP.

-Average cost and % of Failure per Damage/Cause

The number of equipment, the cost and the number of notifications are the same as for the above graphs, that is the number of equipment equals 1953, the total number of notifications is 486 and the total cost sums up to $516 704.

5.1.2. By Model

- Normalised Cost and Notifications per Model

The cost and number of notifications are the same as for the above graphs, that is a total number of notifications of 486 and a total cost of $516 704.

-Object Part Failure/Problem of Failure/Cause of Failure per Model

The number of notifications is as above and therefore correct.

-MTBN

A summation of the number of notifications and number of equipment gives the result 486 notifications and 1953 items. This means that one item of equipment is missing which is mentioned in the investigation above.

The calculation of the MTBN values should be checked. A random check as with the average age in the population query below should be enough.

-Average Cost per Notification per Model

The cost is correct but the total number of notifications differed by four. The reason was that the <null> costs were not displayed due to how the super query in the data model was set up. After some changes were done in the super query the number of notifications is correct.

-Population of Equipment per Model (query)

The number of equipment sums up to 1953, the reason for why one equipment is missing according to SAP (1954) has been discussed earlier in this section. The average age (calculated in Genio Designer) have been checked for three random chosen values and seemed to match with values achieved via calculations with SAP data.

5.1.3. By Region

-Normalised Cost and Notifications per Region

The number of equipment, the cost and the number of notifications are the same as for the above mentioned graphs, that is the number of equipment equals 1953, the total number of notifications is 486 and the total cost sums up to 516 704 $.

-Object Part Failure/Problem of Failure/Cause of Failure per Region

Correct number of notifications as above.

-Population of Equipment per Model (query)

The number of equipment sums up to 1953, which is discussed above.

5.2. Transmission Lines

For the transmission lines, the parameters validated are circuit length, route length and total cost for all functional locations between 1000 and 2000 (these functional locations are the ones for lines).

5.2.1. Current Financial Year

The data in the report is compared with the real data in SAP where route length and total cost are the two parameters compared. The report is checked for all regions (region value left out at prompt), a voltage level between 0-500 kV. Type MVEG for easement cost and not (CHEM, MECH, ACCT, MVEG, TPRN) for plant cost. 01/07/2003 <= finish date <= 30/06/2004

-Easement Maintenance Cost

The total route length according to SAP should be 8165.81 but the presented value in the report is 8099.1 km. The reason for this is because underground cables do not have functional location groups as the overhead lines have, for example vegetation cost (MVEG). This makes it impossible to present them in the report. There are also a number of functional locations without MVEG type or without a voltage level in the SAP data, and should not be included. The total missing route length for the cables plus locations without MVEG or voltage level is 66.86, and is equal to the total difference between the data from SAP and the data model.

There is also a functional location present in the report but not in the SAP data. This is for functional location 1010-SPN-MVEG (Route Length = 0.1), which is of status INAK and should maybe not be included in the report. This is to be discussed, it is kept in the data model for history purpose.

The total cost matched perfectly, $89931. The result was achieved by choosing order types PM02 and PM03, maintenance type MM and MR and a basic start date greater than 01/07/2003.

-Plant Maintenance Cost

The functional locations present in the report are the ones that have had orders on them, and also where the cost has not been zero. This means that the total circuit length in the report most certain will be less than in SAP. This could be changed so all relevant functional locations are presented, but this would probably give a sparse and big table. The data has been checked for all regions and all voltage levels. The lengths match when considering all functional locations.

The cost from SAP is $299 901.61 while it is only $286 785 in the report. The difference is due to the problems mentioned before with underground cables. Underground cables are present in the SAP data while they are not shown in the report. The result is achieved by using PM02 or PM03 orders and maintenance type MA, MB and MR.

5.2.2. Previous Financial Year

The data in the report is compared with the real data in SAP where route length and total cost are the two parameters compared. The report is checked for all regions (region value left out at prompt), a voltage level between 0-500 kV. Type MVEG for easement cost and not (CHEM, MECH, ACCM, MVEG, TPRN) for plant cost. 01/07/2002 <= finish date <= 30/06/2003

-Easement Maintenance Cost

The length is checked the same way as for the reports for current financial year and the result is the same as presented for that report

The cost matched perfectly, $599 424 both in SAP and the report.

-Plant Maintenance Cost

The value of the circuit length is checked in the same way as the current financial year and the result is the same.

The cost does not match. The cost from SAP is $2 298 906.42 and in the report the total cost is $2 233 327. The difference of $65 579.42 is equal to the total cost for all underground cables and a cost of $18 438 which is settlement cost for functional location 1140-xxx which is missing in the reporting (order number 5111792).

5.3. Maintenance Service Providers

The data quality in SAP has to be fixed before any fruitful investigation can be performed. The problem is that when searching for planner-group 091 the data retrieved belongs to, according to SAP, another planner-group, planner-group 092. This problem has been addressed and Andrew Bannister and Peter Loder perform an investigation.

-Cost per Planner Group

-Notifications per Planner Group

-Median Closure Time per Priority per Planner Group

6. CONCLUSIONS

xxxx

7. RECOMMENDATIONS

It is recommended:

- xxxx

Johan Andersson/Alexander Jansson

Temporary Researchers

Asset Monitoring Team

8. REFERENCES

[1] xxxx

APPENDIX A – XXXX

xxxx

D Maintenance Procedure - Data Distribution

The maintenance procedure *AMTPRO073 Procedure for exception raising in SAP Notifications Data Model* is attached to give the reader a better idea of the work that have to be done when checking the distribution of data and whether the data change over time.

Asset Monitoring Team - Procedure

Procedure Name: **AMTPRO073 Procedure for exception raising in SAP Notifications Data Model**	Procedure No.: **AMTPRO073**	
	Revision Date **17** / **11** / **2003**	Revision No. **1**

AMTPRO073 Procedure

for exception raising in SAP Notifications Data Model

5					
4					
3					
2					
1					
0	17/11/03	ORIGINAL DOCUMENT	J.A/A.J		
REV	DATE	REVISION DESCRIPTION	AUTHOR	CHECKED	APPROVED

Asset Monitoring Team - Procedure

Procedure Name: **AMTPRO073 Procedure for exception raising in SAP Notifications Data Model**	Procedure No.: **AMTPRO073**	
	Revision Date **17 / 11 / 2003**	Revision No. **1**

Summary of how exceptions are raised

The data in the reports are based on the data visible in the tables. Those tables are sorted on one of the columns. Exceptions are raised on the data when some values exceed the calculated threshold. These values will change colour to red and then be centred in the table for visibility purpose. In the sorted columns the exceptions will appear in the top and bottom of the column. For the other non-sorted columns they might appear randomly.

The threshold limits are calculated from the data in question. The procedures is:

1. Determine distribution of data
2. Decide on what and how to raise exceptions
3. Implement this in the report

It has been determined, with significance test, that all data in the reports for all item of plants can be assumed to be gamma distributed. This gamma distribution is in fact very close to an exponential distribution.

Two possible cause to raise exceptions exist, either for the values that deviates from the curve form of the data, above or below the theoretical distribution, or for values in the tails of the distribution. Exceptions in the report are only raised for values in the tails, that is, high and low values. The reason is that this is the most logical way and exceptions raised for the other cause would be hard to implement with the existing software.

The exceptions are raised for the bottom 5% of the data and the top 5% of the data, and hence 90% of the data should lie inside the threshold limits. The reason for using such wide limits is due to the data is gamma distributed and almost exponential, 95% would be to narrow.

The result is then shown as raised exceptions and the threshold limits is visible in the threshold box in the reports where exceptions are used.

Asset Monitoring Team - Procedure

Procedure Name: **AMTPRO073 Procedure for exception raising in SAP Notifications Data Model**	Procedure No.: **AMTPRO073**	
	Revision Date **17** / **11** / **2003**	Revision No. **1**

Table of Contents

Asset Monitoring Team - Procedure

Procedure Name: **AMTPRO073 Procedure for exception raising in SAP Notifications Data Model**	Procedure No.: **AMTPRO073**	
	Revision Date **17 / 11 / 2003**	Revision No. **1**

1 Work that have to be done before new exceptions or reports are implemented

The exceptions implemented so far work completely automatic and no further work should have to be done. However, if new exceptions or new reports are to be introduced to the data model there is some work that has to be performed. First the raw data used for the reports has to be examined to determine the distribution of it. Then the raw data will be used to calculate the upper and lower threshold, this is done both in the reports and in Genio Designer. The implementation of exceptions in the reports and in Genio Designer differs a little bit.

2 How to determine the distribution of the raw data

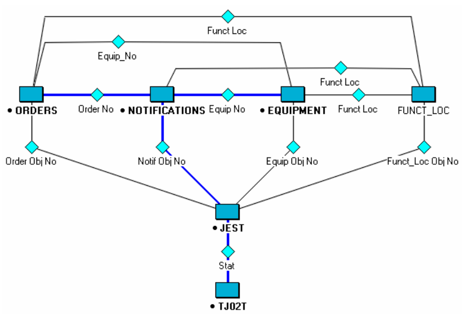
The only reason the distribution of the raw data has to be decided is due to that the way of calculating the thresholds differs some depending on what kind of distribution the data has. The data used in this project is expected to be gamma distributed (for more information, contact the Asset Manager, Brian Sharp, bsharp@powerlink.com.au). Therefore the data will be tested for how well it fit to a gamma distribution. All data used so far have fitted well to the gamma distribution sometimes after small modifications, for example removal of two or three peaks.

The process of determining the distribution of the data will be described in the following step by step explanation, and to give some help there is a template excel spreadsheet (Distribution Template.xls) to which the data merely has to be copied.

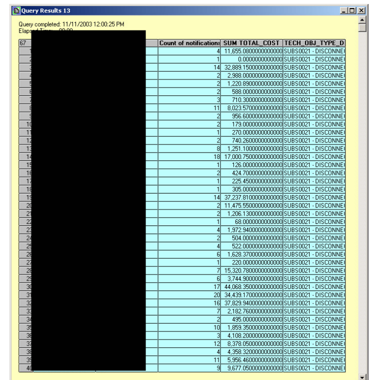
a. Collect the raw data that is to be used in the reports. Conduct all calculations on it, for example normalisation, and then pick the column containing the data for which the distribution should be determined. This column should be loaded into a spreadsheet in Excel.

Asset Monitoring Team - Procedure

Procedure Name: **AMTPRO073 Procedure for exception raising in SAP Notifications Data Model**	Procedure No.: **AMTPRO073**	
	Revision Date **17 / 11 / 2003**	Revision No. **1**



Conduct query to receive wanted data



Query Results 13

Query completed: 11/11/2003 12:00:25 PM

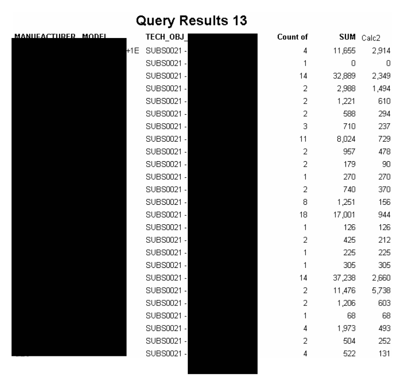
[redacted]

67	[redacted]	Count of notifications	SUM TOTAL_COST	TECH_OBJ_TYPE_D
1		4	11,655.0800000000000	SUBS0021 - DISCONNE
2		1	0.0000000000000	SUBS0021 - DISCONNE
3		14	32,889.1500000000000	SUBS0021 - DISCONNE
4		2	2,988.0000000000000	SUBS0021 - DISCONNE
5		2	1,220.8900000000000	SUBS0021 - DISCONNE
6		2	588.0000000000000	SUBS0021 - DISCONNE
7		3	710.3000000000000	SUBS0021 - DISCONNE
8		11	8,023.5700000000000	SUBS0021 - DISCONNE
9		2	956.6000000000000	SUBS0021 - DISCONNE
10		2	179.0000000000000	SUBS0021 - DISCONNE
11		1	270.0000000000000	SUBS0021 - DISCONNE
12		2	740.2600000000000	SUBS0021 - DISCONNE
13		8	1,251.1000000000000	SUBS0021 - DISCONNE
14		18	17,000.7500000000000	SUBS0021 - DISCONNE
15		1	126.0000000000000	SUBS0021 - DISCONNE
16		2	424.7000000000000	SUBS0021 - DISCONNE
17		1	225.4500000000000	SUBS0021 - DISCONNE
18		1	305.0000000000000	SUBS0021 - DISCONNE
19		14	37,237.8100000000000	SUBS0021 - DISCONNE
20		2	11,475.5500000000000	SUBS0021 - DISCONNE
21		2	1,206.1300000000000	SUBS0021 - DISCONNE
22		1	68.0000000000000	SUBS0021 - DISCONNE
23		4	1,972.9400000000000	SUBS0021 - DISCONNE
24		2	504.0000000000000	SUBS0021 - DISCONNE
25		4	522.0000000000000	SUBS0021 - DISCONNE
26		6	1,628.3700000000000	SUBS0021 - DISCONNE
27		1	220.0000000000000	SUBS0021 - DISCONNE
28		7	15,320.7800000000000	SUBS0021 - DISCONNE
29		6	3,744.9000000000000	SUBS0021 - DISCONNE
30		17	44,068.3500000000000	SUBS0021 - DISCONNE
31		20	34,439.1700000000000	SUBS0021 - DISCONNE
32		16	37,829.9400000000000	SUBS0021 - DISCONNE
33		7	2,182.7600000000000	SUBS0021 - DISCONNE
34		2	495.0000000000000	SUBS0021 - DISCONNE
35		10	1,859.3500000000000	SUBS0021 - DISCONNE
36		3	4,108.2000000000000	SUBS0021 - DISCONNE
37		12	8,378.0500000000000	SUBS0021 - DISCONNE
38		4	4,358.3200000000000	SUBS0021 - DISCONNE
39		11	5,956.4600000000000	SUBS0021 - DISCONNE
40		9	9,677.0500000000000	SUBS0021 - DISCONNE

Load the actual column directly to Excel if it not is to be processed

Load the raw data to a report and process

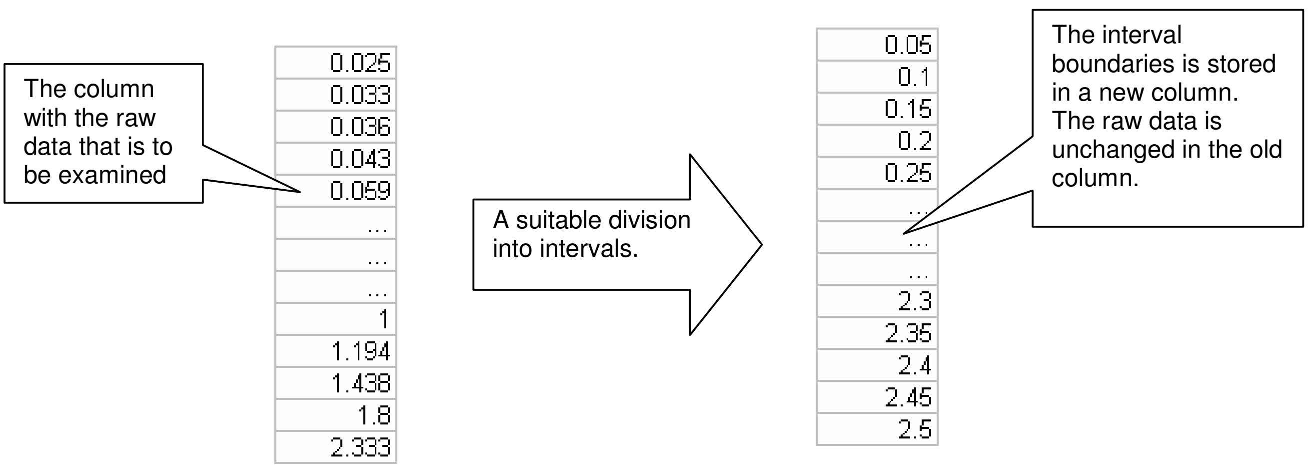
Load column with the data the distribution shall be determined for



Query Results 13

MANUFACTURER_MODEL	TECH_OBJ_	[redacted]	Count of	SUM	Calc2
[redacted]	SUBS0021 -		4	11,655	2,914
	SUBS0021 -		1	0	0
	SUBS0021 -		14	32,889	2,349
	SUBS0021 -		2	2,988	1,494
	SUBS0021 -		2	1,221	610
	SUBS0021 -		2	588	294
	SUBS0021 -		3	710	237
	SUBS0021 -		11	8,024	729
	SUBS0021 -		2	957	478
	SUBS0021 -		2	179	90
	SUBS0021 -		1	270	270
	SUBS0021 -		2	740	370
	SUBS0021 -		8	1,251	156
	SUBS0021 -		18	17,001	944
	SUBS0021 -		1	126	126
	SUBS0021 -		2	425	212
	SUBS0021 -		1	225	225
	SUBS0021 -		1	305	305
	SUBS0021 -		14	37,238	2,660
	SUBS0021 -		2	11,476	5,738
	SUBS0021 -		2	1,206	603
	SUBS0021 -		1	68	68
	SUBS0021 -		4	1,973	493
	SUBS0021 -		2	504	252
	SUBS0021 -		4	522	131

b. The data loaded into Excel should only be one column, for example the column with *normalised notifications per equipment* from the report *Normalised Cost and Notifications per Model* that is used in this example.
- Copy the data to a column in a spreadsheet and sort the data ascending. A histogram should now be created, but first some work has to be done.
- Start first with checking the highest value of the data and then split up the range from zero till max into suitable interval sizes.
- Create a new column with the interval boundaries, this is depicted below. Notice that this new column will probably not have the same number of rows as the raw data column.



The range is now split up into intervals of length 0.05. From 0 – 0.05, 0.05 – 0.1, … , 2.45 – 2.5.

c. Use the two columns in the spreadsheet (one with the raw data and the other just created) to create a histogram with the existing functions in Excel. The function for creating histograms is accessed and used by:
- Click on *Tools* on the menu row
- Click *Data Analysis…*
- Chose *Histogram* and click *Ok*
- Click in the white box after *Input Range* and then mark the column with the raw data
- Click in the white box after *Bin Range* and then mark the column with the interval boundaries
- Mark the checkbox in front of *Chart Output*
- Click *Ok*
- Now a new sheet should appear that contains a table with the number of data values within each interval and also a graph with the histogram.

If the alternative *Data Analysis …* does not exist under the *Tools* menu the following have to be done:
- Click *Tools* on the menu row
- Chose *Add-Ins…*
- Mark Analysis *Toolpack* and click *Ok*
- Now the go back and create the histogram

d. Go back to the sheet with the raw data and perform the following simple calculations with it, Excel functions shown:

- Quadrate the data in a new column (both columns will be used), x=x^2
- Count the number of data values, n=COUNT(…)
- Calculate the mean of the raw data, μ=AVERAGE(…)
- Calculate the standard deviation of the raw data, σ=STDEV(…)
- Summarise the raw data column, Σx=SUM(…)
- Summarise the quadrated column, Σx^2=SUM(…)
- Calculate the frequency λ, used to calculate the theoretical distribution, with:

$$\lambda = (n-1)\frac{\sum x}{\left(n\sum x^2 - \left(\sum x\right)^2\right)}$$

- Calculate η, also used for the theoretical distribution, η=μλ

e. Next step is to create the actual probability function for the data.
- Create a new column next to the interval column with the plotting points. This column should contain the middle of each interval and is used when the data is plotted. If the interval is 0 – 0.05, the plotting point is then 0.025.
- Copy the column called *Frequency* from the sheet with the histogram and put it next to the column created above.
- Next to the column added above calculate the actual probability for each interval f(t)=Number of data values in interval (frequency) / Total number of data values (n)
- Use the just calculated column and calculate the cumulative probability in each interval. F(t)=f(t)+f(t-1)

This might look something like the picture below

5.71% + 4.29% = 10.0%

The frequency column from the table on the sheet with the

Intervall	Ploting point	Count	f(t) actual	cumulative
0.05	0.025	4	5.71%	5.71%
0.1	0.075	3	4.29%	10.00%
0.15	0.125	9	12.86%	22.86%
0.2	0.175	9	12.86%	35.71%
0.25	0.225	7	10.00%	45.71%
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…

7 data values in the interval 0.2 – 0.25

f. To be able to determine how close the real data is to a gamma distribution the theoretical distribution is determined.
- This is done with the following function:

$$f(t) = \frac{t^{(\eta-1)}e^{-\lambda t}\lambda^{\eta}}{e^{GAMMALN(\eta)}}l$$

, where t represents the plotting point, l the interval length and gammaln is a function in Excel.
- This function is then used for every interval to calculate the theoretical probability distribution in a new column.
- Another new column is added in which the cumulative distribution of the theoretical probability is calculated, F(t)=f(t)+f(t-1)

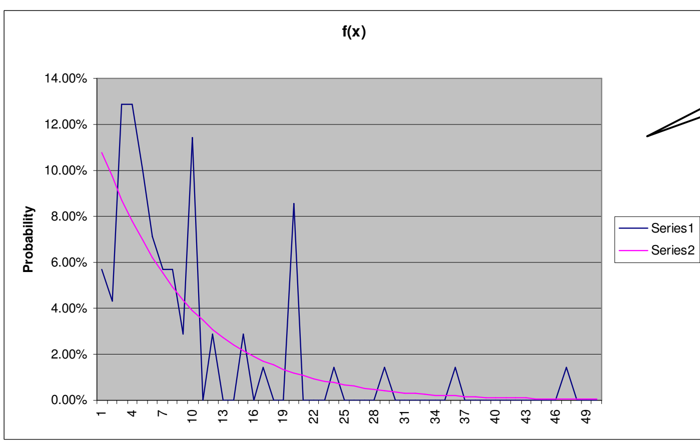
With the new columns the figure above should change to something like the one below

The two new columns with theoretical data

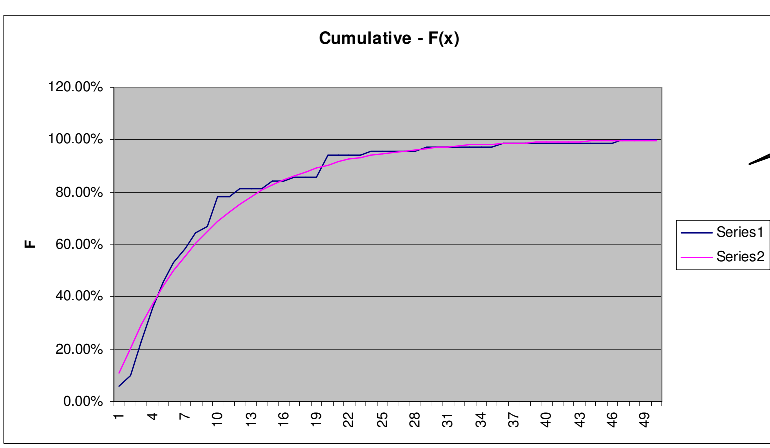
	Intervall width	0.05				
Intervall	Ploting point	Count	f(t) actual	cumulative	f(t) theory	F(t)
0.05	0.025	4	5.71%	5.71%	10.75%	10.75%
[illegible]	0.075	3	4.29%	10.00%	9.74%	20.49%
0.15	0.125	9	12.86%	22.86%	8.73%	29.23%
0.2	0.175	9	12.86%	35.71%	7.80%	37.03%
0.25	0.225	7	10.00%	45.71%	6.96%	43.99%
...	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...

The intervall width used to calculate the theoretical values

g. A graphical comparison can be done to give an overview of how good the data fit to the theoretical curve. This is done both for the ordinary- and the cumulative probability function. Just use the graphing tools in Excel and plot the actual versus the theoretical. This could look like the figures below.



Actual- versus theoretical probability function



The cumulative comparison

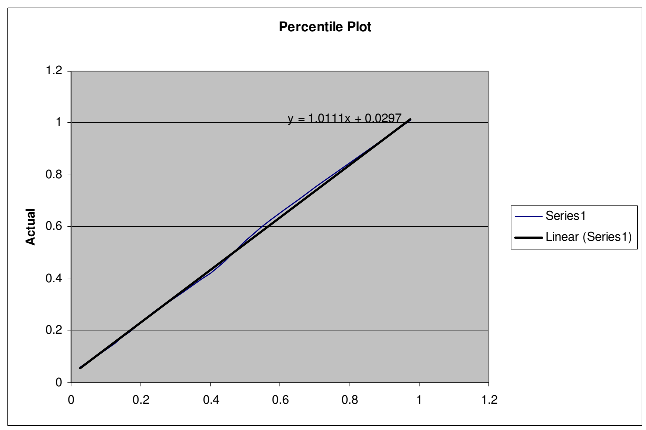
h. Another way of graphically look at how well the data fits is to use the percentile plot. This is done by choosing a couple of cumulative probability values where the actual plotting point is compared with the one achieved from the Excel function PERCENTILE when used on the actual data.

- Chose a couple of spread cumulative values, around 5, and put them in a separate column.
- Copy the appropriate plotting point and put it next to its cumulative value.
- Use the function PERCENTILE to get the actual plotting point. PERCENTILE needs two inputs, the raw data column and the different cumulative percentage value separate.
- Plot the theoretical plotting points (the copied ones) versus the actual (the calculated ones). The result could look like this (theoretical along x-axis, actual along y-axis):

Chosen percentage values

Percentile plot		
Percentile	Theory	Actual
5.71%	0.025	0.058086
22.86%	0.125	0.151486
45.71%	0.225	0.258143
67.14%	0.425	0.447614
81.43%	0.575	0.627857
94.29%	0.975	1.011086

Values calculated with PERCENTILE



i. The last step is to conduct the significance test. To verify if the data is gamma distributed a chi-squared goodness of fit test is performed. The test follows these steps:

- State the null hypothesis H_0: The data can be modelled by gamma distribution
- (State the alternative hypothesis H_a: The data can not be mod…)
- Set up the test statistic X^2:

$$X^2 = \frac{(x - E_i)^2}{E_i}$$, for all i=1..n

where x is the actual- and E_i is the expected number of data values in interval i.
$E_I = f(t)_{theoretical} * n$, where the $f(t)_{theoretical}$ is the theoretical probability value in a certain interval, calculated before.

- Calculate the test statistic with the data and summarise all the results

$$\sum_{i=1}^{n} \frac{(x - E_i)^2}{E_i}$$

The result could look like this:

Actual	E_i	$(x - E_i)^2/E_i$
4	7.525196	1.651387
3	6.820546	2.140088
9	6.113608	1.36274
9	5.461922	2.291866
7	4.871859	0.929622
…	…	…
…	…	…
…	…	…

- Calculate the degrees of freedom
  deg. of freedom = n-1
- Check the fitness level with the Excel CHIDIST function. This tells how well the data fits to a gamma distribution, a value above 50% confirms the null hypothesis and the data can be assumed to be gamma distributed.
  CHIDIST(sum of test static, deg. of freedom)

  Hopefully a result like this is achieved:

$\Sigma (x - E_i)^2/E_i$	42.488113
Deg of freedom	47
Fit level	**65.97%**

A result of 66% clearly states that the data is gamma distributed

If this is not achieved the result can be improved by trying to remove the worst peaks, the result above is actually received after two peaks are removed. Note that the peaks should be removed from the raw data column and the whole process then have to be repeated. A quick and dirty version though is to only remove the peaks from last step (seems to give pretty similar results) but then changes of the parameters will be missed.

Now it has been shown that the data can be seen as gamma distributed, in fact the distribution is almost exponential. Though the data can change over time, this will be tested for in the next paragraph.

3 T testing of how data change over time

The distribution of the data might change over time, from gamma distribution to some other distribution. By performing a t test of data from two different time intervals it is possible to verify that the distribution does not change, or do change, over time. The t test is conducted by first gathering the data. In this project the test is done on disconnectors, where the data is the normalised cost per equipment for different models. The data is divided into two time periods, 1999-2001 and 2001-2003. These two populations of data are then compared to see if there are any changes over time.

As with the significance test for gamma distribution there is also a excel spreadsheet template to help with the test (Change Over Time Template.xls).

a. The data from the two periods is copied into two columns in an excel spreadsheet and then the columns are sorted so that the same objects from the different periods are matched with each other.
b. The differences, D_i, of the data values from the above matched objects, X_i and Y_i, are calculated:

$$D_i = X_i - Y_i$$

Equation 3-1 Difference of the data from two time periods

c. The next step is to estimate the variance, σ^2, for the differences with Equation 3-2.

$$\sigma^2 = \sum_{i=1}^{n} \frac{\left(D_i - D\right)^2}{n-1}$$

Equation 3-2 Estimation of variance

In Equation 3-2 D is a parameter calculated in Equation 3-3 below.

$$D = \sum_{i=1}^{n} \frac{D_i}{n}$$

Equation 3-3 Calculation of parameter D used to estimate variance

d. The significance test can now be performed by first setting up the null and alternative hypothesis.

- Null hypothesis, H_0: $\mu_x=\mu_y$, the two means are equal
- Alternative hypothesis, H_a: $\mu_x/=\mu_y$, the two means are not equal

- Set up the test statistic T:

$$T = \frac{D}{\left(\frac{\sigma}{\sqrt{n}}\right)}$$

Equation 3-4 Test statistic for t test

- State the level of significance, α, in this test the level of $\alpha = 0.1$.
- By using a level of significance of 0.1, α=0.1, H_0 will be rejected if the absolute value of T is greater than 1.697.
- Now calculate the test statistic and check whether or not H_0 can be accepted at the chosen significance level.
- Also another calculation should be performed, similar to the last one in the previous chapter when the fit level is decided. The internal excel function to be used is called TDIST. TDIST requires the absolute value from the test statistic, degrees of freedom and shape of curve as input variables.

TDIST(abs(T), deg. of freedom, 2)

The "2" stands for two-tailed distribution

- The result from this function should exceed 50%, otherwise H_0 should be rejected. To achieve a higher result values with high difference should be removed. In the template file models which are removed to achieve a high result has been highlighted with yellow.

It is recommended to check the distribution of the data every now and then, once a year might be enough. But since the t test showed that the data is not expected to change this might be done even more seldom. Since all data now can be said to be gamma distributed, a solution for how to find the limits has to be developed

4 How the threshold limits for the exceptions are calculated

The exceptions will be raised for objects in the top and bottom tail of the distribution. Since the distribution is almost exponential it is recommended to set the limits at 5% and 95%, using smaller limits would be to narrow. Since the distribution almost is exponential and μ/σ in general is one the following rules can be used:

$$\begin{aligned} &Lower\,threshold: \mu - 0.95\sigma = 5\% \\ &Upper\,threshold: \mu + 2\sigma = 95\% \end{aligned}$$

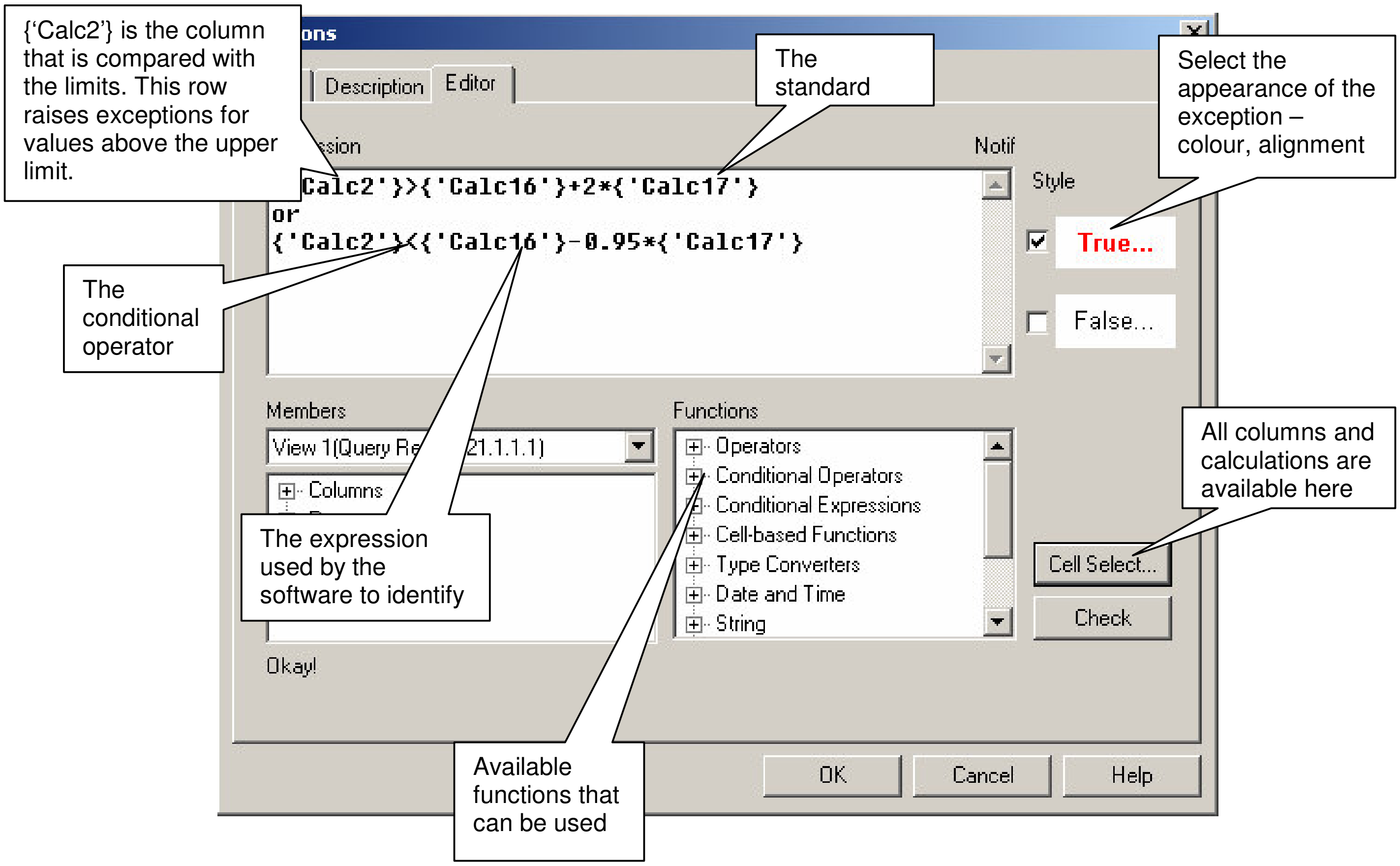
, where μ is the mean and σ is the standard deviation calculated when verifying the distribution of the data.

5 Implementation of exceptions in the reports

The way to calculate the exceptions in the report is just to use the rules from the previous chapter and implement them in the report. The reports have possibilities to calculate μ and σ directly with internal functions in the software.

The way to implement exception raising for a column in a report is first to create the values for μ and σ, and then to create an exceptions that will be applied to a column.

a. Right click the actual column and chose *Calc-O-Matic…* from the menu that appears.
b. Click on *Average* in the left text widow and then *Ok*. This will calculate the average and a new box with the average is added to the end of the column. This box can be deleted, the value still exists in the background and can be used for further calculations.
c. Again right click the actual column and chose *Calc-O-Matic…* from the menu.
d. This time click on *Standard Deviation* in the bottom of the left text box and *Ok*. The same procedure as with the average.
e. Now create the actual exception by clicking on *Tools* on the menu row.
f. Then click on *Exceptions* under the *Tools* menu (make sure the table is marked, otherwise *Exceptions* will be grayed out). This will bring up the exception window.
g. Create a new exception by clicking on the *New* button.
h. Press the *Description* tab and give the exception a name and a description of how it works.
i. Go into the *Editor* tab, which is where the calculations for the exception will be entered.
j. Click *Cell select…* , press the blue box after the text **(every Column)*. This will bring up a new window with all available columns in the table (also the recently calculated μ and σ are available there), click the column for which exceptions should be raised. This will copy a text into the editor window.
k. After the text inserted from j) enter the conditional operator that should be used, > (less than) for the lower – and < (greater than) for the upper threshold limit.
l. Enter the calculation of the limit after the operator. μ and σ are available by pressing *Cell Select…*
m. Since two limits are to be used, enter one expression on the first row, write *or* on the second row and then on the third row enter the last expression. This should look something like the expression in the *Expression textbox* in the picture below.



6 Implementation of exceptions in Genio Designer projects

The exceptions in Genio Designer are raised so that e-mails can be sent out with the exceptions. They are created in the same way as described above, the only difference is that μ and σ have to be calculated manually, that is there is no internal function for them.

More about the implementation of exceptions in Genio Designer is found in the AMT report *AMTREP116 Vxx SAP Notifications – End to end process for GENIO* (\\PQPVFS01\OPSpub\Asset Monitoring\Documentation\Reports\AMTREP116 SAP Notifications - End to end process for GENIO).