Enabling End-User Datawarehouse Mining
Contract No. IST-1999-11993
Deliverable No. D10

Case Base of Preprocessing
Deliverable D10

Olaf Rem

Perot Systems Nederland B.V.
P.O. box 2729, NL-3800 GG Amersfoort, The Netherlands
{Olaf.Rem}@ps.net

December 20, 2002

Abstract

The case base of preprocessing is an important part in the objective of creating a "Mining Mart", that is, a central place (on the Internet) where data mining experience is shared and re-used. The objectives for this workpackage are to design, implement and fill the case base. It should support browsing, searching, re-using and addition of applictions. At the heart of the case base is an object model that stores meta data about cases and provides business level information about the application. The case base has been implemented using InfoLayer (developed earlier by the University of Dortmund). In this report the object model, the presentation using InfoLayer, and using the case base are described.

Contents

List of Figures

Chapter 1

Introduction

The objective of workpackage 10 in the Mining Mart project is to develop an environment for storing cases and to fill this environment with a number of cases. The case base should become connected to the Internet. Connecting the case base to the Internet is the subject of workpackage 9. With the case base it should become possible to create a "marketplace" where cases are shared between interested parties. It should be possible to browse cases, search for specific types of cases, re-use cases of others, and add new cases. Re-using cases and learning from other experienced users is expected to help "speed-up the discovery process" and to "improve the quality of mining". These are two important success criteria for the Mining Mart project.

Non expert users should be supported in finding interesting cases by adding high level features to the case descriptions. This means adding a business level description to cases. An important part of the work in this work package focused on developing a model to capture this business level.

The position of workpackage 10 is shown in Figure 1.1. Workpackage 19 about Problem Modeling gives input for the business level description. Workpackage 8 delivered the Mining Mart Meta Model (M4), which is at the hart of the Mining Mart system (workpackage 12). The M4 also plays an important role in the case base as we will show further on in the report. With the Mining Mart system users can create new preprocessing cases or edit existing cases.[1] Workpackage 6 delivered business data that has been used as an example case for the case base. Workpackage 9 connects the case base to the Internet and workpackage 17 is involved in evaluating the case base.

There are various other sources available that provide more information

[1] Both in workpackage 10 and 12 the term "case" is used. There is a difference in meaning, however. In workpackage 12 the term means a set of meta data describing preprocessing steps and involved data; in workpackage 10 it additionally includes a business level description. In order to limit confusion we will use the term "application" when we mean a case that includes a business level description and "case" when we refer to a case from the Mining Mart system.

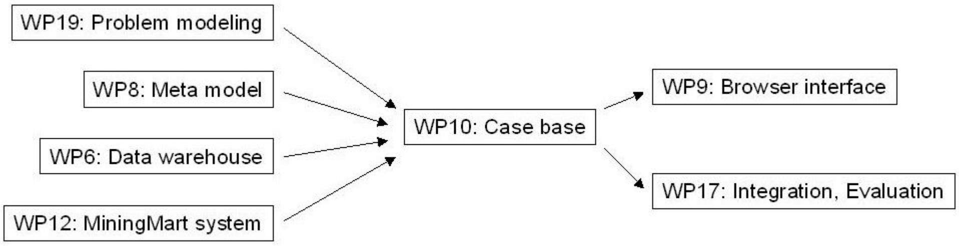


Figure 1.1: The position of workpackage 10 within the Mining Mart project.

about the Mining Mart project and the Mining Mart system. The Mining Mart approach is described in: [MoS02], [ZKV01] and [KZV00]. Further information about the Mining Mart system can be found in: [PeF02] (requirements, system overview), [Eul02] (Compiler and operators), [LaR02] (M4Interface), [VKZ01] (the Mining Mart Meta Model), [Zu01b] (Compiler) and [Zu01a] (M4 schema).

The report is divided into two main parts: first the design and implementation of the case base is described and then we focus on how to use the case base.

Chapter 2

Case Base Object Model

2.1 Which Objects Need to be Stored

In order to capture an application different types of objects need to be stored. An application consists of meta data from a case from the Mining Mart system and objects describing the business level. The meta data from a case consists of objects that are defined in the M4. The M4 has three main parts:

- The conceptual case model. It describes the applied preprocessing operators, their order, and their inputs and outputs. It contains objects such as: Case, Step, Parameter, Operator, and StepSequence.
- The conceptual data model. This part of the M4 is a data abstraction layer. It represents data objects and their relationships. These objects are inputs and outputs for preprocessing operators. The conceptual data model contains objects such as: Concept, FeatureAttribute, and Relationship.
- The relational data model. It gives a close representation of the business data and the relationships in the business data. It contains objects such as: ColumnSet, Column, ForeignKey, and PrimaryKey.

Both the conceptual case model and the conceptual data model are included in the application. The relational data model is, however, not part of an application. The reason for this is that the relational data model closely reflects the business data, which a company will likely not want to share with others.

Figure 2.1 gives an overview of objects involved in the object model for the case base. It clearly shows the division in business level objects and M4 related objects. The business level objects are divided in objects that are more text oriented or descriptive of character on the left and other business level objects on the right. The descriptive objects make it possible

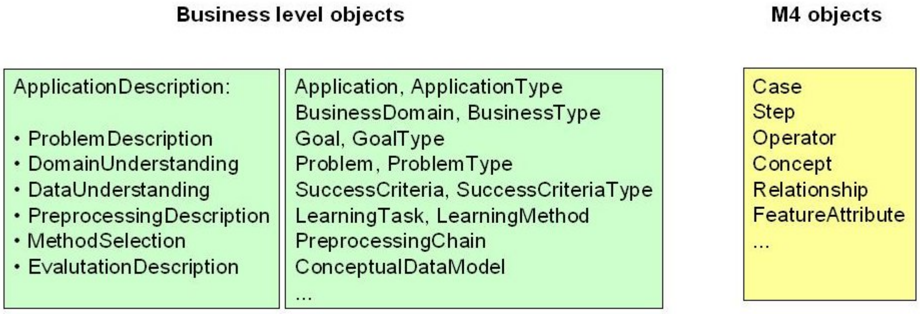


Figure 2.1: Objects from the object model for the case base.

to provide a very readable and clearly structured high level description for an application. The other business level objects provide additional information that is especially suited to support browsing and searching of applications. For a full overview of all objects see Appendix A.

2.2 Relationships Between Objects

In order to support browsing and searching through application information it is necessary to define relationships between objects. A balance needs to be found in the number of relationships, however. Connecting all objects to all other objects is very flexible, but may make the system very crowded. Our goal was to define relationships between objects that logically belong together.

Figures 2.2 and 2.3 show relationships between objects. Figures 2.2 concentrates on relationships between the major business level objects. The two central objects are Application and ApplicationDescription. They are linked to each other and to many of the other objects. Figure 2.3 shows where the business level objects are connected to the M4 objects. So, for example: from a ConceptualDataModel relevant Concepts and Relationships may be found; a PreprocessingChain, Problem, and Goal may be connected to a Step. Note also that Application and Case are linked to each other, as one would expect.

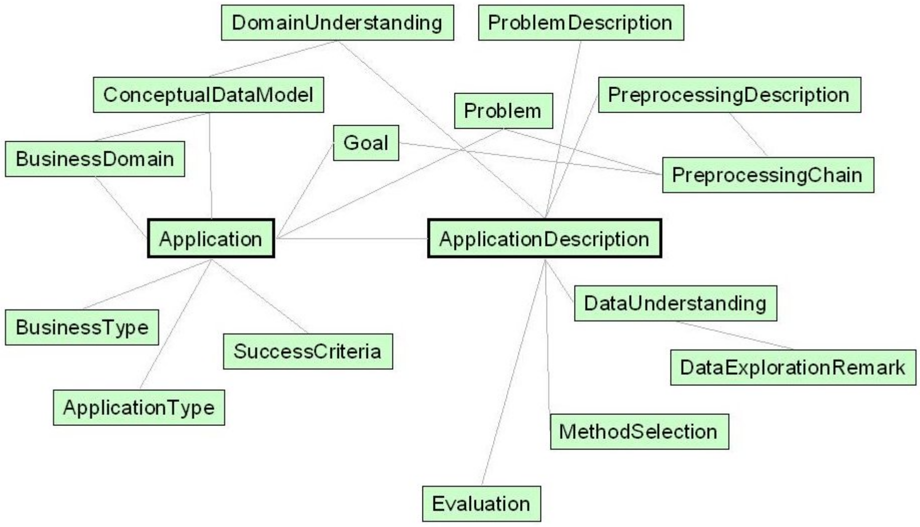


Figure 2.2: Relationships between the major business level objects.

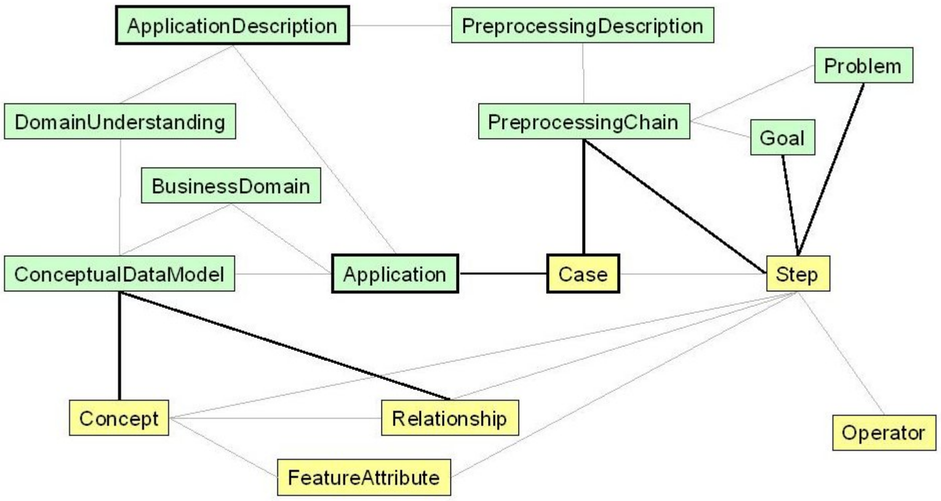


Figure 2.3: Connections between business level objects and M4 objects.

Chapter 3

Case Base Implementation using InfoLayer

It was decided to implement the case base using the InfoLayer[1] tool, which has been developed by the University of Dortmund. The tool well suits our needs. It has the following features:

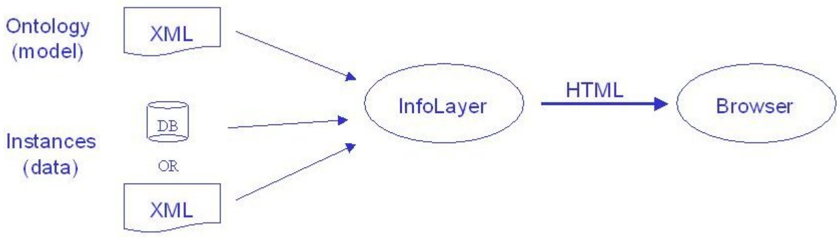
- It works with an ontology (object model). The object model is defined using the XML format. InfoLayer can automatically create HTML pages given an ontology and instances of objects.
- Allows browsing from object to object. The relationships that are defined in the ontology are represented as links between objects.
- Provides an overview of instances of objects as a list of hyperlinks.
- Instances are also defined using an XML format or (in later versions of InfoLayer) can be read directly from a RDBMS. Figure 3.1 illustrates this showing the inputs and output for InfoLayer.
- It can use HTML templates and therefore provides flexibility in how pages are presented to users[2].
- Finally an important feature is that InfoLayer also allows users to add object instances using their browser.

In order to get a feeling for how InfoLayer presents information, some screen shots are presented here. InfoLayer basically presents four type of screens to the user:

- An overview page. This page shows the ontology that is used by InfoLayer (see Figure 3.2).

[1] More information about InfoLayer can be found on the website: http://www.infolayer.org

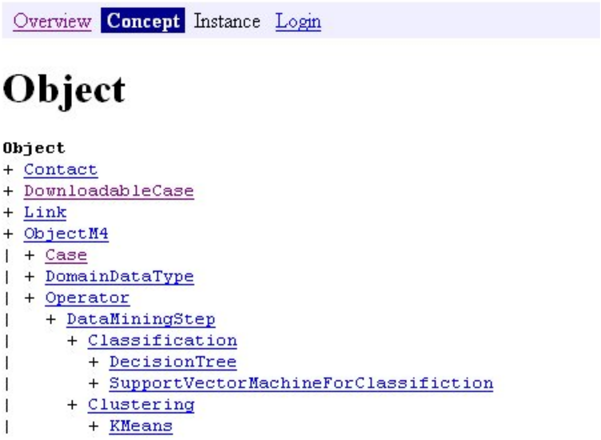
[2] The InfoLayer figures in this report show the default presentation of InfoLayer



The figure shows InfoLayer inputs (object model and object instances) and the output (HTML).

Figure 3.1: Conceptual view of the how application data is presented.

- A concept page. This page shows the concept name, its place in the object hierarchy, and instances available for the concept (see Figure 3.3).
- An instance page. Here property fields and relationships to other object instances are shown. The relationships are presented as hyperlinks allowing to browse easily through the object model (see Figure 3.4).
- An edit page. Here the user can edit and create instances. InfoLayer requires users to provide a username and password to enter the edit mode.

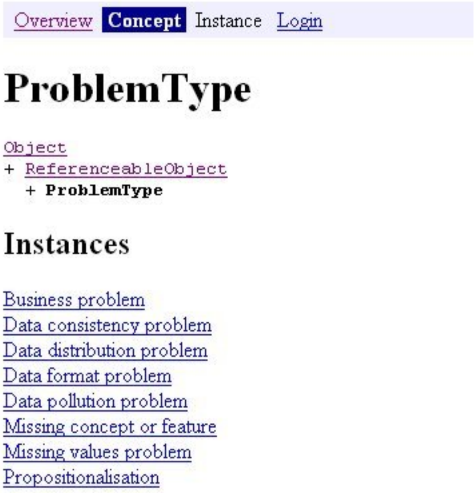


Object

```
Object
+ Contact
+ DownloadableCase
+ Link
+ ObjectM4
| + Case
| + DomainDataType
| + Operator
|   + DataMiningStep
|     + Classification
|       + DecisionTree
|       + SupportVectorMachineForClassifiction
|     + Clustering
|       + KMeans
```

The InfoLayer overview page shows the object hierarchy. "Object" is always placed at the top of the hierarchy. For a full overview of the object hierarchy see Appendix A

Figure 3.2: InfoLayer screen shot of the "overview" page.



ProblemType

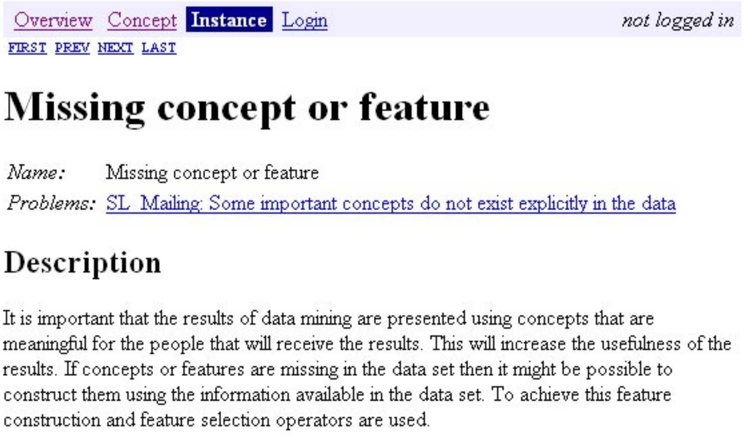
```
Object
+ ReferenceableObject
  + ProblemType
```

Instances

Business problem
Data consistency problem
Data distribution problem
Data format problem
Data pollution problem
Missing concept or feature
Missing values problem
Propositionalisation

The InfoLayer concept page shows a concept (here "ProblemType"), its place in the object hierarchy, and available instances for the concept.

Figure 3.3: InfoLayer screen shot of the "concept" page.



Missing concept or feature

Name: Missing concept or feature

Problems: SL_Mailing: Some important concepts do not exist explicitly in the data

Description

It is important that the results of data mining are presented using concepts that are meaningful for the people that will receive the results. This will increase the usefulness of the results. If concepts or features are missing in the data set then it might be possible to construct them using the information available in the data set. To achieve this feature construction and feature selection operators are used.

The InfoLayer instance page shows property fields for the instance (here "Name", and "Description") and links to related object instances (here "Problems").

Figure 3.4: InfoLayer screen shot of the "instance" page.

Chapter 4

Using the Case Base

The main functionality of the case base is to allow users to: browse through applications or search for specific applications, re-use available applications, and contribute to the case base by adding their own applications. In the following sections we further elaborate on these items.

4.1 Browsing and Searching Applications

A number of objects have been added to the object model to facilitate browsing and searching of applications in different ways. The main objects that have this function are:

- Application. This would normally be the place to start. When selecting the Application concept all Application instances will be listed. The names of the Application instances should be descriptive enough for directly choosing an application.

- ApplicationType. This concept defines application categories. Currently the following categories have been defined: credit risk analysis, customer retention, fraud detection, mailing action, market basket analysis, and optimize control.

- BusinessDomain. The business domain describes the domain of application. Examples are: insurances, loans, mortgages, phone call handling by a telecommunications company, and sailing.

- BusinessType. The BusinessType concept provides the possibility to present applications by the type of business they are related to. The BusinessType concept can be seen as a more generalized version of the BusinessDomain concept. Categories that have been defined are: accomodation and food services, arts, entertainment and recreation, finance and insurance, health care and social assistance, manufacturing, real estate and rental and leasing services, retail, science, telecommunications, transportation, and utilities.

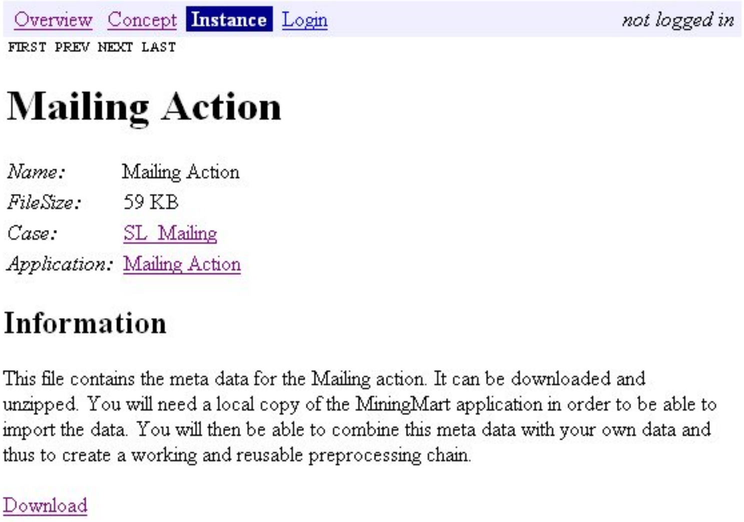
- Goal. The Goal concept allows to specify the goal for an application. An application may have several goals that may be structured in a hierarchy.
- GoalType. Describes the type of goal. Currently only the following types are defined: business goal, and data mining goal.
- Problem. Goals and Problems are closely coupled concepts. Generally a Problem can be reformulated into a Goal. This concept allows to look at an application from a "problem perspective."
- ProblemType. Allows to list problems by type. The following types have been defined: business problem, data consistency problem, data distribution problem, data format problem, data pollution problem, missing concept or feature, missing values problem, propositionalisation.
- SuccessCriteria. Allows to specify success criteria for an application. In a real world project there are always a lot of factors that influence the success of a knowledge discovery project and more specifically the steps taken in the preprocessing stage. These factors can be very diverse. This concept allows to capture such external factors.
- SuccessCriteriaType. Defines types of success criteria. The following types have been defined: cost specification, material specification, resource specification, result specification, and time specification.

These concepts together allow users to search applications or specific parts of applications in various ways and from different perspectives.

4.2 Re-using Applications

The case base should present a set of applications that can be re-used. Related workpackages that are important to make this possible are workpackages 9 and 12. Workpackage 9 provides the Internet access to the case base. Workpackage 12 delivers the Mining Mart HCI. The Mining Mart HCI allows to export and import cases. This functionality is essential for re-use of cases. With an application stored in the case base also an export of the corresponding case should be stored. This will then allow users to download the case file and import it into their local version of the Mining Mart HCI. With the Mining Mart HCI the user may then adapt the case or re-use parts of it as needed.

The case base object model contains the object "DownloadableCase", which allows to present a case file to end users (see Figure 4.1).



Mailing Action

Name:	Mailing Action
FileSize:	59 KB
Case:	SL Mailing
Application:	Mailing Action

Information

This file contains the meta data for the Mailing action. It can be downloaded and unzipped. You will need a local copy of the MiningMart application in order to be able to import the data. You will then be able to combine this meta data with your own data and thus to create a working and reusable preprocessing chain.

Download

Figure 4.1: Example of a download page for a case.

4.3 Availability of Applications

One part of the workpackage was to design and implement the case base; another part was to fill the case base. For two main reasons it has been quite cumbersome (until recently) to fill the case base:

- The Mining Mart system became later available in the project than originally planned. An important part of an application is M4 meta data from a case. The Mining Mart system provides a user-friendly way to create meta data for a case. Without this tool it becomes very difficult and time consuming to construct this meta data.

- The second issue was that the Mining Mart Meta Model was updated a few times in the beginning of this year. This made it difficult to still use an earlier constructed case. Now the M4 has become more stable. This strongly reduces the risk that newly constructed cases may become unusable in the future.

Currently several partners are still working to complete application descriptions. It is expected that at least three complete applications will be shortly available for the case base.

4.4 Creating New Applications

The precise procedure for creating a new application and adding it to the case base using the Internet access to the case base, is still to be established.

We expect, however, that the procedure will involve the following steps:

- Create a case using the Mining Mart system, export the case, and transfer it to the University of Dortmund for storage. The case base will be physically located on a computer from the University of Dortmund. It will consist of the InfoLayer tool connected to an instance of the M4. The University will then store the case in this M4 instance (for example by using the Mining Mart system).

- Create the business level description using the InfoLayer interface. The case export file should be included in the business level description as a "DownloadableCase".

- Connect the business level description to the case meta data. When the case has been stored in the M4 instance at the University of Dortmund, it should be possible to connect business level objects to case objects.

In the following sections we will further focus on creating a business level description and connecting it to the case meta data.

4.4.1 Creating a business level description

Creation of the business level description can be divided into two steps:

- Complete the business level descriptive objects.

- Complete the business level objects that enrich the search functionality.

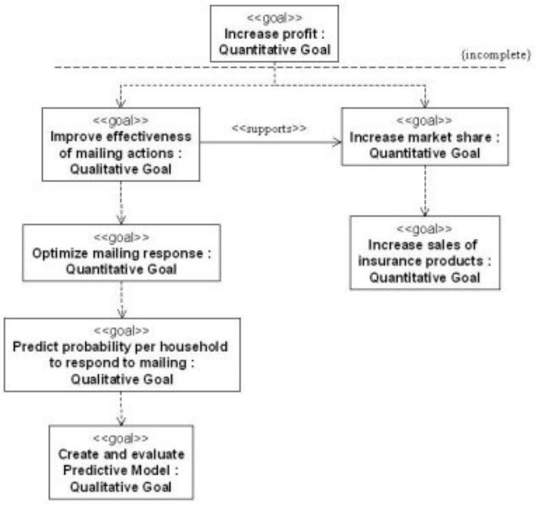
The business level descriptive objects are:

- ApplicationDescription. This concept is a container for the other descriptive business level objects. It contains a field "Description" that should be used to present a short summary (abstract) of the application.

- ProblemDescription. The "Description" field in this concept should be used to clarify what the (business) problem is, which goals are to be achieved and what the plan of approach is. For clarifying goals a goal hierarchy diagram is very useful (see Figure 4.2).

- DomainUnderstanding. This concept explains how goals are connected to relevant business processes and describes the main concepts involved and their relationships. Process diagrams and domain model diagrams can be used to support the description.

- DataUnderstanding. This concept is meant to provide data characteristics that are relevant for preprocessing the data. Useful topics here are for example: overall data quality, missing values, noise, and amount of data.


<<goal>>
Increase profit : Quantitative Goal
{incomplete}
<<goal>>
Improve effectiveness of mailing actions : Qualitative Goal
<<supports>>
<<goal>>
Increase market share : Quantitative Goal
<<goal>>
Optimize mailing response : Quantitative Goal
<<goal>>
Increase sales of insurance products : Quantitative Goal
<<goal>>
Predict probability per household to respond to mailing : Qualitative Goal
<<goal>>
Create and evaluate Predictive Model : Qualitative Goal

The goal hierarchy diagram is based on the Unified Modeling Language. It is described in [ErP00].

Figure 4.2: Example of a goal hierarchy.

- PreprocessingDescription. This section gives an overview of the preprocessing steps that are involved in the application. Process diagrams may also be used here to clarify the steps.
- MethodSelection. Describes which analysis method was chosen and why.
- EvaluationDescription. Evaluates the knowledge discovery process. What are the results? What was learned during the process? What improvements could be made?

After the descriptive part of the business level description is finished, the objects supporting the case base search functionality should be defined. These are objects such as: ApplicationType, BusinessDomain, BusinessType, Problem, ProblemType, Goal, GoalType, and SuccessCriteria. This can be done in a straightforward manner using the InfoLayer interface.

4.4.2 Connecting the business level to the case level

After the case meta data has been stored in the M4 instance of the University of Dortmund and the business level description has been completed, the connection can be made between the business level and the case level. The main connections are between (see also Figure 2.3):

- Application - Case

- PreprocessingChain - Case
- PreprocessingChain - Step
- Goal - Step
- Problem - Step
- ConceptualDataModel - Concept
- ConceptualDataModel - Relationship

This can again be done in a straightforward manner using the InfoLayer interface. Of course knowledge is needed of the specific case in order to know which high level objects should be connected to which M4 meta data objects.

Chapter 5

Conclusions

In this workpackage an object model (ontology) has been created for storing applications. The object model supports storage of M4 meta data and business level information. The business objects can be divided into descriptive type of objects and objects that support browsing and searching the application. The case base has been implemented using InfoLayer, a tool developed earlier by the University of Dortmund. The case base depends on the work of workpackage 9 to become connected to the Internet. Currently the number of stored cases is very limited, due mainly to the late availability of the Mining Mart system. However, partners are still working on completing applications and it is expected that at least three applications will be available shortly.

We believe that the business level objects provide a convenient way to add application descriptions and will improve browsing and search experiences for end users. The InfoLayer system also provides a convenient way of defining and editing object instances. This kind of interface will certainly make it easier for parties to further contribute to the case base.

Bibliography

[ErP00] Eriksson, H.-E., Penker, M., *Business Modeling with UML*, John Wiley & Sons, New York, 2000.

[Eul02] Euler, T., *Compiler Constraints and Operator Parameters*, Technical report TR 12-02, October 30, 2002.

[KZV00] Kietz, J.-U., Zücker, R., Vaduva, A., "Mining Mart: Combining case-based reasoning and multistrategy learning into a framework for reusing kdd-applications", in *Proc. of the Int. Conf. on Multi-Strategy Learning, MSL-2000*, 2000.

[LaR02] Laverman, B., Rem, O., *Description of the M4 Interface used by the HCI of WP12*, Deliverable D12.2, July 16, 2002.

[MoS02] Morik, K., Scholz, M., *The MiningMart Approach.*, Workshop Management des Wandels der 32. GI Jahrestagung, 2002, to appear.

[PeF02] Perot Systems Netherlands, Fraunhofer Institute AiS, *Mining Mart Human Computer Interface*, Technical report TR 12-01, March 15, 2002.

[VKZ01] Vaduva, A., Kietz, J.-U., Zücker, R., Dittrich, K., Morik, K., Marco, B., Luigi, P., *M4 - The MiningMart Meta Model.*, Deliverable, D8/9, IST Project MiningMart, IST-11993, 2001.

[Zu01a] Zücker, R., *Description of the M4-Relational Metadata-Schema within the Database.*, Deliverable D7a, IST Project MiningMart, IST-11993, 2001.

[Zu01b] Zücker, R., *Description of the Metadata-Compiler using the M4-Relational Metadata-Schema.*, Deliverable, D7b, IST Project MiningMart, IST-11993, 2001.

[ZKV01] Zücker, R., Kietz, J.-U., Vaduva, A., "MiningMart: Metadata-Driven Preprocessing.", in *Proceedings of the ECML/PKDD Workshop on Database Support for KDD*, 2001.

Appendix A

Case Base Object Hierarchy

The following list shows the complete object hierarchy as it has been implemented in InfoLayer.

```
Object
+ Contact
+ DownloadableCase
+ Link
+ ObjectM4
| + Case
| + DomainDataType
| + Operator
|   + DataMiningStep
|     + Classification
|       + DecisionTree
|       + SupportVectorMachineForClassifiction
|     + Clustering
|       + KMeans
|     + DeviationDetectionOrSubgroupdiscovery
|       + Midos
|       + Sidos
|     + Regression
|       + SupportVectorMachineForRegression
|   + FeatureConstruction
|     + Discretization
|       + ECDiscretization
|       + EWDiscretization1
|       + EWDiscretization2
|       + UDDiscretization
|     + DualPivot
|     + Grouping
|       + ECGrouping
|       + UDGrouping
|     + MissingValue
|       + AssignAverageValue
|       + AssignDefault
|       + AssignMedianValue
|       + AssignModalValue
|       + AssignPredictedValueCategorial
|       + AssignPredictedValueContinuous
|       + AssignStochasticValue
|       + MissingValuesWithDecisionTree
|       + MissingValuesWithRegressionSVM
|     + Pivot
|     + Scaling
|       + LinearScaling
|       + LogScaling
|   + FeatureSelection
|   + MultiRelationalFeatureConstruction
|     + Aggregation
|     + Chaining
|     + Propositionalization
|   + RowSelection
|     + Sampling
|       + RandomSampling
|       + StratifiedRandomSampling
|     + Segmentation
|       + NNSegmentation
|       + Partitioning
|       + StratifiedSegmentation
|     + SelectCases
|       + DeleteRecordsWithMissingValues
|       + SelectByQuery
|   + TimeOperator
|     + MovingFunction
|       + ExponentialMovingFunction
|       + WeightedMovingFunction
|     + SignalToSymbolProcessing
|     + Windowing
| + OperatorConstraint
| + Parameter
|   + Concepts
|   + FeatureAttribute
|     + BaseAttribute
|     + MultiColumnFeature
|   + Relationship
|   + Value
| + RoleRestriction
| + Step
|   + LoopStep
|   + MultiStep
| + UserInput
+ Person
+ Publication
| + Article
| + Book
| + Journal
| + Proceeding
+ Publisher
+ ReferenceableObject
+ Algorithm
+ Application
+ ApplicationType
+ BusinessDomain
+ BusinessType
+ ConceptualDataModel
+ DataExplorationRemark
+ DataExplorationRemarkType
+ DescriptionObject
  + ApplicationDescription
  + DataUnderstanding
  + DomainUnderstanding
  + Evaluation
  + MethodSelection
  + PreprocessingDescription
  + ProblemDescription
  + Goal
  + GoalType
  + LearningMethod
  + LearningTask
  + PreprocessingChain
  + Problem
  + ProblemType
  + QualityMeasure
  + SuccessCriteria
  + SuccessCriteriaType
  + System
```