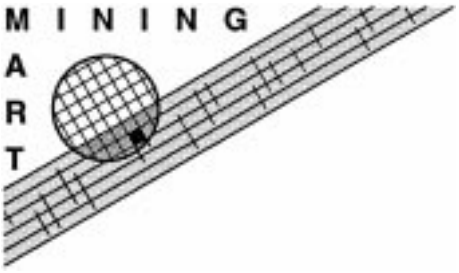

Enabling End-User Datawarehouse Mining
Contract No. IST-1999-11993
Deliverable No. D6.2

Deliverable 6.2
Data Sets, Meta-data and Preprocessing Operators at Swiss Life and CSELT

Jörg-Uwe Kietz[1], Regina Zücker[1], Anna Fiammengo[2], and Giuseppe Beccari[2]

[1] Swiss Life, CC/ITRD
Information Technology Research & Development
CH-8022 Zrich, Switzerland
{uwe.kietz,regina.zuecker}@swisslife.ch
http://research.swisslife.ch

[2] Cselt - Telecom Italia Group
Via Reiss Romoli 274, 10143 Torino, Italy
D-87293 Turikel, Germany
{Anna.Fiammnego,Giuseppe.Beccari}@cselt.it

June 23, 2000

Abstract

In this deliverable the Swiss Life data warehouse excerpts from deliverable 6.1a and 6.1b are described. and evaluation of the usefullness of the approach for CSELT. It is supplemented with a paper published in the Proceedings of the fifth Intenational Multistrategy learning Workshop (MSL-2000), which describes our view of the future use of the Mining Mart System and KDD-Cases.

Chapter 1

Introduction

This is deliverable no. D6.2. It describes the results of the work done by Swiss Life (chapter 2, by the first and second authors), and CSELT (chapter 3, by the third and fourth authors) for WP6, i.e. the first data sets and data mining problems which will be used as test cases for the future development of the "Mining Mart System".

The Swiss Life view of how this "Mining Mart System" supports data mining applications like the one's described in this deliverable is outlined in a paper presented at the fifth International Workshop on Multistrategy Learning, which is also included as appendix A of this Deliverable.

Chapter 2

Swiss Life

2.1 The data set in Deliverable 6.1a

This data set contains an excerpt from the 10 most important relations in our datawarehouse describing the relationship between Swiss Life partners, i.e. in particular customers, insurance contracts and components of insurance tariffs.

The relation 'part' contains all partners. The relations 'eadr' resp. 'padr' relates them to their electronical resp. postal addresses. Details about the households they are living in can be found in 'hhold'. Most of the household data is derived from the geographical region the household belongs to[1]. Each partner can play roles in certain insurance policies (relation 'vvert') which is realized by a relation 'parrol'. If a partner is the insured person of the contract then tariff role records (relation 'tfrol') specify certain further properties. An insurance contract can have several components (e.g. the main contract part plus a component for insuring the case that the ensured person becomes invalid) each of which (records in 'tfkomp') is related with a

[1]This household-relation corresponds to the houshold-relation in section 2.2. In this data-set these are the household-data supplementing the partners of Swiss Life, whereas in the data set described in section 2.2. these are the only data about the persons we have, as the mailing action is addressing new partners.

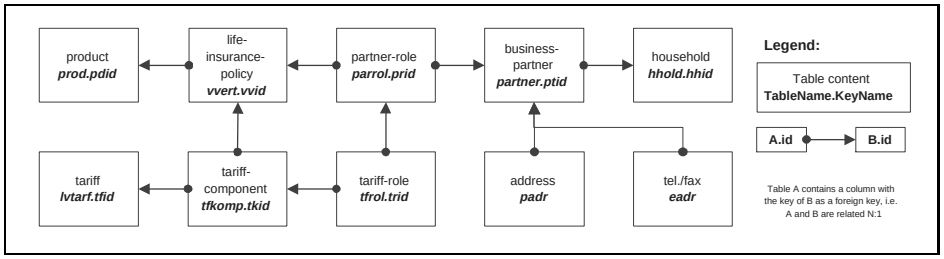

Figure 2.1: Data Schema Deliverable 6.1a

tariff role of the respective partners. Finally, each policy concerns a certain product ('prod') and tariff components are bound to dedicated insurance tariffs (relation 'lvtarf').

A graphical overview about the schema is given in fig. 2.1

2.1.1 Overview about files

The data set consists of the following files

```
Readme.txt     A textfile with the contents of this section
kddERdia.gif   Diagram of the schema similar fo fig.1
    .bmp
eadr.txt          505 records for relation eadr
hhold.txt       12934 records for relation hhold
padr.txt        17970 records for relation padr
parrol.txt     111077 records for relation parrol
part.txt        17267 records for relation part
tfkomp.txt      73502 records for relation tfkomp
tfrol.txt       73332 records for relation tfrol
vvert.txt       34986 records for relation vvert
lvtarf.txt        239 records for relation lvtarf
prod.txt          184 records for relation prod
aska.txt        17267 records for task A
taskb.txt       12934 records for task B
```

The data-files are in KEPLER Text-format (Tab-Text), a format readable by Kepler and most Database Systems.

2.1.2 Schema Description

Relation: vvert - Life Insurance Policy

vvid:NUMBER insurance policy number

pdid:NUMBER identifier of the product the insurance contract is bound to

vvstacd:NUMBER status code of the contract

vvinkzwei:NUMBER modus of payment for premium rates

vvbeg:NUMBER year of contract start (earliest start date among all tariff components) see attribute tkbeg of relation tfkomp

vvend:NUMBER year of contract end (last date among all tariff components)

vvinkprl:NUMBER yearly overall premium to be payed by the policy holder sum over all tariff components; see attribute tkinkprl of relation tfkomp

vinkpre:NUMBER single premium to be paid at contract start sum over all tariff components; see attribute tkinkpre of relation tfkomp

vvwae:NUMBER currency for payments

vvversart:NUMBER type of insurance contract, e.g. capital insurance, pension insurance

vvaendart:NUMBER type of last modification

vvaendat:NUMBER year of last modification

vvabvb:NUMBER responsible agent

vvabga:NUMBER responsible general agency

vvstifcd:NUMBER place where policy is left; most (own) employee's contracts are kept in a collective depot

vvvorscd:NUMBER type of (official Swiss) precaution type of the contract

vvbvgcd:NUMBER slot for storing whether the insured person participates in a company's pension insurance scheme

vveucd:NUMBER code for included disablement insurance; depends on respective tariff components

Relation:parrol - Partner Roles

prid:NUMBER role identifier

ptid:NUMBER partner identifier; refers to attribute ptid of relation partner

vvid:NUMBER identifier of the insurance contract where the partner ptid plays the role prid

prtyp:NUMBER type of role, i.e.

- **6** - special agent,
- **9** - premium payer,
- **10** - pension receiver,
- **11** - insurance holder,

12 - insured person,

14 - pledgee,

15 - swiss representative,

17 - legal representative,

18 - guardian

prtypnr:NUMBER role number wrt. type and contract, e.g. insured person 2

Relation:part - Partners Involved in Insurance Contracts

ptid:NUMBER partner identifier

hhid:NUMBER household identifier (see attribute hhid of relation hhold)

epberuf:NUMBER profession code

eperweart:NUMBER professional status (independent or employed)

epgebudat:NUMBER year of birth

epsta:NUMBER personal state, i.e. nothing special (0), dead (1), missed (2) or under revision as disabled person (3), disabled person (4)

eplnamecd:NUMBER name code for married persons

epsexcd:NUMBER sex: 1 - male, 2 - female

epzivstd:NUMBER marital status: 0 - unknown, 1 - single, 2 - married, 3 - widowed, 4 - divorced, 5 - separated

Relation:tfkomp - Tariff Components

tkid:NUMBER component identifier

vvid:NUMBER insurance contract number the component belongs to (see attribute vvid of relation vvert)

tknr:NUMBER (running) component number wrt. contract

tfid:NUMBER number of tariff the component is bound to; see attribute tfid of lvftarf

tkbeg:NUMBER starting date of tariff component

tkend:NUMBER end date of tariff component

tkexkzwei:NUMBER payment mode (payments per year in particular for pension insurances)

tkstacd:NUMBER status of the component (regular premium (1) or single premium (2)

tkleist:NUMBER insured benefits of the component

tkinkprl:NUMBER regular premium for the component

tkinkpre:NUMBER single premium for the component

tktarpra:NUMBER calculated (internal) premium

tkuebvwsysp:NUMBER type of profit spending for running contracts

tkuebvwsysl:NUMBER type of profit spending during pension payment period

tkprfin:NUMBER current component state wrt. premium: required or not

tkdyncd:NUMBER states whether tariff component is dynamic (i.e. regular adaption of premium dependent on inflationary rate e.g.) 0 - no, 1 - yes, 2 - not applicable or unknown

tkausbcd:NUMBER states whether extension guarantee (higher insured sum without additional risk check) was claimed for tariff component 0 - no, 1 - yes, 2 - not applicable or unknown

tkrauv:NUMBER type information for the special case of risk extension and exchange insurances

tksmed:NUMBER states special medical (risk) cases

tkrizucd:NUMBER extra premium charge for certain risk cases 0 - no, 1 - yes, 2 - not applicable or unknown

tklfleist:NUMBER (regular) current payment (dynamically increasing)

tktodleista:NUMBER payment for death of insured person

tkerlleista:NUMBER (initial) payment for regular contract termination

tkrenleista:NUMBER (initial) payment of pension

tkeuleista:NUMBER (initial) payment for the case of disablement

tkunfleista:NUMBER (initial) payment for the case of accident

Relation:tfrol - Tariff Roles

trid:NUMBER tariff role identifier

prid:NUMBER partner identifier (see relation partner)

tkid:NUMBER link to tariff component (see attribute tkid of relation tfrol)

trteceinal:NUMBER (technical) age of insured person at contract agreement

truntcd:NUMBER states whether medical risk check was necessary 0 - no, 1 - yes, 2 - not applicable or unknown

trklauscd:NUMBER states whether special contract clause was negotiated 0 - no, 1 - yes, 2 - not applicable or unknown

trstafcd:NUMBER states whether staggered payment was negotiated 0 - no, 1 - yes, 2 - not applicable or unknown

trricd:NUMBER states that premium surcharge for higher risks was set 0 - no, 1 - yes, 2 - not applicable or unknown

Relation:eadr - electronical address

ptid:NUMBER partner identifier (see attribute ptid of relation partner)

azart:NUMBER type of address usage (e.g. private, business)

eatyp:NUMBER type of address (e.g. Fax, Phone etc.)

Relation:padr - postal address

ptid:NUMBER partner identifier (see attribute ptid of relation partner)

azart:NUMBER type of address usage (e.g. private, business)

kanton:STRING district (inside Switzerland)

gbeadmgeb:NUMBER region

relation: hhold - household data

hhid:STRING household identifier

nknt:STRING

nagl:STRING

nlnggrz:STRING

nwmggbt:STRING

nzstga:STRING

nzstrvz:STRING

cmig:STRING

cstrart:STRING

nlngcdeh:STRING

nsexcde:STRING

cakszvl:NUMBER

cakl05:NUMBER

cakl10:NUMBER

cakl20:NUMBER

nebfcde:STRING

nbgrcde:STRING

nbrccde:STRING

cfnkstf:STRING

nanzvrm:NUMBER

nkptsum:NUMBER

nqlf:STRING

nekmin:NUMBER

nekzwv:NUMBER

neknbn:NUMBER

nvrmg:NUMBER

nkflkls:NUMBER

clngstm:STRING

cantfzg:STRING

nfrzhlt:STRING

csgfzhg:STRING

czvgsgf:NUMBER

ngbd:NUMBER

nrgntyp:STRING

nznt:STRING

neinkls:STRING

ngbdart:STRING

nbauprd:STRING

nanzwhr:STRING

nwhgeig:STRING

nfnkstf:STRING

nhhltyp:STRING

nhhlgrs:STRING

Relation:prod - Product

pdid:NUMBER product identifier

pdbest:NUMBER source of product information

pdlokid:STRING

pdtext:STRING

Relation:lvtarf - life insurance tariff

tfid:NUMBER tariff identifier

tfbest:NUMBER source of tariff information

tflokid:STRING

tftext:STRING

2.1.3 Mining Tasks

The following 3 tasks should be tackled with arbitrary Data Mining, Statistics or Machine Learning algorithms applied to the data set. Please note, that the main focus lies on the preprocessing steps which make the algorithms applicable rather than on achieving high-precision mining results.

Tasks A and B are concept learning (resp. classification or deviation detection) tasks and aim at finding concept descriptions for subsets of partners (A) and households (B). The tables 'taskA' and 'taskB' relate the respective identifiers 'ptid' (A) and 'hhid' (B) with a value indicating concept membership.

taskA - class membership wrt. concept in relation task A

Try to find a concept description for those partners (attribute 'ptid' of 'taskA') who have value '1' for attribute 'class'.

ptid:NUMBER partner identifier (see attribute 'ptid' of relation 'part')

class:NUMBER 0 - not applicable, 1 - Yes, 2 - No

taskB - class membership wrt. concept in relation task B

Try to find a concept description for those households (attribute 'hhid' of 'taskB') who have value '1' for attribute 'class'.

hhid:NUMBER household identifier (see attribute 'hhid' of relation 'hhold')

class:NUMBER 0 - not applicable, 1 - Yes, 2 - No

taskC - Find Clusters of typical insurance behaviour

Task C is a clustering task intended to relate properties of households with the features of their insurance contracts.

Try to find clusters on 'hhold' such that members of a cluster resemble each other not only wrt. household attributes but also wrt. the insurance policies they are involved in.

2.2 Data, Meta-Data and Support-Code in Deliverable 6.1b

For Swiss Life there are several application areas where central business-cases could be supported by data mining [Staudt *et al.*, 1998], especially:

- Marketing
- Product development and controlling

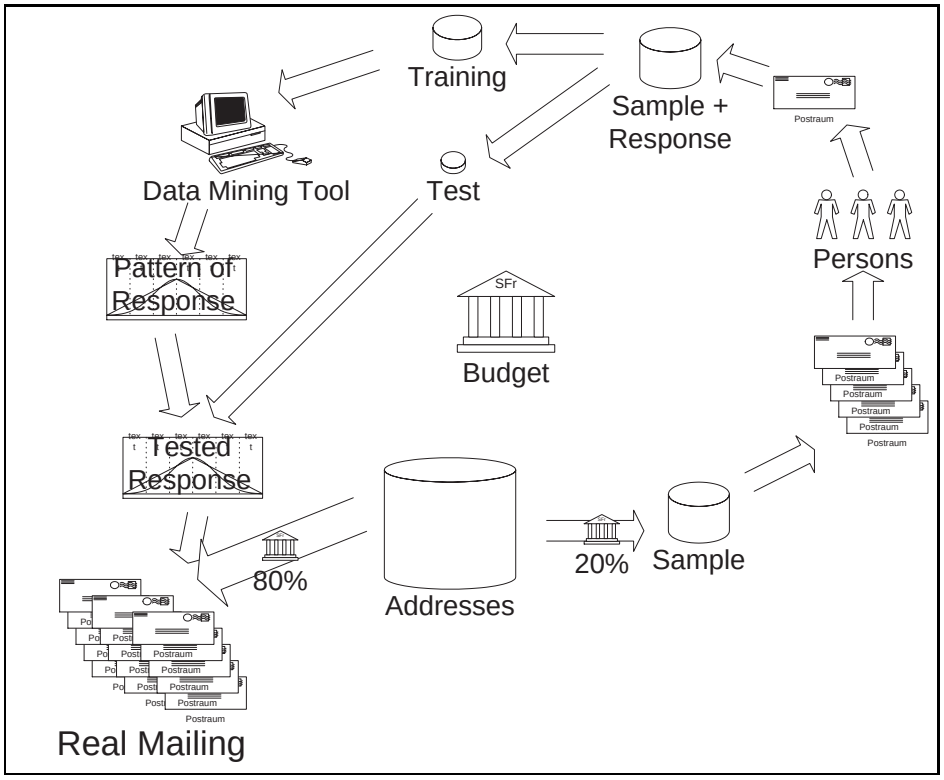

Figure 2.2: The Business Case Mailing Action

- Business reporting

As a first application, where we have data available we will use the optimization of responses to mailing actions in direct marketing as an well known example of such a problem. This business-case is illustrated in figure 2.2[2].

If we describe this business-case on a more technical level we come to the following steps:

0. Use an existing data warehouse (DWH) as base.

1. Construct a household view on this DWH, which provides all relevant information in a form, that allows the next step to be done by an end-user.

2. Select the target segment, e.g. households,

   (a) which have a child with an age below 2 and which are not mailed since the birthday of this child, or

[2]Ling and Li [Ling and Li, 1998] describe another problem and solution analysis of this business-case. However their analysis is based on the assumption, that existing customers are persons answering to mailings, which is not the case in our setting, where most contracts are still sold by insurance-agents and not by mailing-actions.

   (b) which have already bought a single-premium insurance, but this is more than two years ago, or
   (c) ...

3. select a random-sample, with size proportional[3] to 20% of the budget, i.e. generate a sample to gather labels for the training and test set.
4. Export the addresses of this sample, do the first mailing, and store the responses, i.e. label the sample.
5. Split the sample into training and test-set.
6. Select/Construct the relevant attributes for the current response prediction task.
7. Train the selected mining-tool, which output could be used to order (not just classify) the data.
8. Apply the data-transformations (pre-processing) done in step 6 to the test-data as the mined pattern relies on it.
9. Test the mined pattern on the test-set, i.e get an estimated response-rate for the mined pattern on the target-group.
10. Apply the data-transformations (pre-processing) done in step 6 to the target-segment as the mined pattern relies on it.
11. Select the best (ordered by the mined reponse-pattern) records (proportional to 80% of the budget) from the target segment of step 1.
12. Export the addresses of this selection and do the real mailing.
13. Compute a final evaluation, and store all the mailing-information (date, (non-) responses, segment, product, mined pattern, data-transformations, evaluation, ...) in the DWH/DWH-meta-data-Repository, such that it could be used as background knowledge for further actions.

The data and the meta-data based preprocessing in deliverable 6.1b which are described in the following subsections correspond to the step 0 (base tables) and step 1 final household view generated by the meta-data based preprocessing.

2.2.1 Data in Deliverable 6.1b

Following there is a description of all basis tables within the data-set of Deliverable 6.1b.

[3] If your budget is X and a single letter cost Y you can send a total of X/Y letters, spending 20% of the budget to get the training/test data means to select X/Y*0.2 household-records from the target segment.

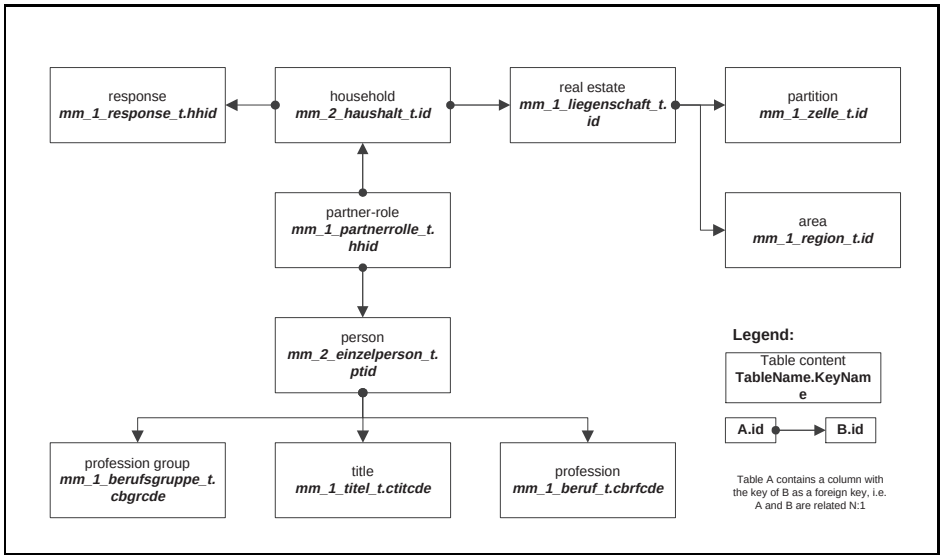

Figure 2.3: Data Schema Deliverable 6.1b

The naming of tables and views follows a convention. Every database object has the prefix 'MM_'. Tables have the suffix '_T', views the suffix '_V' and snapshots the suffix '_S'. The prefix follows a number which shows a hierarchy for creation.

'1' means that the table content is exactly the same as in the original data warehouse MASY. The attributes are only changed in the way of applying functions on it (e.g. decode-function). The number of the data records is unchanged.

'2' has the same meaning as '1' except that these tables have a reduced number of data records compared to the original table. Only the relevant data for the mailing task is within these tables.

'3' and more means that the attributes of these views or tables are based on the basis tables (marked with '1' or '2').

The basis tables contain only attributes, which have useful values in the original table of the data warehouse, i.e. attributes are dropped, which are currently of too bad quality.

A graphical overview about the schema is given in fig. 2.3

Table MM_2_HAUSHALT_T This table contains basic information about a household, for example income and assets of a household, double income, vehicle type and so on.

Table MM_1_LIEGENSCHAFT_T This table contains basic information about real estates, in which a household is located. Mainly it gives information about the type of building.

Table MM_1_REGION_T This table contains information about the area in which the estate is located. E.g. the zip code, district within Switzerland, etc.

Table MM_1_ZELLE_T This table contains information about a partition in which the estate is located. E.g. it gives estimated information about what kind of buildings and apartments dominate in this partition, what profession has the head of the household, how much money spends the household on health, food, etc.

Table MM_2_EINZELPERSON_T This table contains all information about a person. It gives the age and sex of a person, the income and so on.

Table MM_1_BERUF_T This table gives information about the profession of a person.

Table MM_1_BERUFSGRUPPE_T This table contains the profession group for a profession.

Table MM_1_TITEL_T This table contains the title of a person, e.g doctor or professor.

Table MM_2_PARTNERROLLE_T This table contains on the one hand information about the type of a person. If attribute 'ROLLENTYP' has the value '22' or '24' then the person is the head or partner of the household. If the value is '23' the person is a household member. On the other hand this table contains the relationship between a person and his contracts.

Table MM_1_RESPONSE_T This table contains mainly information about the mailing action and the responses. If the attribute 'RESPONDED' has the value 'NO' together with the attribute 'CHANGE_DATE' it gives information, when a mailing is sent to the household. If there exists a record with 'RESPONDED' value 'YES' and 'CHANGE_DATE', then the household has responded to the mailing. So, each household, which has responded, has two records within this table.

2.2.2 Meta-Data in Deliverable 6.1b

The implementation of the preprocessing operators in SQL is based on the following schema:

```
Sampling & segmentation
 view -> view (row reduction)

Feature Selection
```

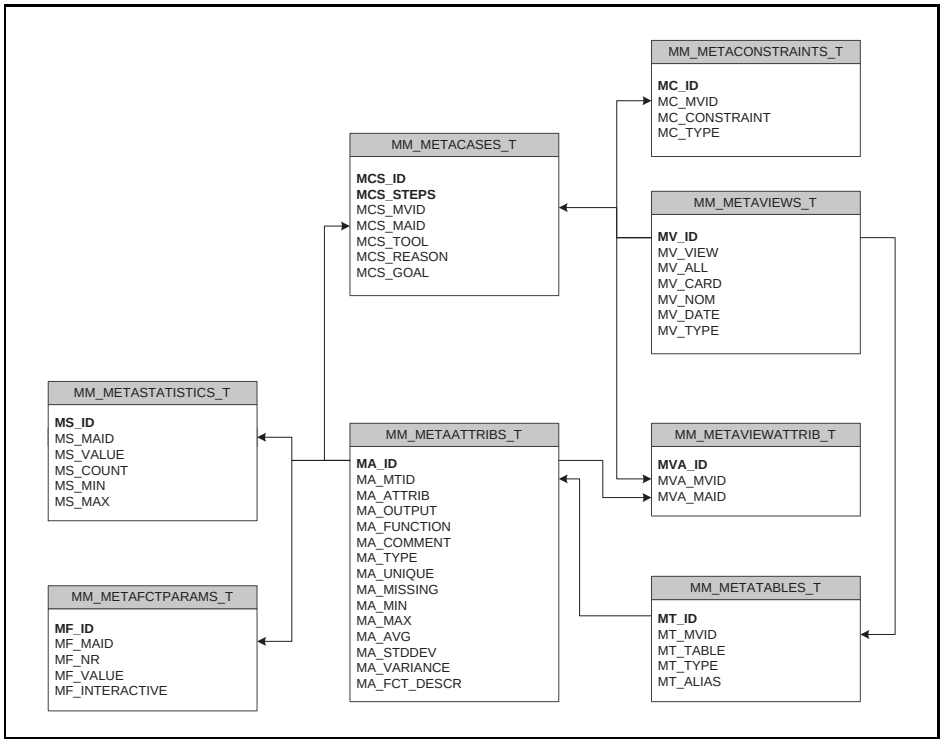

Figure 2.4: Metadata Schema

```
 view -> view (column reduction)

Attribute construction
 view.c1 X view.c2 X .... X view.cn -> view.cnew

Multi-relational attribute construction
 (select <v1.aggregate>
    from view, v1
   where v1.id = view.id
   group by view.id)
 -> view.cnew
```

The Metadata Schema used so far is shown in Figure 2.4. Besides the information needed to generate the views, it also contains the basic univariate statistics of the used views and attributes. For working with the Metadata several tools are developed which are described following.

For creating views a generator has been developed, which reads the necessary information from the Metadata and creates an object within the database. The generator (generator.jar) is realized in Java jdk1.2.2 using embedded SQL (package java.sql). The function call is:

java CreateView database-name user password view-name
The parameter 'view-name' has to be the same string as in attribute MV_VIEW. Otherwise the generator cannot find the right view. Dependent on the value within MV_TYPE the generator creates a view, table or snapshot. It's necessary to set the environment variable CLASSPATH to the path, where the jar-file is stored.

For creating statistics for views there exists a stored procedure within the database. It's realized with dynamic PL/SQL and calculates attribute values, e.g. how many values or different values an attribute has. The function call, using the tool SQL*Plus, is:
exec CreateStatistics('view-name')
Again the parameter 'view-name' has to be the same string as in attribute MV_VIEW.

To visualize the calculated statistics an EXCEL-macro has been developed. This macro connects itself to the database, reads the statistical information for every attribute of the requested view, creates a diagram and stores it within a file. It's decidable if only one or all views should be visualized. Also, you can define the path where to store the created diagrams. This information as well as the database-string and the user-password-string are asked via input boxes within the macro. The name of the file and the macro are:
Statist1.xls, StatistikGrafik
As described before, when defining the view-name it has to be the same string as in attribute MV_VIEW. The database string has be the database name, the user-password-string has to have the format user/password.

Table MM_METACASES_T

This table is used to store all information about a case. The whole case can be executed if every step is applied sequentially in order of their number either to a view or an attribute. Within a step the reference to a view but not to an attribute means that every attribute, which belongs to the view (table MM_METAVIEWATTRIB_T) is applied by the step. Referencing a view and an attribute means, that just this attribute of a view is relevant. Referencing only the attribute means, that the basis-attribute is used without a view.

- MCS_ID
  Id of a case. Cannot be NULL.
- MCS_STEP
  Number of a step, which has to be applied to execute the case. Cannot be NULL.
- MCS_MVID

Foreign key referencing the id of the table MM_METAVIEWS_T. Can be NULL if MCS_MAID is filled.

- MCS_MAID
  Foreign key referencing the id of the table MM_METAATTRIBS_T. Can be NULL if MCS_MVID is filled.
- MCS_TOOL
  Function call with parameters to start an external Mining tool.
- MCS_REASON
  Comment to describe the reason why this step is necessary.
- MCS_GOAL
  Comment to describe the goal of this step.

Table MM_METAVIEWS_T

This table stores the name of a view and some statistical information about it. Actually the object, which is named a view can also be a table or snapshot. The difference is declared in attribute MV_TYPE.

- MV_ID
  Id of the view. Cannot be NULL.
- MV_VIEW
  Name of the view. Cannot be NULL.
- MV_ALL
  Total number of records within the based table or view.
- MV_CARD
  Number of cardinal attributes belonging to the view.
- MV_NOM
  Number of nominal attributes belonging to the view.
- MV_DATE
  Number of date attributes belonging to the view.
- MV_TYPE
  Type of the view which is the suffix of the created object. Possible values are '_V' (view), '_T' (table), '_S' (snapshot).

Table MM_METAVIEWATTRIB_T

This table is used to assign attributes to a view. An attribute can belong to several views.

- MVA_ID
  Id of the record. Cannot be NULL.
- MVA_MVID
  Foreign key referencing the id of the table MM_METAVIEWS_T. Cannot be NULL.
- MVA_MAID
  Foreign key referencing the id of the table MM_METAATTRIBS_T. Cannot be NULL.

Table MM_METATABLES_T

This table stores the table name needed to create a view. Like explained in MM_METAVIEWS_T, the so called table can also be a view or snapshot. It's possible to store more tables for one view.

- MT_ID
  Id of the table. Cannot be NULL.
- MT_MVID
  Foreign key referencing the id of the table MM_METAVIEWS_T. Cannot be NULL.
- MT_TABLE
  Name of the table, view or snapshot.
- MT_TYPE
  Type of the table. Possible values are '_V' (view), '_T' (table), '_S' (snapshot).
- MT_ALIAS
  Alias for this object. E.g. if there are ambiguous attribute names when creating a view over joined tables, the alias name is necessary.

Table MM_METACONSTRAINTS_T

This table contains all constraints which are necessary to create a view. Especially if there are more tables used for a view, the join between them must be stored here. Besides also grouping conditions can be stored in this table. The constraint conditions will be executed in order of the id.

- MC_ID
  Id of the constraint. Cannot be NULL.

- MC_MVID
  Foreign key referencing the Id of the table MM_METAVIEWS_T. Cannot be NULL.
- MC_CONSTRAINT
  String containing the constraint condition.
- MC_TYPE
  String containing a grouping function, e.g. 'GROUP BY' or 'HAVING'. There can be more grouping functions defined, but they always have to be the last conditions within all constraint conditions. E.g. it's not possible to define a constraint with id 2 which is a normal join and another constraint with id 1 with type 'GROUP BY'.

Table MM_METAATTRIBS_T

First this table contains every attribute with some statistical information which exists in the original data warehouse, no matter whether it is used in a view or not (table MM_METAVIEWATTRIB_T). New generated attributes during preprocessing are also stored here.

- MA_ID
  Id of the attribute. Cannot be NULL.
- MA_MTID
  Foreign key referencing the id of the table MM_METATABLES_T. Cannot be NULL.
- MA_ATTRIB
  Original name of the attribute, e.g. the attribute name used in the data warehouse. Cannot be NULL.
- MA_OUTPUT
  Name of the attribute after the preprocessing function is applied. Cannot be NULL.
- MA_FUNCTION
  Name of the SQL function which is applied to the attribute. The function must exist in the database. If this attribute has a value, the parameters of the function must be stored in the table MM_METAFCTPARAMS_T.
- MA_COMMENT
  Comment describing the attribute.
- MA_TYPE
  Type of the attribute. The only possible values are KEY, NOMINAL, CARDINAL, DATE.

- MA_UNIQUE
  Number of different values within the based table or view.
- MA_MISSING
  Number of missing values within the based table.
- MA_MIN
  Minimal value of the attribute within the based table.
- MA_MAX
  Maximal value of the attribute within the based table.
- MA_AVG
  Average value of the attribute within the based table.
- MA_STDDEV
  Value for standard deviation of the attribute within the based table.
- MA_VARIANCE
  Variance value of the attribute within the based table.
- MA_FCT_DESCR
  If a function is used for this attribute, it can be described why the function is used.

Table MM_METASTATISTICS_T

In this table the distribution of each NOMINAL, CARDINAL and DATE attribute is stored. For a nominal attribute every value is counted and grouped. For a cardinal attribute the values are grouped in at last 1000 blocks. For a date attribute the distribution depends on the months between the minimal and maximal value. If months_between > 600 the values are grouped into years. If months_between > 60 and <= 600 the values are grouped into quarters. If months_betweeen > 3 and <= 60 the values are grouped into months. If months_between <= 3 the values are grouped into days.

- MS_ID
  Id of the record. Cannot be NULL.
- MS_MAID
  Foreign key referencing the id of the table MM_METAATTRIBS_T. Cannot be NULL.
- MS_VALUE
  Name of one distribution block, e.g 'YOUNG' for the attribute 'AGE'. If the attribute is of the type CARDINAL, the average value of the block is used.

- MS_COUNT
  Number of counted records for this distribution block.
- MS_MIN
  Minimum value of the distribution block.
- MS_MAX
  Maximum value of the distribution block.

Table MM_METAFCTPARAMS_T

This table contains the parameters of a SQL function used for an attribute.

- MF_ID
  Id of the record. Cannot be NULL.
- MF_MAID
  Foreign key referencing the id of the table MM_METAATTRIBS_T. Cannot be NULL.
- MF_NR
  Order of the parameters, e.g. if the parameter is used as the first, second, etc.
- MF_VALUE
  The parameter itself. This can be a number or a string.
- MF_INTERACTIVE
  Boolean flag if the parameter can be filled interactively by a user.

Chapter 3

CSELT

3.1 Evaluation of preprocessing operations on other data mining tasks

The preprocessing operations furnished by Swisslife are also suited to Cselt mining applications.

Random and stratified sampling, typical/atypical cases, clustering and segmentation are often used to prepare input data for the learning algorithms. Furthermore, variable selection and construction, as a mean of reshaping the available information, are also useful in several mining analyses of which aim is discover customers' behavioral model.

As a matter of the fact, industrial tools support different components that allow the analyst performing many preprocessing operations. Unfortunately, these modules rarely exhibit user friendly interfaces, thus resulting in an increased amount of time spent by the analysts to prepare his/her data. For instance, variable construction is poorly supported by typical industrial mining applications, and efforts in this direction are securely appreciated.

Cselt analysts usually make their own ad-hoc procedures with proper languages in order to construct the data they need.

To the knowledge of the authors, neither metadata have been designed for mining purposes, and any ad-hoc procedure has been generalized within a proper framework. For this reason the schema in [Kietz2000] results in a useful effort for the definition of metadata for the Mining Mart project. At the moment, metadata are exploited only for DataMart construction to describe the loading process and how it performs at run-time.

The mining process proposed by Swisslife, detailed in [Kietz2000], is adequate for Cselt analysis; in fact, it can be compared to other approaches that have inspired our analysts for years [Galdino98], [CRISP99], [SASEM].

3.2 A fraud detection data mining task

The number of TICKET a Telecommunication Operator has to tackle is incredible large. Usually, million of customer exploit the services it offers, thus also the dimension of the database (DataWarehouse/DataMart) containing customer information are considerable. If each customer makes 3 call/day and there are 3 million of customers for a certain phone service, 9 million of TICKET/day (Firewall Log analysis presents analogous problems) will be generated, and analyzed. With the goal of ameliorating the performance of the fraud detection system, its selection capabilities, and adapt its behavior to the dynamic of the TICKET generation, it is often necessary to perform off-line mining analysis over historical TICKET properly memorized. Actually, monitoring and detection systems are difficult to control. If the supervisor does not select narrow filtering policy, the entire antifraud infrastructure will be flooded by alarms that could not be properly handled. Overload occurs and possible shutdowns could also be required. OLAP and Data Mining are exploited as decision support mechanisms both to understand how the detection system performs and eventually which kinds of changes are required to ameliorate its behaviors. This analysis exploit TICKET tables, information about customers, and the signals produced by the monitoring and detection systems in a given time window (e.g. a few months). A DDS is exploited to analyses these data.

A classical "star schema" has been used to implement the DM included in the DDS: a fact table and a collection of dimension tables are implemented as storage support. The fact table contains detailed numerical information whereas the dimension tables contain data about time, geographical area, costs, rate, etc.

3.2.1 Structure of the data

The fact table of the TICKET DM (see Figure 3.1) holds the TICKET sequence generated by the customers of a telecommunication service. Let consider a CDR flow, the TICKET will contain the following attributes:

1. Caller
2. Call data/time
3. Call length
4. Call cost
5. Callee
6. Direction.

The dimension tables are:

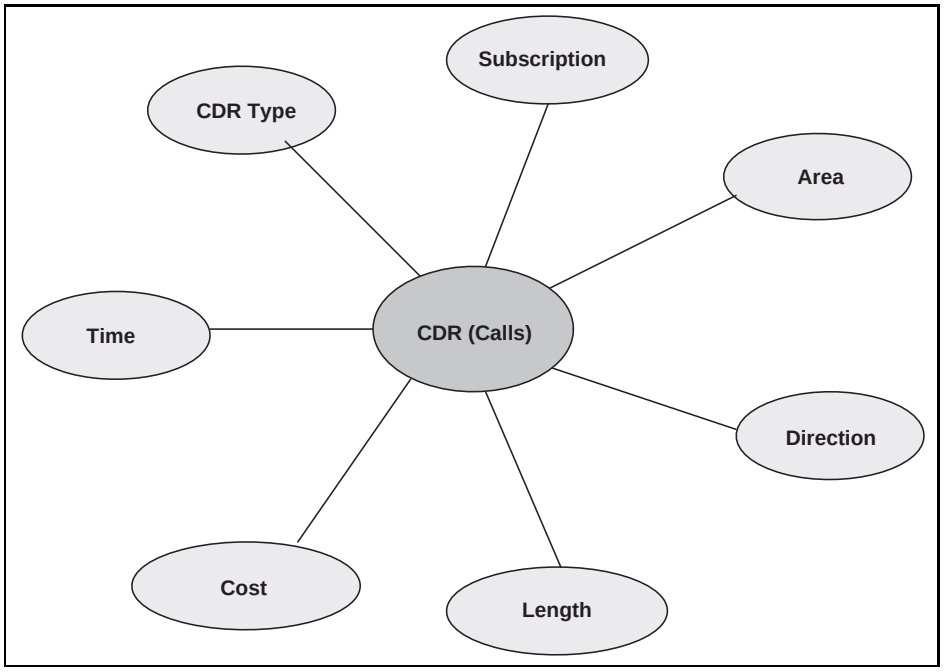

Figure 3.1: Star-Schema

1. Subscription contains customer information;
2. Call type: identifies the call type, i.e. national call, call origin/destination roaming call, follow me, (there are 5 possible classes);
3. Time: maintains the date/time of the call, it is structured with the following hierarchy:
4. Direction: contain information about the network used by the callee (telephone, cellular, international, special number, etc.); there are about 260 classes, namely one for any country code plus other dedicated classes; the direction posses a hierarchy the usually depend upon the network of the callee:
   - for the national network, the hierarchy is region, province, district;
   - for international network, the hierarchy is group of nation, nation;
   - for mobile network, the hierarchy is operator, network type, subscription type.
5. Area: indicates the caller position in the case of mobile network (there are 15000 different classes) The hierarchy depends upon the caller position:

- if the caller is within the country border, the complete hierarchy of all network plants that support the connection, is available;
- if the caller is abroad, the hierarchy is group of nation, nation.

6. Length: associated to each call there is its effective call length and a length class (20 level)

7. Cost: associate with each call there is its effective cost and a cost class (20 level)

More details of this data mining application can be found in deliverable D1 of WP1.

Bibliography

[Ling and Li, 1998] C. X. Ling und C. Li. Data mining for direct marketing: Problems and solutions. In R. Agrawal und P. Stolorz, editors, *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pages 73–79. AAAI Press, 1998.

[Staudt *et al.*, 1998] M. Staudt, J.-U. Kietz, und U. Reimer. A data mining support environment and its application on insurance data. In R. Agrawal und P. Stolorz, editors, *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pages 105–111. AAAI Press, 1998.

[Galdino98] J. Galdino, P. Tamayo, "Credit Risk Assessment using Statistics and Machine Learning Basic Methodology and Risk Modeling Applications", Procs. of Computational Economics '97, February 1997 (http://www.oracle.com/).

[Kietz2000] J.-U. Kietz, R. Zucker, "Mining Mart: Combining Case Base Reasoning and Multi-Strategy Learning into a Framework to reuse KDD-Applications", ..., May 2000.

[CRISP99] P. Chapman, R. Kerber, J. Clinton, T. Khabaza, T. Reinartz, R. Wirth, "The CRISP-DM Process Model", 1999 (http://www.spss.com/).

[SASEM] A SAS White Paper, "Finding the solution to Data Mining", The SAS Institute, 2000 (http://www.sas.com/).

Chapter 4

Appendix A

Jörg-Uwe Kietz and Regina Zücker: "MINING MART: Combining Case-Based-Reasoning and Multi-Strategy Learning into a Framework to reuse KDD-Application", In: P. Brazdil and R. Michalski (ed.): "Proc. of the fifth International Workshop on Multistrategy Learning", 5-7 June 2000, Guimares, Portugal, 2000.

Chapter 5

Appendix B

This Appendix lists the meta-data describing the views generated for the data in Deliverable 6.1b, and supplements the table description of section 2.2.1 in this document.

Chapter 6

Appendix C

This Appendix lists the fields contained in all the tables and views generated for the data in Deliverable 6.1b, and supplements the table description of section 2.2.1 and the view description of Appendix B in this document. Base table (section 2.2.1) attribute descriptions are direct copies of the Data Warehouse data dictonary description, therefore the comments are mostly in german. However, for all newly generated attributes (attributes of the views in appendix B) we have also given english comments.