MINING MART

Enabling End-User Datawarehouse Mining
Contract No. IST-1999-11993
Deliverable No. D5

Domain Knowledge and Data Mining Process Decisions

Arno Knobbe[1], Adriaan Schipper[1], and Peter Brockhausen[2]

[1] Perot Systems Netherlands
PO Box 2729 (Hoefseweg 1)
NL–3800 GG Amersfoort, Netherlands
{arno.knobbe,adriaan.schipper}@ps.net
http://www.perotsystems.com

[2] University of Dortmund
Computer Science, LS VIII
D–44221 Dortmund, Germany
brockhausen@ls8.cs.uni-dortmund.de
http://www-ai.cs.uni-dortmund.de

June 23, 2000

Contents

Chapter 1

Introduction

This document describes the role of Domain Knowledge in the Data Mining process, which Mining Mart supports. It is intended as the deliverable for work package WP5. In more detail, we will be examining how informal knowledge about the application domain, can be formalised into a set of Domain Knowledge Elements. These DKE's can then be used to determine a set of predefined Mining Process Decisions. We will describe in detail how each DKE contributes to the MPD's necessary to shape a single Data Mining project.

On an abstract level the role of Domain Knowledge in the Data Mining process is outlined in figure 1.1. There are basically three streams of information relevant in the process of which only the first one is examined in detail in this report:

- Domain Knowledge. This is the stream that starts in the top left corner. It describes the steps mentioned above.
- Mining Experience. This stream starts in the top right corner. It is concerned with using experience from prior projects derived by meta-level learning from detailed descriptions of these projects. The necessary environment for storing such a case-base of projects is defined and filled during work packages WP6, WP10 and WP16.
- Database. This stream starts in the bottom left corner. The idea is to use heuristics to have a first glance at the data and thus determine the best approach or set parameters. This line of thought is pursued in work packages WP4, WP13, WP14 and WP15.

As mentioned before, WP5 is limited to the Domain Knowledge stream. However because this stream is an integrated port of the whole environment there will be overlap with other work packages. Where it occurs, such overlap will be described, but not examined in detail. The emphasis is on correct

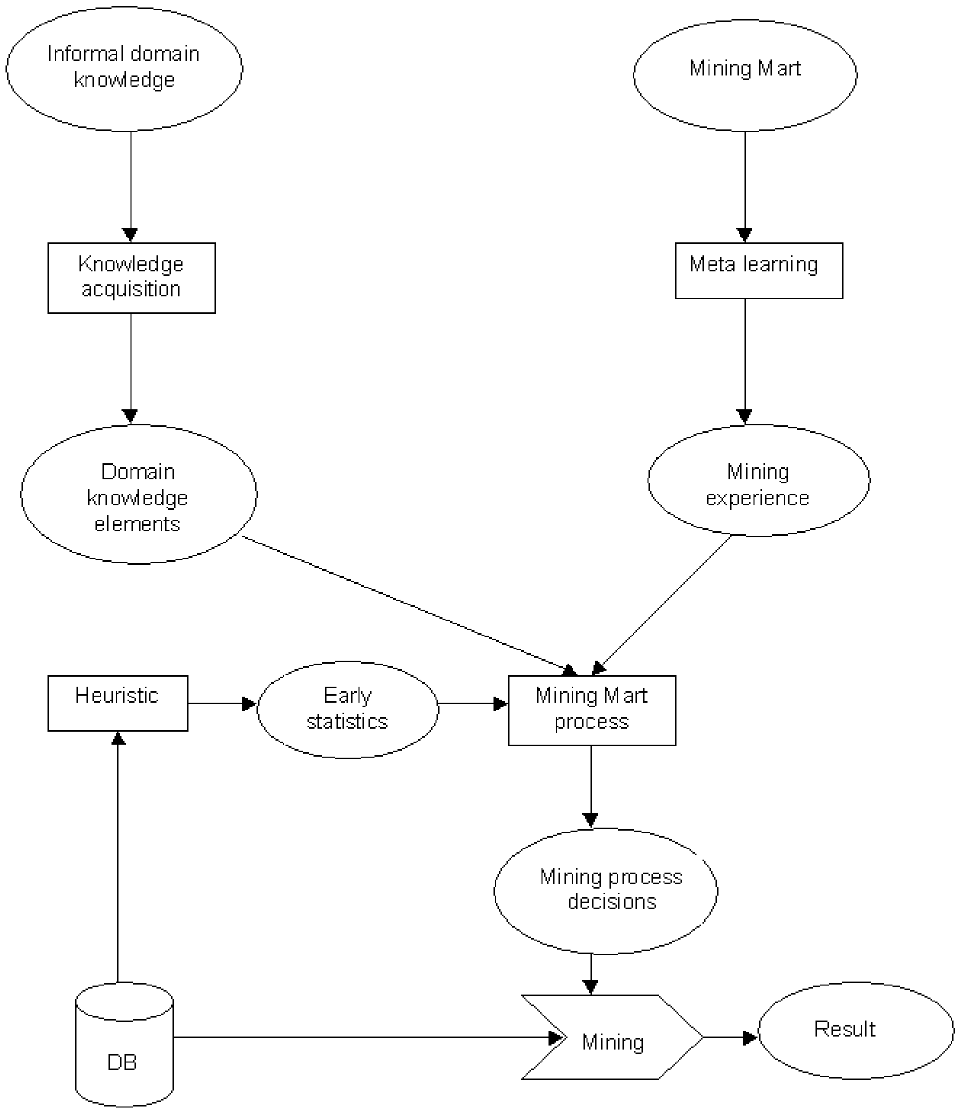

Figure 1.1: Information streams in Data Mining Process

interfacing rather then repeating work. The same is true for the follow-up work package WP19: Problem Modelling.

Chapter 2

Domain Knowledge Process

2.1 What is Domain Knowledge?

The question what domain knowledge is should begin with a description of knowledge itself. Knowledge can be described as a set of models that describe various properties and behaviours within a domain. Knowledge may be recorded in an individual brain or stored in documents, organisational processes, products, facilities and systems.

There exist many possible, equally plausible definitions of knowledge of which one could be "the ideas or understandings which an entity possesses that are used to take effective action to achieve the entity's goals" (University of Texas, Graduate School of Business, 1998). This knowledge is specific to the entity that created it and thus is the familiarity, awareness or understanding gained through experience or study.

In the framework of this project an extreme position on domain knowledge would be: "any knowledge that shapes the data mining process". The entity and its goals in this process would be made up of all involved in the data mining project.

In many applications the concepts of domain knowledge and background knowledge are used interchangeably. In this project however, following the position mentioned above, background knowledge is seen as part of what comprises domain knowledge.

Background knowledge refers to the knowledge about the relevance and meaning of attributes, so to speak the semantics of the application domain. On the basis of background knowledge for instance derived attributes might be constructed because they are considered to be a better data representation.

One also encounters the term "expert knowledge", but in fact this can be seen as an alternative for data mining, consulting an expert could be an option to answer certain questions. However it is certainly true that the results from data mining can eventually extend, confirm or falsify the expert knowledge.

2.2 Necessity to formalise Domain Knowledge

The data mining process can be seen as a problem-solving situation. In this respect it is interesting to discover similarities with the role that expertise and knowledge play in the problem solving process in general as studied for instance by researchers of the Osaka University (Mizoguchi Laboratory, 1995).

The expertise necessary to carry out any problem solving can be seen "as the product of an on-going process in which a structure on knowledge emerges as adaptation to a history of interactions with the problem-solving environment. Knowledge being processed comes from various sources such as domain theory, objects being reasoned about, workplace environment, and so on". "The emerging structure allows for effective application of this knowledge in a problem-solving situation. Expertise is thus tuned to the specific environment in which problem solving is carried out".

Domain knowledge that is useful and necessary to carry out the data mining process is, traditionally speaking, informally available with 'experts' in the organisation. However, without an 'emerging structure' or formalisation, the effective application of data mining is questionable. The same holds for achieving continued or efficient application of data mining since the possibility to reuse knowledge is limited without a structure to incorporate it.

On this reuse the Osaka research group comments: "Because of the specificity of problem solving knowledge, its reuse is limited. To allow for reuse of expertise, a technique of "knowledge decomposition" is widely recognised as being useful. This technique decomposes expertise into several kinds of knowledge, making explicit and justifying the role this knowledge plays in the problem solving process. Understanding knowledge content is a fundamental issue to allow for knowledge reuse and sharing."

Thus the mechanism that is proposed to formalise knowledge is to decompose it. In the setting of the current project this is reflected in the identification of Domain Knowledge Elements. Secondly it is imperative that the elements are justified, explaining their role in the problem solving process. The parallel here is in describing the relation between the knowledge elements and the data mining process decisions.

A preliminary definition of a Domain Knowledge Element would be a well-defined part of Domain Knowledge that shapes a well-defined part of the Mining Process Decisions.

Following the Osaka research group, problem-solving knowledge can be decomposed into task-dependent and domain-dependent portions. The former is called task knowledge and the latter domain knowledge. Furthermore, task knowledge is deeply related to the environment, called workplace, in which the problem solving takes place.

This division seems to be coherent with the flow of three streams of information relevant in the mining process as described in chapter 1 and figure 1.1:

1. domain knowledge can be mapped with the Domain Knowledge stream
2. task knowledge can be mapped with the Mining Experience stream
3. workplace can be mapped with the Database stream involving heuristics and the actual mining

2.3 Domain Knowledge Process Flow

Knowledge acquisition is known as a serious bottleneck in building knowledge-based systems, since it is difficult to elicit expertise from domain experts. Efficient systems for supporting knowledge acquisition are badly needed to overcome this difficulty.

Since not much is elaborated to a great extend in the field of data mining, it is useful to investigate similar fields and analogies. One of these is the field of Requirements Engineering, commonly practised in the domains of software application development. Typically during the formulation of the functional design, input from various parties is acquired from for instance end-users.

Another area with much discussion on knowledge acquisition and representation is that of Knowledge Management. Here it is common to talk about for instance making Knowledge Maps, creating knowledge bases and the problems of capturing non-tacit information.

The diagram below represents the flow of the incorporation of informal domain knowledge into the data mining process. In the current view a form of interview or model editors are used to interrogate an expert on the relevance and content of certain Domain Knowledge Elements in a particular application of the data mining process. Different experts may need to be consulted for the different elements identified. This would be the first translation, or first step of the incorporation, of informal domain knowledge. The second step involves outlining how each element affects each Mining Process Decision. The objective of the Mining Mart project is that this should be standardised as much as possible.

Depicted like this as a process flow it would seem that all starts from the huge body of potentially useful domain knowledge from which information flows in a forward driven manner. However it would be pointless to use the interview to try to extract "all that the expert knows" and distil from that the information to complete the first translation. This would make the incorporation extremely tedious and inefficient.

Figure 2.1: Domain Knowledge incorporation flow

This is however exactly what would happen in the absence of a proper framework for domain knowledge. The key in this scheme of things thus is naming and defining the Domain Knowledge Elements, as these would serve as the guiding principles in the design and completion of a questionnaire or fill-in screens in the KDDSE.

Part of the work in WP5 was to determine the necessary set of Domain Knowledge Elements that covers the concept of Domain Knowledge and that provides all the necessary input for the Mining Process Decisions. This set was derived from a list of Mining Process Decisions considered to reflect the state of the art in the field of data mining. In the end, as reflected in figure 2.2, one could say that the whole process is guided by the mining process decisions that have to be taken.

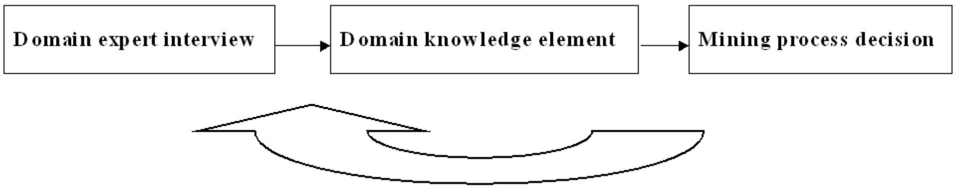

Figure 2.2: Domain Knowledge incorporation driven by Mining Process decisions

The importance thus of formalising the Mining Process Decisions becomes paramount as the identification of Domain Knowledge Elements depends on it as well as the possibility to standardise the preprocessing steps in Mining Mart.

Chapter 3

Data Models and Preprocessing

Before we analyse Domain Knowledge Elements and Mining Process Decisions in detail we have to consider data models first. During the process of designing our project, we will be administrating a collection of data models and step by step filling in the details of these models as we consider more domain knowledge and make decisions. The end product is the data model of our mining table (or tables) that has all the necessary preprocessing operations in it. During the design phase we will thus be considering the collection of data models as a blackboard of notes about our approach.

So what do we mean when we are talking about a data model? By data model we mean the meta-data that forms the schema of a relational database. This includes information about tables, attributes, attribute-types and the relations between tables. We will use the popular UML modelling language or in fact a subset of that which allows us to model just the information mentioned above.

We will be storing a collection of data models and how new ones are derived from existing ones. This will result in a tree of data models with the database data model at the root and polished data models ready for mining at the leafs. This view focuses on the data models rather than the preprocessing necessary to transform a data model into a child in the tree, which is desirable.

By emphasising on the data models we gain a user oriented approach where the user is defining and reasoning about models, and the preprocessing steps are just a logical result of the tree of models.

Figure 3.1 shows the levels of data models that are applied. We distinguish four levels of data models (examples of each data model are given in the following chapter). The levels all use UML to model the data but they differ in their purpose.

Database model This is the model that describes the data as it was found

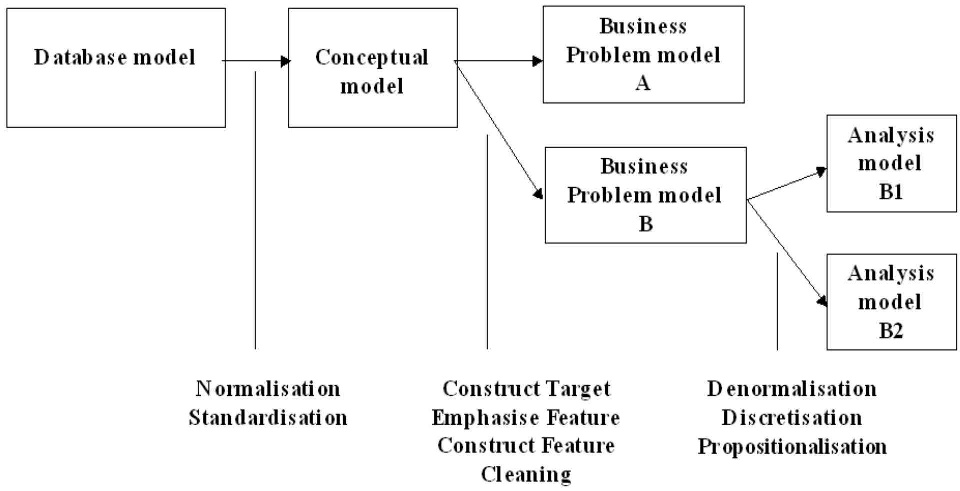

Figure 3.1: Data mining process considered as applying a sequence of data models

in the database. For historic database storage or transaction processing reasons, this model will usually only partially coincide with the entities in the business domain.

Conceptual model This is the data model that describes the entities as they exist in the business domain. It is thus a cleaned up version of the database model. The transformation of the database model to the conceptual model also allows for some standardisation of attributes such as date fields etc.

Business Problem model This data model represents a view on the data that is specific to a particular business problem. Because several problems may be supported by the same data, several business problem models may be derived from a single conceptual model. Business problem models will often contain a subset of tables and attributes based on their relevance to the problem at hand. Domain specific features may be constructed, including new targets.

Analysis model This data model reflects a representation, which is optional for a particular choice of analysis technique. When working with a propositional algorithm, for example one has to propositionalise a multi-relational database (flatten) in order to have a single mining table. For other algorithms it may be necessary to discretise numeric attributes.

It should be noted that at least four data models exist for an application. We do not require the associated data of each model to be actually present in a database. The collection of data models is merely descriptive information available for subsequent reasoning. For example it may be used to come up with a good data storage strategy, which balances computation times of views and spare consumption of tables. This could be hidden from the user. Alternatively the information may be used to base Mining Process Decisions on.

Chapter 4

Production Case Illustration

To illustrate some of the concepts introduced in the report we introduce an imaginary Data Mining application. This example will referenced throughout the report, for instance when illustrating the Domain Knowledge Elements.

The application deals with the industrial production process of steel plates of different shapes and sizes. The plates go through a series of operations and each operation may produce a fault, which requires some handling. The factory is interested in reducing costs associated with these faults.

Of each step data is stored automatically in a database. The database model is shown in figure 4.1. The database does not explicitly store information about plates, so the entity "plate" was added to the conceptual model in figure 4.2.

To emphasise some extra features, (some of) the constructed attributes are shown in figure 4.3 in the business problem model. This includes the new target "step-fault", which is derived from the relation with the "fault" table. Figure 4.3 also shows the final analysis model, in this case prepared for propositional analysis. Stable information about operations is denormalised into the "step" table.

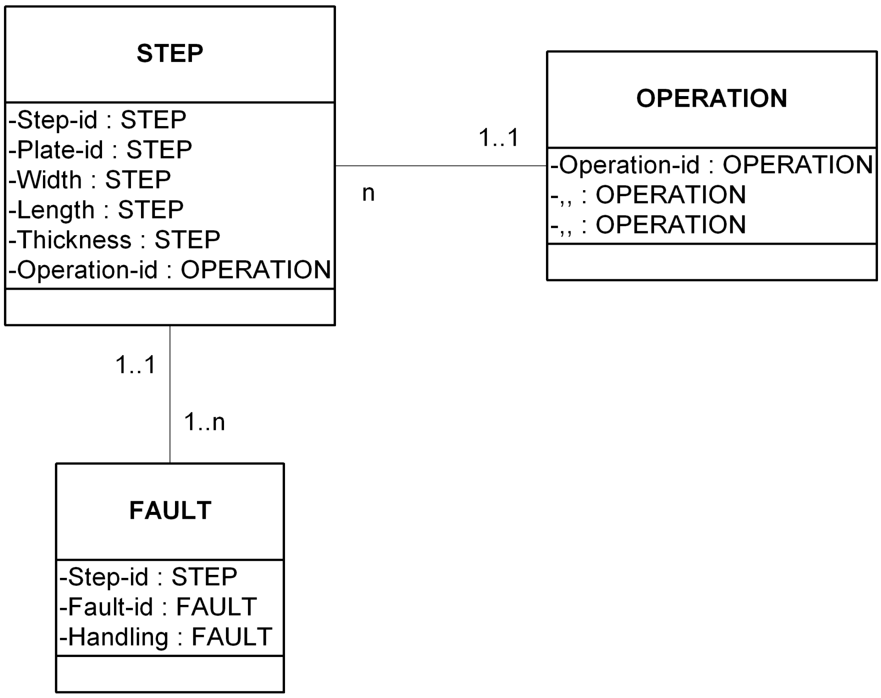

Figure 4.1: Database Model for production case

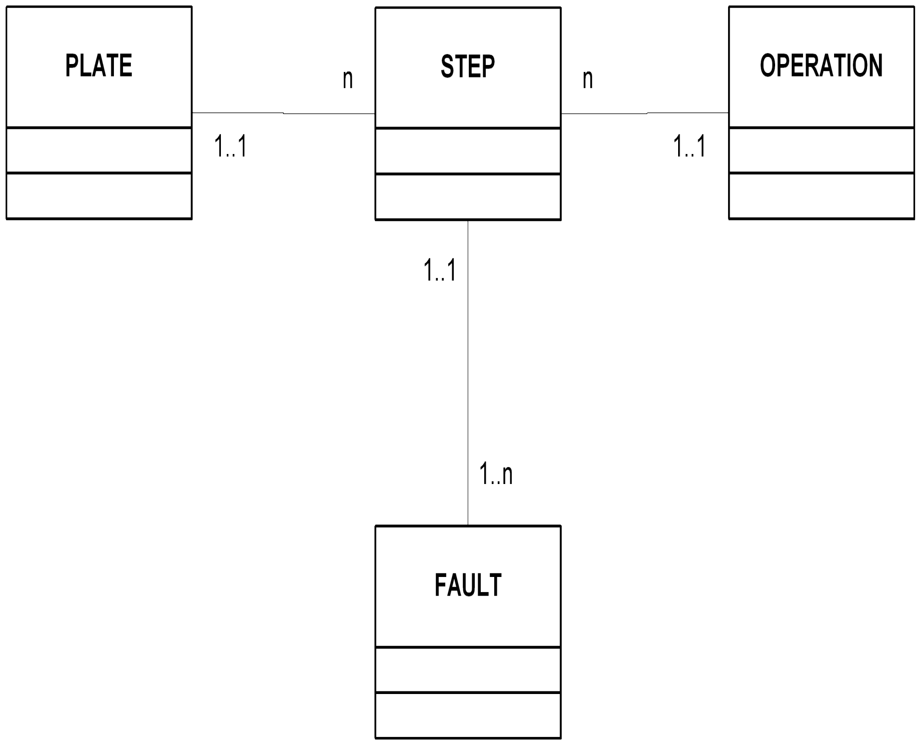

Figure 4.2: Conceptual Model for production case

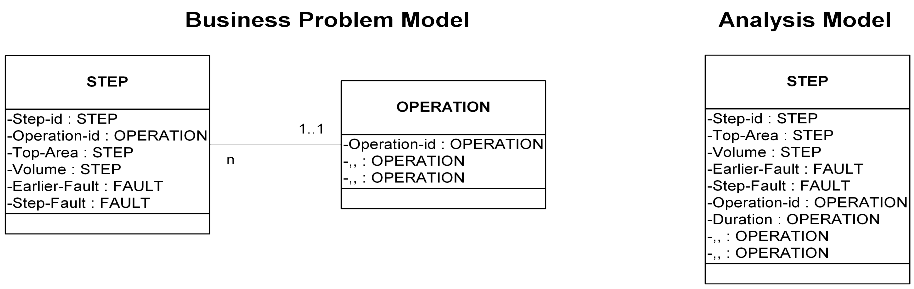

Figure 4.3: Business Problem Model and Analysis Model for production case

Chapter 5

Domain Knowledge Elements

The elements that should provide a complete decomposition of the concept of domain knowledge are found to be the following:

- I Data collection history
- II Data models
  - II.A Database data model
  - II.B Conceptual data model
- III Causal model
- IV Design pattern
- V Analysis paradigm
- VI Goal specification
- VII Background knowledge
- VIII Integrity constraints

In this chapter each will be discussed in turn, providing their definition after which the relevance for the data mining process is explained. Pointers for the acquisition of the knowledge from the expert are given and finally the application of the element in the overall case illustration is indicated.

5.1 Data Collection history

Definition

A definition of this knowledge element would be: Knowledge of the changes of the data representation that could be of influence for the data mining project.

Explanation

Changing conditions or methods of operation in the organisation could lead to changes in what the data is supposed to represent.

For instance, the underlying business process could have changed while the actual data model in place did not (yet). Also it could have been decided to collect (additional) data, that wasn't collected before and therefore it is vital for the data mining process to know the starting date of the addition of attributes and tables.

A third point is that the interpretation of what attributes represent might shift by changing insights or by change of data collection (thus without changing the attribute's label).

Pointers to knowledge acquisition from expert

The following are the acquisition pointers for the domain knowledge expert:

- based on the formulation of the business problem, indicate for the time-span considered if changes in data representation occurred
- based on problem model or analysis model determine if important factor have to be considered, such as change in interpretation of attributes and differing timestamps of the included attributes

Example

Data collection history considerations that might play a role in the materials faults case could be:

- Changes in production process with monitoring and storing of extra parameters
- Slight changes in handling of particular faults while maintaining same handling ID

5.2 Data models

An explanation of data models and their use in the data mining process was given in chapter 3. Examples of these were given in chapter 4. Important to notice is that only the Database data model and the Conceptual data model are, as such, considered Domain Knowledge Elements. The other two, Business Problem model and Analysis model, are derived logically from these models by the mining process decisions such as feature construction and discretisation (see figure 3.1).

5.2.1 Database data model

Pointers to knowledge acquisition from expert

The expert should be asked to deliver the schema of the (relational) database with elements such as tables, attributes, attribute-types and the relations between tables. The latter includes an overview of the foreign key relationships, important to assess referential integrity issues.

5.2.2 Conceptual data model

Pointers to knowledge acquisition from expert

For the conceptual data model the expert should be asked if the entities as they exist in the business domain are well mapped or can be mapped to the database data model.

5.3 Causal Model

Definition

Causal models represent principles or interrelated sets of principles. In general they are presented as a set of nodes and a network of relations drawn between these nodes (graph).

Causal models are understood (Reigeluth, 1999) primarily by:

1 establishing relationships between the real events that constitute a causal model and the generalities (principles or causal models) that represent them, and

2 learning about the network of causal relationships among those events (changes).

In physical settings, a causal model characterises a physical system in terms of state variables and causal influence relations among the variables. In the causal graph defined by the state variables and influence relations, changes in any variable may be propagated to other variables through the influence relations. Implicit in the notion of causality are the concepts of event and causal time. Events comprise change in state variables due to a specific influence relation and with respect to a specific moment in time. Hence, causal time moves forward due to delays in the propagation of changes in the causal model.

Explanation

Producing a causal model can be seen as an important step in reformulating business problems and objectives in an analysable scheme of cause and effect relations between measurable entities. This is necessary in order to identify the objectives of the data mining process and perform any type of data analysis.

For instance selecting indicative and target attributes for a specific analysis implies the application of a causal model, either implicitly of following an explicitly formulated and documented model.

From the model it should be possible to establish relationships with the real events that constitute a causal model, in this case relationships have to be established with tables and variables. Most often the entities or nodes in the model are constructs or features, representing for example behaviour or profiles and the relationship with single variables measuring these features has to be indicated. Since causal models represent cause and effect relationships, formulating a causal model provides the starting point for separating variables in indicative or explanatory classifiers and target or output attributes.

Several alternative divisions of input and output variables are possible if the causal model consist of several related entities in (parallel) chain(s).

Pointers to knowledge acquisition from expert

There are two ways in which domain experts contribute with respect to this knowledge element.

Firstly, presented with the business problem they might be asked to generate a scheme based on his knowledge of the domain, representing the causal model as nodes and relations.

Secondly, once a causal model is generated, the expert or end-user, being presented with the causal model should identify the measurable entities or variables that possibly serve as translation of the causal model. This can be accomplished by presenting both the model and the optional list of tables and variables to choose from.

In summary, the following are the acquisition pointers for the domain knowledge expert:

- list the entities
- determine causality
- map attributes and tables on causal model
- try different divisions of input and output variables

Example

Applying the materials process fault illustration to this domain element, the resulting causal model for the problem could be depicted as given below.

Entities and their relations as well as two alternatives for the division in input and output variables are suggested.

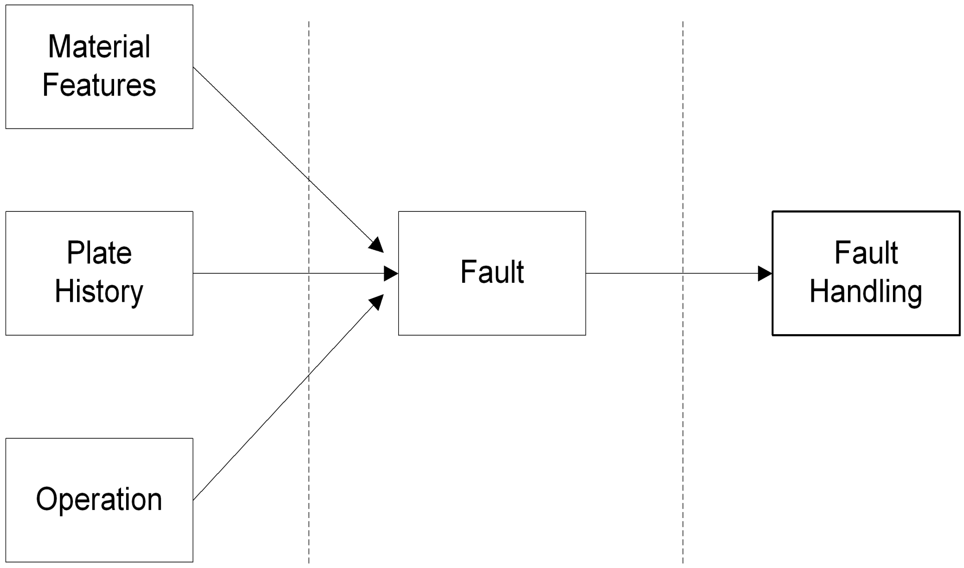

Figure 5.1: Mapping of the attributes is done by relating entities from the causal model to the corresponding tables and attributes from the (conceptual) data model. In this case the example of mapping of the "Material Features" entity is given.

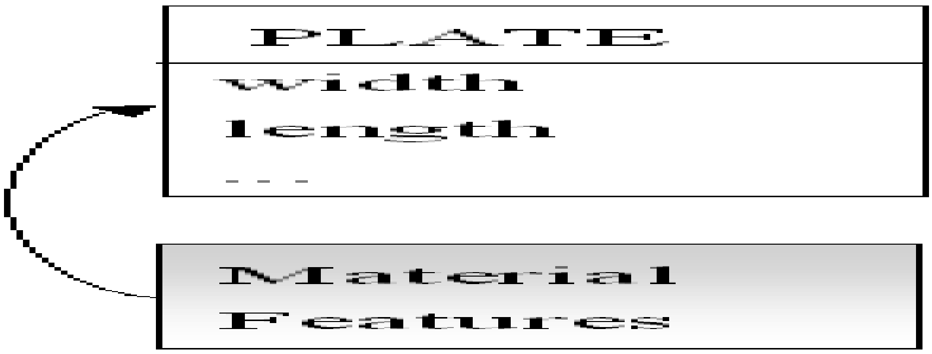

Figure 5.2:

5.4 Design Patterns

Definition

Design patterns have their origin in the field of architectural studies, mainly by the work of Alexander, discussed in (Shalloway and Trott, 2000) who postulated that patterns exist which solve virtually every architectural problem that one will encounter. He defined a pattern as “a solution to a problem in a context”. *“Each pattern describes a problem which occurs over and over again in our environment and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice.”*

Looking for the commonalties, especially commonality in the features of the problem to be solved, design patterns are often thought of as useful solutions to recurring problems. Relating to object-oriented design principles, another domain where design patterns are applied is that of developing and designing application architectures.

Apart from data abstractions, design patterns often are more than a kind of template to solve ones problems. “They are a way of describing motivations by including both what we want to have happen along with the problems that are plaguing us.”

Explanation

Also in the data mining process, design patterns can be useful to identify recurring situations or courses of actions. These courses of actions are in many cases referred to as 'experimental set-ups' or 'test designs'. For instance in a test design for cross-validation, one will always encounter the use and division of data sets for training and validation. Other designs also can be characterised by their particular steps and treatment of data. In a particular data mining project, design patterns may be applied in sequence.

Establishing a design pattern can be seen as creating a generic framework to look at a process abstractly, making it independent of particular tasks and analysis steps.

The design patterns distinguished and modelled so far in this project are listed below.

- Cross-validation
- Extrapolation
- Evolution
- Modelling
- Boosting

- Feedback
- Forecasting

In the following only for Cross-validation a short description is given. The complete elaboration on all design patterns will be the subject of WP19. For instance the components required in a pattern description such as the purpose of the pattern and how the pattern provides a solution to the problem that it solves, will be defined in WP19.

Cross-validation

Cross-validation is defined (Fayyad et al., 1996) as "Mechanism that uses a given sample set to generate hypotheses and estimate their validity and accuracy in the population. The sample set is repeatedly and randomly divided into disjoint training and test data sets".

What is described here as a series of actions and steps, can be modelled as a generic design to be applied in differing situations. The design pattern for Cross-validation is then as shown in figure 5.3.

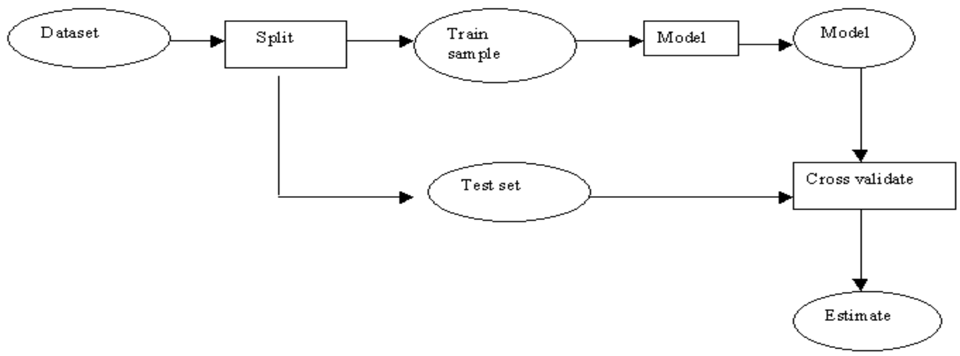

Figure 5.3: Design pattern for Cross-validation

Extrapolation

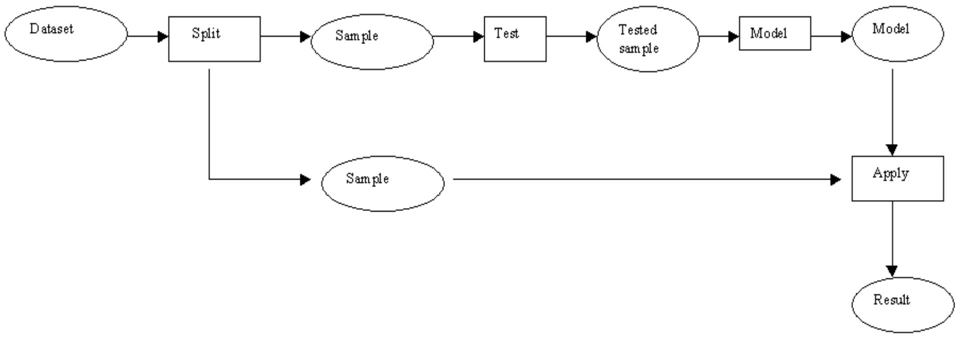

Figure 5.4: Design pattern for Extrapolation

Evolution

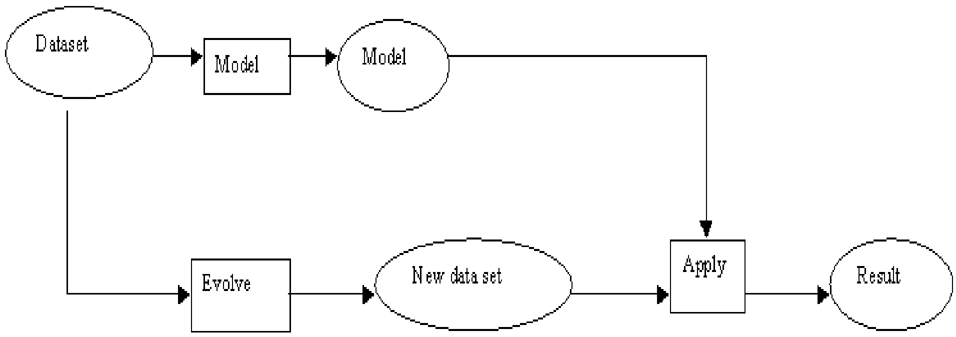

Figure 5.5: Design pattern for Evolution

Modelling

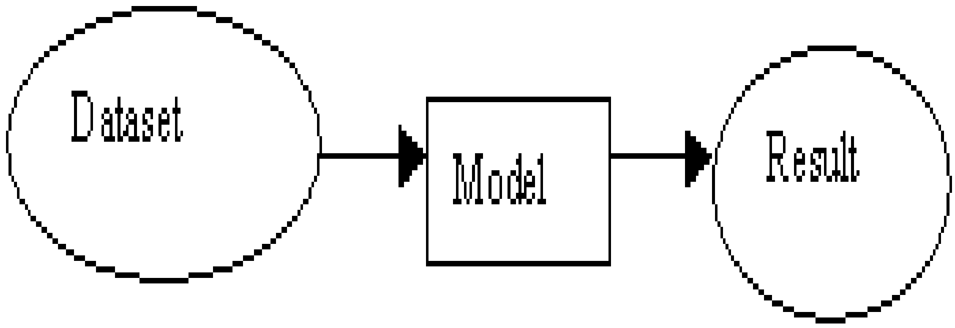

Figure 5.6: Design pattern for Modelling

Boosting

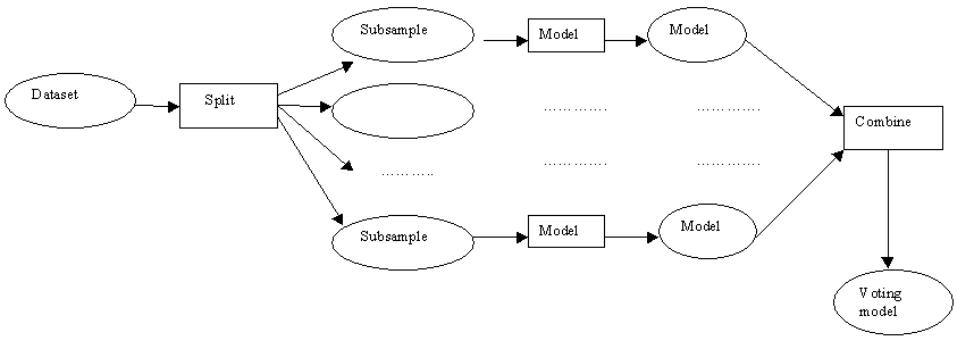

Figure 5.7: Design pattern for Boosting

Feedback

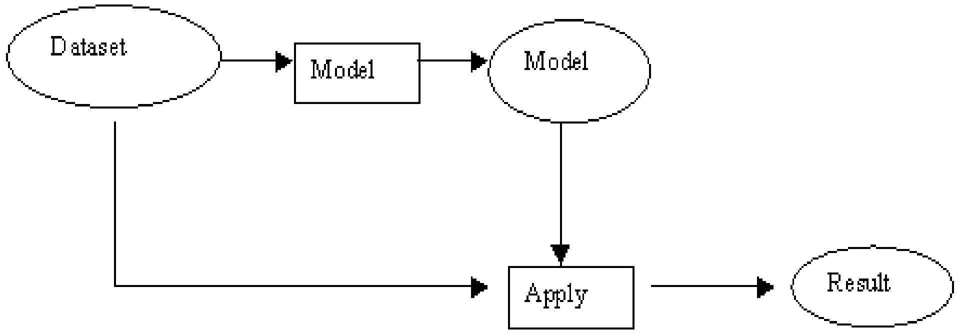

Figure 5.8: Design pattern for Feedback

Forecasting

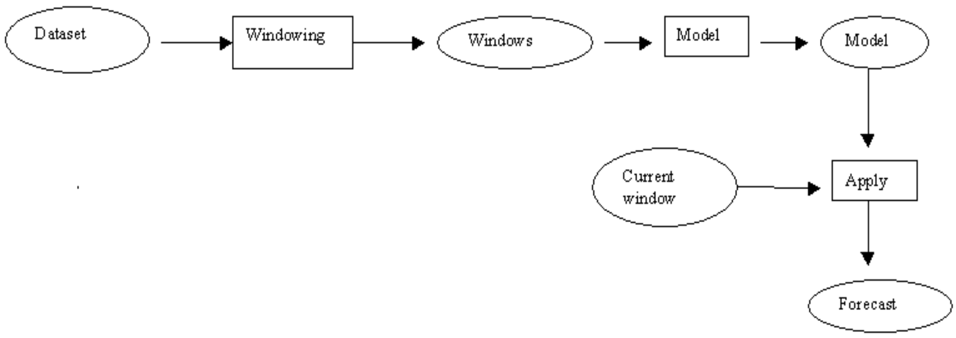

Figure 5.9: Design pattern for Forecasting

Pointers to knowledge acquisition from expert

For this domain knowledge element it is important that the expert identifies which test design is actually followed in the data mining process. As stated before this might be a sequence of designs, but in most cases it is not, as the unit of analysis, thus what is it that actually comprises a single data mining process, normally would be covered be a single design.

In summary, the following are the acquisition pointers for the domain knowledge expert:

- looking at definitions, choose the pattern that complies
- the exact procedure of how to choose has to be established still

Example

The design patterns that are applicable to the materials faults case are:

- Modelling ⇒ Understanding of conditions generating faults
- Cross-validation ⇒ Quality measure of models applied to different series of plates
- (Extrapolation ⇒ Prediction of fault rates)

5.5 Analysis Paradigm

Definition

A definition of Analysis Paradigm is that it is a high level description of the data mining task at hand, depending on the goal of the task.

In general one can state on paradigms that paradigmatic related are those entities that belong to the same set by virtue of a function they share. In natural language studies for instance there are grammatical paradigms such as verbs or nouns (Chandler, 1994).

Explanation

In the methodology for data mining described in the Cross-Industry Standard Process Model for Data Mining CRISP-DM (Chapman et al., 1999) a distinction is made between data mining problem types, each describing a specific class of objectives which the data-mining project deals with. A data-mining project normally involves a sequence of different problem types working to the final solution of the problem. Establishing an analysis as a case of 'segmentation', 'classification', 'prediction' or 'dependency analysis' influences for instance the choice from the available modelling techniques as some are more appropriate than others to be used with each. Rather than choosing 'neural networks' or 'decision trees' one should choose the reigning analysis paradigm.

Establishing a paradigm can be seen as creating a generic framework to look at a process abstractly, making it independent of particular tasks and analysis steps. Or in the context of data mining: one has to prevent that one "Can't See the Forest Because of the Decision Trees" (Chelst, 1997).

This sounds very much the same as the description of design patterns, but here the data mining process is viewed alongside another axis. When the modelling step is reached, in most design patterns one has the alternatives of a variety of modelling techniques. Or stated otherwise and reversed: segmentation can be executed using cross-validation or feed back.

In analogy again to natural language studies, design patterns then could be seen as sentences (a syntagm of words). "A syntagm is an orderly combination of interacting signifiers which forms a meaningful whole (sometimes called a 'chain'). Such combinations are made within a framework of syntactic rules and conventions (both explicit and implicit). In language, a sentence, for instance, is a syntagm of words" (Chandler, 1994).

Various concurrent distinctions of primary data mining tasks exist in the literature, all resembling each other to some extend. For this project the following analysis paradigms (or primary data mining tasks) are identified as given in the list below:

- Classification
- Regression
- Rule induction
- Association
- Clustering
- Visualisation

Pointers to knowledge acquisition from expert

For the knowledge element of analysis paradigm it is important that the expert identifies which problem type actually governs a particular step of the data mining process. As stated before this might be a sequence of steps, for instance the output of a rule induction model may serve as input to solve a classification problem.

In summary, the following are the acquisition pointers for the domain knowledge expert:

- looking at definitions, choose the paradigm that complies
- the exact procedure of how to choose has to be established still

Example

The analysis paradigms that are applicable to the materials faults case are:

- Rule induction/interesting subgroup discovery ⇒ Description of combinations of features responsible for generating faults
- Classification ⇒ Applied to cross-validate induced rule sets on different series of plates

5.6 Goal Specification

Definition

Goal Specification can be defined as "a statement of the ultimate reason why the data mining project is undertaken and when it is successful".

Explanation

The goal specification process is important, because without thorough understanding of the objectives one might end up giving unneeded solutions. Similar to the reformulation of the data to achieve an optimal data representation and therefore quality of data mining results, the reformulation of the problem is a core technique in problem solving. The formulation of discovery tasks in terms of business applications is a difficult one and bridges the gap between the users and the technology.

The reformulation of the problem is preceded by clearly stating the problem in business terms. This can be seen as the first level of the goal specification process in which it should be determined from a business perspective what one really wants to analyze and see answered. In the CRISP-DM methodology this is also described as determining the business success criteria: "Describe the criteria for a successful or useful outcome of the project from the business point of view."

Examples given of these specifications are either quite specific: reduction of customer churn to a certain level, or general and subjective such as "give useful insights into the relationships". In the latter case it should be indicated who will make the subjective judgement.

The second level of goal specification is the translation of the higher-level business goals into a data mining problem definition and associated goals, thereby reformulating the problem. Again the CRISP-DM methodology also refers to these as success criteria, which describe the intended outputs in technical terms. The assumption is that complying with the technical criteria will bring about the fulfilment of the business goals.

For example, the business goal might be "Increase catalogue sales to existing customers" while a data mining goal might be "Predict how many widgets a customer will buy, given their purchases over the past three years, demographic information (age, salary, city, etc.), and the price of the item".

For example a certain level of predictive accuracy, or a propensity to purchase profile with a given degree of "lift". As with business success criteria, it may be necessary to describe these in subjective terms, in which case the person or persons making the subjective judgement should be identified.

Pointers to knowledge acquisition from expert

From the domain knowledge expert the following is expected:

- define the 2 goal levels as items
- show diagram relating the levels
- specify criteria for model assessment (accuracy, performance and complexity)
- clarify the role of the measurement, define benchmarks for evaluation criteria

Example

The 2-level goal specification for the materials faults case is:

- High level: reduce number of faults
- Low level: understand reasons/circumstances for fault

5.7 Background Knowledge

Definition

Background Knowledge can be defined as empirically verified or proven information specific to the application domain that serves to restrict the problem or search space.

This definition shows that what is considered background knowledge in this project is in fact limited compared to other publications. For instance Fayyad et al. (Fayyad et al., 1996) stress that background knowledge of the domain expert is crucial in the process of model development. The examples given however include instances of what in this project is seen as the 'causal model' knowledge element.

Typical forms that fall within the definition are sets of contingent formulae in first order logic, systems of equations and taxonomies.

Explanation

A typical example of the use of background knowledge is that for deriving attributes. Based on background knowledge it is judged that some fact is important and ought to be represented although there is no attribute to represent it directly. A new attribute might be constructed combining others in a certain formula. This is also what happens when an attribute ought to be corrected, for instance applying a (non-linear) compensation of air pressure for height.

Although incorporating background knowledge in the algorithm is strictly speaking not a necessity for the learning process, it will considerably speed up this process. Furthermore some analysis tools can actually take advantage of explicitly represented background knowledge. This is for example seen in the incorporation of first order logic rules in inductive logic programming.

Pointers to knowledge acquisition from expert

In summary, the following are the acquisition pointers for the domain knowledge expert:

- looking at attributes the expert should indicate if some are to be combined of re-scaled etc.
- the expert should indicate, based on data dictionary knowledge, if the data contains sequential aspects and how this is represented.

Example

Examples of the incorporation of background knowledge in the materials faults case are:

- Plates are rectangular, so dimensions may be multiplied etc.
- Step-id's are chronological, so preceding steps may be identified

5.8 Integrity Constraints

Definition

Adapting definitions common to the domain of database administration, Integrity Constraints can be defined as a mechanism to prevent invalid data, not in compliance with business rules, to enter into the data model and the data mining process.

Explanation

The importance of preventing invalid data to enter the data mining process may be obvious as a means to prevent drawing the wrong conclusions or even that algorithms get stuck.

Integrity constraints might be divided in constraints for values of single attributes and constraints for combinations of attributes.

Examples from the first class are:

- not NULL constraint, when values for certain attributes may not be missing for cases

- UNIQUE constraint, when attributes in the data may not contain duplicate values for cases
- Domain constraint, when the data values for some attributes may only come from a certain value range (e.g. 0-100, male/female)

Examples from constraints on combinations of (values of) attributes are:

- Combination of values, when a value (range) of one attribute should coincide with some values on another attribute
- Clausal integrity constraints, a variant from the previous constraint refers to constraints on relationships between attributes.
  - For instance (De Raedt and Bruynooghe, 1993) a database could contain facts about family relations: mother(X,Y), father(X,Y), and parent(X,Y). Integrity constraints could state that it is impossible that X is at the same time both the mother and the father of Y and that it is impossible for a person to be his own parent, thus parent(X, X).

Referential integrity, in fact constraints on combinations of values of attributes between different tables, is not mentioned here as it is considered part of the Data Models domain knowledge element.

Pointers to knowledge acquisition from expert

The acquisition pointer for the domain knowledge expert in this case would be:

- looking at attributes (proposed to include in the analysis) the expert should indicate if some bare or should bare constraints on them that need to be checked

Example

For the materials faults case some of the integrity constraints that would play a role are:

- not NULL constraints for step-id and operation-id attributes
- If "step-id's are chronological, so preceding steps may be identified" as noted on the basis of background knowledge, the UNIQUE constraint should apply

Chapter 6

Mining Process Decisions

The elements that should provide a complete coverage of the data mining process decisions are found to be the following as presented in the list below. In continuation each will be discussed in this chapter.

1 Standardisation
   - 1.1 Normalisation
   - 1.2 Standardisation

2 Target selection

3 Result
   - 3.1 Quality measure
     - 3.1.1 Quality function
     - 3.1.2 Cost matrix
   - 3.2 Baseline
     - 3.2.1 Expectations
     - 3.2.2 ROI
     - 3.2.3 Naïve approach
     - 3.2.4 Previous attempts
   - 3.3 Interestingness

4 Analysis technique

5 Pollution/cleaning
   - 5.1 Pollution estimate
   - 5.2 Filtering
   - 5.3 Cleaning

6 Record selection

- 6.1 Scope
- 6.2 Validity (historic)

7 Table/attribute selection

- 7.1 Table
  - 7.1.1 Scope
- 7.1 Attribute
  - 7.1.1 Scope
  - 7.1.2 Validity
  - 7.1.3 Causality

8 Feature construction

- 8.1 Background knowledge
- 8.2 Algorithmic limitations
  - 8.2.1 Propositional
  - 8.2.2 Multi-relational

9 Temporal aspects

10 Presentation

- 10.1 Expertise level
- 10.2 Use of results

11 Interpretation

- 11.1 Table annotation
- 11.2 Attribute annotation
- 11.3 Value annotation

12 Scope

- 12.1 Computation time
- 12.2 Search depth

13 Deployment

- 13.1 Platform
- 13.2 Embedding

6.1 Standardisation

6.1.1 Normalisation

Different tables of a large database schema are in different normal forms, e.g. in third or fifth normal form. Having the tables partially denormalised has the advantage the all attributes describing the same entities are concentrated in one table. Obviously this process will lead to universal relation. Therefore, one has to take the decision, when this denormalisation step has to stop. What is needed is the knowledge about the different kinds of entities, which should be kept apart in different tables.

6.1.2 Standardisation

All the values of different attributes, which belong to the same type, must agree with certain standards. For date values, this can be the European format: day, month, and year. All symbolic values must use the same character encoding, e.g. ASCII or the German character set. And attributes of type number must have the same precision, if they all represent e.g. monetary values in the same unit. Moreover, it is almost always assumed that the values for one concept only represent one kind of information. If e.g. the values from 1 to 100 represent the age of male persons and the values between 101 and 200 the age of female persons, then this attribute will be split up into two, one for the sex and one for the age.

6.2 Target Selection

One has to specify the target objects of the data mining process. For a mailing action of an insurance company, this can be certain private households, which have be to characterised according to their responding behaviour to mailing actions. Since the target objects may not exist as such in the database, this decision may result in complex process of constructing the target objects out of the entities in the database. If the databases stores only business partners as objects, private household may be characterised as groupings of those partners, which live together, i.e. have the same address, and which play specific roles in their respective contracts.

6.3 Result

6.3.1 Quality measure

6.3.1.1 Quality function

One has to define a function to evaluate the results of the data mining process. Although predictive accuracy is most often used as evaluation function for the model building step, the quality function for the data mining process may be different. It may try to weigh the costs and benefits of the results by using cost functions.

6.3.1.2 Cost matrix

In many data mining projects it is impossible to learn a 100% model. However, the costs for making an error are not necessarily equal concerning the under- or overestimation of values in regression problems or the different kinds of errors in classification problems. In the case of two-class classification problems, there exist four different possibilities to classify an example. True positive examples are those, which are positive and classified as positive. True negatives are defined accordingly. False positives are negative examples, which were wrongly classified as positive, and false negative examples are positive examples which were wrong classified as negative examples. A cost matrix can be defined which associates a different cost value with the different kinds of outcomes.

6.3.2 Baseline

6.3.2.1 Expectations

To evaluate the success of a data mining project one has to define a baseline, to which the results can be compared. The baseline consists of the high level goals of the project and the data mining success criteria.

6.3.2.2 Return on investment (ROI)

First, one has to determine all the costs, associated with the data mining project. For this investment, a certain percentage of return has to be calculated, which defines the break-even point for the financial success of the project.

6.3.2.3 Naïve approach

To evaluate the success of the data mining process, its results can be compared with a naïve approach. In the case of a mailing action, the ROI for a data mining project resulting in mailings to certain customers should be higher simply mailing to everyone without any data mining at all.

6.3.2.4 Previous attempts

One success criterion for the data mining project can be the comparisons with previous attempts. If e.g. a mailing action is carried out several times a year, it can be a goal that the rate of respondents should be higher than the last time.

6.3.3 Interestingness

The results of the data mining process have to be interesting. Usually, interestingness is defined as a general measure for the value of the discovered pattern. It combines the validity, novelty, usefulness, and understandability. A discovered pattern should be valid on new data. This can be approximated be requiring a certain degree of predictive accuracy for the learned model. If discovered patterns are distinct from the expected patterns, they can be regarded as novel. If the results of the mining process lead to e.g. an increase in profits of a company, they can be considered as useful patterns. Understandability is often substituted by simplicity, assuming that less complex patterns are more understandable. This ranges from syntactic measures, e.g. the size in bits of patterns, to semantic measures, e.g. easy for humans to comprehend in some setting.

6.4 Analysis Technique

In Domain Knowledge Element: DKE V we listed different analysis paradigms like segmentation, classification, regression, clustering and dependency analysis. In the first place, one has to decide to which problem type the data mining process or a step of it belongs. This delivers a range of applicable analysis techniques. Among them, e.g. algorithms suitable for classification problems, one has to select at least one technique for model building.

6.5 Pollution / Cleaning

6.5.1 Pollution estimate

Decide which measure to use to estimate the degree the data contains noise or missing values. This depends on the process, producing the data, and the kind of data collection. If data is automatically acquired, how accurate are the measurements. Manual data input is also a source of noise.

6.5.2 Filtering

The Mining Process Decision to take here is which filtering strategy to use if the data is noisy and one has to filter out data records. The decision could be that one excludes data of the analysis, because the noise in that record is to high, e.g. more than three missing values.

6.5.3 Cleaning

Noisy, numeric attributes can be cleaned by deciding to substitute all values below and above some bounds by the mean plus or minus x times the standard deviation. For symbolic attributes, it may be useful to group very seldom values together and replace them by a new value. For missing values one can predict the most probable value or use the mean value and insert it into the record.

6.6 Record Selection

6.6.1 Scope

One has to decide which records have to be selected for the analysis task. Records at this point are not necessarily tuples in one database table, but rather objects on the conceptual level. If the database contains records about private households and companies as business partners in the same tables, the decision can be to select all data about private households with children.

6.6.2 Validity (historic)

In many cases, the data is time stamped, marking e.g. the time point when the data was inserted into the database. In order to prevent the analysis of outdated data to analyse only valid data, a decision is made to analyse e.g. the data of the last year.

6.7 Table / Attribute Selection

Find the relevant attributes for the data mining task.

6.7.1 Table

6.7.1.1 Scope

A subset of all database tables has to be determined, which is of interest for the analysis task. Among these tables, all further data selection steps take place.

6.7.2 Attribute

6.7.2.1 Scope

For each table, select the attributes, which contain useful information for the analysis task.

6.7.2.2 Validity

One has to decide, if the attribute is valid for the analysis. Due to privacy reasons e.g., some attributes are not allowed to be examined, although they may contain relevant data.

6.7.2.3 Causality

All attributes, which are a part of the causal model, (confer DKE III), will be selected for analysis and assigned to the set of either input or output attributes.

6.8 Feature Construction

The attributes used in the tables are not necessarily the best features for every mining task.

6.8.1 Background knowledge

Feature construction by background knowledge can be accomplished by using mathematical formulas to compute the values for a new attribute. Or, an external algorithm is applied on certain attributes to compute e.g. level changes of blood pressures in a medical application. As another example, if one has data about children and their parents, it is possible to introduce attributes describing all the other family relationships like grandmother or relative.

6.8.2 Algorithmic limitations

The decision necessary to take because of algorithmic limitations refers to the limitations on the input that algorithms can work with. This might make it necessary to create new features or transform existing ones.

6.8.2.1 Propositional

In propositional analysis, some algorithms can only take input of the nominal level and thus numeric variables have to be transformed to contain discrete values. The opposite case can also occur.

6.8.2.2 Multi-relational

In multi–relational analysis, more limitations may be of importance above the ones for the propositional case. In the current situation, algorithms like for decision trees and neural networks can only operate on a single table, making it necessary to execute transformations on the data e.g. propositionalise a multi-relational database (flatten).

6.9 Temporal Aspects

First, one has to decide if temporal aspects play any role at all for the mining task. If the mining goal is a forecasting task, then it is obvious, that one has to treat the data as a time series. But that does not prescribe automatically the use of time series analysis techniques. However, even if the mining goal is one of the other tasks, e.g. classification or regression, temporal aspect may still be important. This depends to a large extent on the process, where the data to be analysed comes from. One example are technical processes which may change over time and where theses changes have to be taken into account. This leads to the selection of data, only after the last process change. But even if the process or the business remains unchanged, e.g. offering the same products all the time, there can be external factors, e.g. seasonal effects, which influence the buying behaviour. Moreover, the interests of the customers may have changed resulting in "concept drift". Here, one can distinguish between processes with periodic changes, like e.g. a sine wave, but the process remains the same. Or, in the case of concept drift, there are breaks at some point in time, which can result in abrupt changes in the data, collected from the process. In either case, time effects must be handled either in the data representation or by the algorithm. If this direct handling is not possible, then one can use a more indirect approach by moving a window over the time ordered data.

A more in-depth study of temporal aspects in the mining process is subject of WP3.

6.10 Presentation

For the presentation of the mining results one has to distinguish between the audience, to whom the results will be presented according to their respective level of expertise, and how the results itself will be used.

6.10.1 Expertise level

For a domain expert, specific learned rules may be of particular interest, which reveal new details about certain customers. The business manager may be only interested in the monetary consequences of the learned models.

6.10.2 Use of results

Two possible ways to use the learned model are to regard the mining results as a piece of knowledge or as a software module. In the former case, the learned model, e.g. a decision tree, may be presented as a whole as the learning result. Or only parts of the tree are shown to the domain expert, maybe only after the conversion of the tree into logical rules. Or only the performance measures of the model like accuracy and coverage are presented. This depends to a large extent on the audience and their level of expertise. In other cases, the learning results will be treated as a piece of software, e.g.

a trained neural net. Then, it will not be presented at all, but integrated or embedded into other software systems.

6.11 Interpretation

6.11.1 Table annotation

For each table a description has to be given. Here it is interesting to know if a table builds an intermediate concept. In a database about cars, it is possible to model the relationships which car has which engine and gear in one table or two. In the former case, the table has two attributes, identification number of the car and identification number of the part, where engines and gears have different and distinct numbers. In the latter case, the second attribute is identification number of engine and gear respectively, thus building intermediate concepts.

6.11.2 Attribute annotation

Every attribute needs an annotation describing its relationship to the real world. One has to explain what aspect of an entity is modelled by which attribute. Since some attributes are a result of feature construction mechanisms, the denormalisation, or other data selection steps, their meaning will not be obvious. They have to be described. In addition, for all attributes, their position in the different data models like the conceptual model or the causal model will be indicated.

6.11.3 Value annotation

For all attributes, the meaning of their values will be given. In some cases like numbers or descriptive symbolic names, it may be obvious. But numbers can be used for codes, where e.g. the exploitation of the normal ordering is inappropriate. In addition, one has to indicate the range of the values or a list of all possible values, since many mining techniques rely on this information. If null values are allowed, the special symbol for them must be given.

6.12 Scope

There are different reasons, which can make it necessary to restrict the computation time and search depth of the learning algorithm. These restrictions may be caused by time and resource constraints of the project itself. But here, we do not discuss that issue. Instead, we assume that all general resources are sufficient to choose any analysis technique and that these restrictions are caused by other, more specific reasons.

6.12.1 Computation time

Most algorithms do not allow to control their computation time directly via a parameter. Since the computation time depends on the amount of data to be processed, the search depth (or size of the hypothesis space), the model selection techniques to be used, or further parameters of the algorithm, these factors are taken into account to control the computation time indirectly. For all learning algorithms the learning time increases with the amount of data. In addition, the accuracy increases, if a learning algorithm use the whole data set instead of a sample. But, depending on the application, a lower degree of accuracy may be sufficient. By requiring a certain amount of accuracy, most often controlled by the acceptance criterion of the learner, the learning time will be controlled. The choice of the model selection technique like cross–validation, leave–one–out testing, or a simple train–test split has a direct impact on the learning time too. The choice itself is influenced by the analysis technique. Learning algorithms with a high variance may make it necessary to use variance reduction techniques later on. Parameters of the learning algorithm with an influence on the learning time are e.g. the stop criterion of neural nets to prevent overfitting.

6.12.2 Search depth

Search depth restrictions are used because of background knowledge, which can be exploited to exclude parts of the search or hypothesis space. This knowledge may have the form that one knows that there are irrelevant parts in the search space, e.g. certain combinations of attributes. Or the form of the resulting hypothesis is not relevant, e.g. one excludes the learning of recursive rules. Nevertheless, search depth restrictions are also used to simply control the learning time indirectly.

6.13 Deployment

For the deployment of the data mining results, the prospective computing platform and the embedding into the computing infrastructure of the business may be important.

6.13.1 Platform

The execution and deployment of learned models in a specific hard- and software environment may exclude or prescribe the use of specific algorithms or analysis techniques. This decision is only based on technical reasons, because e.g. specific software is not available on UNIX systems.

6.13.2 Embedding

Often, it is necessary to integrate the learning results into application systems like e.g. a mailing system or production planning systems (PPS). If it is necessary to convert the learned model into decision lists for the PPS system, this task will be rather straightforward in the case of learned decision trees, whereas it may be unknown, how to convert a trained neural net into this required structure. Therefore, decisions about the embedding of learning results are connected with the selection of the analysis technique too.

Chapter 7

From DKE's to MPD's

It was observed before that the purpose of DKE's is to provide input to the Mining Process Decisions. Because not each DKE contributes to a decision and not all decisions are supported by a DKE. We have organised the contribution of DKE's to MPD's in the following matrix in table 7.1. An X indicates a contribution, the thought behind it being that it should constitute a reasonable effect. Most X's are quite straightforward. For instance, domain knowledge on Integrity Constraints (DKE VIII) provides input to the decisions on Pollution/Cleaning (MPD 5) to decide what the pollution estimate should be and how to clean values.

For some X's the precise relation requires more work, which will appear in the next version.

Table 7.1: The Domain Knowledge Translation Matrix

	I	II.A	II.B	III	IV	V	VI	VII	VIII
1	x	x	x					x	
2			x	x				x	
3			x	x	x		x	x	
4				x	x	x			
5	x		x				x	x	x
6	x		x		x	x	x		
7	x		x	x		x		x	x
8			x			x	x	x	
9	x		x	x	x	x		x	
10		x					x	x	
11	x	x						x	
12			x		x		x		
13	x	x			x	x	x		

Bibliography

Chandler, D. (1994). Semiotics for Beginners. WWW document visited June 2000, URL: http://www.aber.ac.uk/media/Documents/S4B/semiotic.html.

Chapman, P., Clinton, J., Khabaza, T., Reinartz, T., and Wirth, R. (1999). The CRISP–DM Process Model. Technical report, The CRIP–DM Consortium NCR Systems Engineering Copenhagen, DaimlerChrysler AG, Integral Solutions Ltd., and OHRA Verzekeringen en Bank Groep B.V. This Project (24959) is partially funded by the European Commission under the ESPRIT Program.

Chelst, K. (1997). Can't See the Forest Because of the Decision Trees, ACritique of Decision Analysis in OR/MS Survey Textbooks. *DecisionAnalysis Society Newsletter*, 16(1).

De Raedt, L. and Bruynooghe, M. (1993). A Theory of Clausal Discovery. In Bajcy, R., editor, *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, volume 2 of *IJCAI 93*, pages 1058–1063, San Mateo, CA. Morgan Kaufmann.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., editors (1996). *Advances in Knowledge Discovery and Data Mining*. AAAI Press Series in Computer Science. A Bradford Book, The MIT Press, Cambridge Massachusetts, London England.

Mizoguchi Laboratory (1995). WWW document visited June 2000, URL: http://www.ei.sanken.osaka-u.ac.jp/projects/reuse-ontologies.html.

Reigeluth, C. M. (1999). Causal understanding, Principles for Learning-Meaningful Knowledge. In *Basic Methods of Instruction*. Indiana University.

Shalloway, A. and Trott, J. R. (2000). Using Design Patterns From Analysisto Implementation. Net Objectives Inc. Draft: 2000, URL: http://www.netobjectives.com/download/what_are_design_patterns.pdf.

University of Texas, Graduate School of Business (1998). Answers toFrequently Asked Questions About Knowledge Management. WWW document visited June 2000, URL: http://www.bus.utexas.edu/kman/ answers.htm.