A Visual Data Mining Environment*

Stephen Kimani, Tiziana Catarci, and Giuseppe Santucci

Universitá di Roma "La Sapienza",
Dipartimento di Informatica e Sistemistica,
Via Salaria 113, 00198 Roma, Italy
{kimani, catarci, santucci}@dis.uniroma1.it

Abstract. It cannot be overstated that the knowledge discovery process still presents formidable challenges. One of the main issues in knowledge discovery is the need for an overall framework that can support the entire discovery process. It is worth noting the role and place of visualization in such a framework. Visualization enables or triggers the user to use his/her outstanding visual and mental capabilities, thereby gaining insight and understanding of data. The foregoing points to the pivotal role that visualization can play in supporting the user throughout the entire discovery process. The work reported in this paper is part of a project aiming at designing a Data Mining system with a visual environment that supports the user in the entire process of mining knowledge.

1 Introduction

It should be acknowledged that a lot of research work has been and is being done with respect to knowledge discovery. However, much of the work concentrates on the development and optimization of data mining algorithms using techniques from other fields such as Artificial Intelligence, statistics and high performance computing, with little consideration, if any, of the other knowledge discovery phases. Consequently, corresponding tools/systems are normally difficult to integrate in the entire knowledge discovery process.

There is a major need to develop an overall framework that can support the entire knowledge discovery process[1]. The framework should accommodate and integrate all the phases seamlessly. Since it is not known *a priori* all the components the framework will be expected to support, the framework should be extensible to any new components. On the same note, the development of the framework and the components should be separated.

One of the interesting issues related to the ongoing discussion is the role of visualization in the knowledge discovery process. Visualization is a very effective means of enabling the user to use his/her outstanding perceptual capabilities to recognize and understand data. Traditionally, visualization has been placed at the beginning and at the end of the knowledge discovery process. Instead, visualization has its place in all the phases of the knowledge discovery process. This

* This work is supported by the MIUR project D2I: *Integration, Warehousing, and Mining of Heterogeneous Sources (http://www.dis.uniroma1.it/~lembo/D2I)*

puts visualization, and therefore the user, at the center of the entire knowledge discovery process. This is a major step toward developing user-centered data mining systems.

This paper describes a Data Mining system with a visual environment aiming at supporting the user throughout the entire data mining process. The visual data mining environment employs a wide range of novel and intuitive visual strategies toward realizing the foregoing aim.

The rest of the paper is organized as follows: section 2 focuses on related research work. Section 3 describes the architecture of the proposed Data Mining system. A detailed description of the visual data mining environment is presented in section 4. Section 5 highlights efforts aimed at defining a mapping between the abstract data mining engine and the visual interface. Work on usability studies is presented in section 6. Future work and a conclusion are presented in section 7.

2 Related Work

In this section, a discussion of some data mining systems that offer a reasonably great and diverse number of data mining and visualization functionalities that have been proposed in the literature is given.

Clementine [8] is a product by Integral Solutions Ltd (ISL). SPSS purchased ISL on December 31, 1998. Clementine provides various data mining algorithms to support various techniques including; clustering, association rules, sequential patterns, factor analysis, and neural networks. The product is easy to use through its visual programming interface.

The FlexiMine system [3] has been designed as a test bed for data mining research. The system is also intended to serve as a generic knowledge discovery tool. At the time of going to writing and to the best of our knowledge, FlexiMine contains algorithms for handling association rules, Bayesian networks, decision trees, and metaqueries.

Another related research effort is the QUEST project [9] at IBM Almaden Research Center. The project was intended to discover useful patterns in large databases. The research effort provides support for a notably wide range of data mining algorithms including association rules, sequential patterns, classification, and time-series clustering. IBM markets the technology using the commercial product DB2 Intelligent Miner.

MineSet [7] is a data mining system developed by Silicon Graphics Inc. MineSet supports association rules and classification models. It uses these models for carrying out prediction, scoring, segmentation, and profiling tasks. Besides its application of robust data mining algorithms, MineSet is notable for its sophisticated visualizations.

GGobi [11] is an interactive data visualization system for exploratory data analysis. The system is the result of significant redesign of its predecessor, XGobi [10]. One of the interesting new features supported by GGobi is the plug-

in functionality. Consequently, GGobi can accommodate functionalities from other applications and/or it can be deployed in other applications.

Viscovery SOMine [12] is a data mining system developed by Eudaptics. Among other data mining methods, the system supports clustering, prediction, regression and association. Viscovery SOMine provides an interactive environment in a bid to support the user in the data mining process.

Another data mining system is Decision Series [6] by Neo Vista Solutions Inc. Accrue Software Inc acquired Neo Vista in February 2000, and has used Decision Series to support analysis applications. On June 27, 2001, Accrue sold Neo Vista intellectual property to JDA Software Inc. Through the sale, JDA assumes the intellectual property of the Decision Series data mining toolset and the RDS-Assort and RDS-Profile applications. Decision Series supports clustering, association rules, and neural networks.

Each of the foregoing systems exhibits at least, either one or both of the following limitations:

- *Non-extensible framework:* The system is based on a framework that supports only some specific phases in the data mining process. Consequently, it is extremely difficult, if not impossible, to incorporate new components.
- *Non-homogeneous environment:* The system makes use of a non-uniform mining environment. The user is presented with "totally" different interface(s) across implementations of different data mining techniques.

Our Data Mining system takes on an integrated approach in terms of its framework. Moreover, the system is geared toward providing the user with a consistent, uniform and flexible interaction environment across the entire process of data mining.

3 The Proposed Data Mining System

At present, the system supports, but is not limited to: metaqueries, association rules, and clustering.

The architecture of the system comprises two primary layers: the user layer, and the data mining engine layer, as seen in Figure 1.

1. The *Parametric User Interface/User Layer* enables the user to interact with the other system components. It invokes the relevant system feature or functionality on behalf of the user. Ideally, this layer/component empowers the user to process data (and knowledge), and also to drive, guide and control the entire discovery process.
   The user component is organized around a GUI container which hosts specific GUI extension cartridges, which in turn contain the knowledge to access their respective underlying data mining components/modules in the *Data Mining Engine Layer*. In effect, the GUI container registers the specific data mining technique GUI extension, loading respective specific menu items and other commands specific to the data mining technique. For instance, the specific

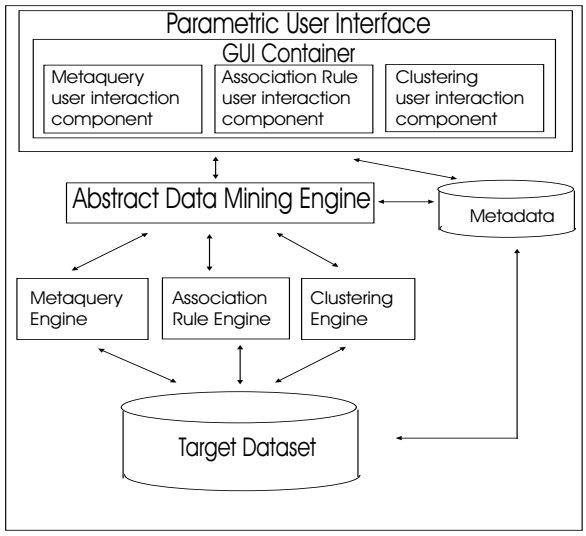

Fig. 1. System Architecture

data mining technique GUI extension for clustering has the knowledge on how to access the clustering engine and also on how to interact with the user in the acquisition of clustering input and in the visualization of clustering output.

The GUI container provides various services to the Data Mining system. These services fall under two categories: infrastructural services and end user services.

The infrastructural services that are supported include:

- Registration of extensions which implement specific interaction contracts.
- Runtime loading on the interaction environment of the features that are relevant to the active GUI extension (e.g. commands and options).
- Advertising new GUI extensions.
- Routing user commands to the active GUI extension.

As for the end user services, the Data Mining system support includes:

- Providing the user with a uniform, consistent and flexible user interface.
- Providing services whose use span across the entire data mining interaction environment (such as start, stop, save and load services).

There are various functionalities that a data mining GUI extension supports. The GUI extension carries out the specific commands that are loaded and made available to the user (loaded on the interface) by the GUI container. The extension also implements specific input and output modalities for the underlying data mining technique or algorithm(s). On the whole, modalities specific to the data mining technique may be added or may substitute some or part of the general pre-existing ones.

2. The *Abstract Data Mining Engine Layer* is completely decoupled from the *User Layer*. However, the structure of the Data Mining engine depicts parallelism with the structure of the GUI.
   The *Data Mining Engine Layer* is structured using an abstract reference model based on the following concepts:
   - A global dataset which contains the data to be mined and all the information necessary to apply and execute a data mining technique (such as metadata information).
   - A command manager which forms the interface between the Data Mining engine and the *User Layer*. On one hand, it interprets user commands originating from the user interface and on the other hand it manages any access to the internal structure of the engine. The command manager therefore serves as a two-way link between the engine and the GUI (the GUI container and the specific GUI extension). Some of the operations performed through the command manager include: defining the initial set of target data, applying some data mining algorithm, storing hypothesis and verifying hypothesis.
   - An abstract Data Mining discoverer for carrying out the discovery data mining goal. The discovery task might use inputs directly given by the user or from previous data mining results. This part of the engine must be specialized to implement some specific algorithm.
   - An abstract Data Mining verifier for carrying out the verification data mining goal. Like the abstract Data Mining discoverer, the abstract Data Mining verifier also must be specialized to implement some specific algorithm.

   The general behavior of the engine is abstract in that, it must be instantiated/specialized to specific "engines", one for every data mining technique. It should also be pointed out that a hypothesis that is discovered and/or verified by one specific "engine" can be used by another "engine". Consequently, the result of some data mining task can be used as input to another task. The instantiated "engines" are made available to the general engine framework dynamically. As a consequence, they are also made available to the user through the specific/respective GUI environment.

It is worth pointing out that there are some services that are available to every specific "engine". Such services include: metadata management, configuration savings, intersystem communication[1], data access and database connection management.

As already mentioned, the architecture supports the incorporation of new and the modification of pre-existing components. Specific extension points are defined right where such component additions or modifications occur. It is envisioned that new components will be incorporated as plug-ins[1].

In the current implementation, the *User Layer* is developed using JBuilder. It runs on Windows operating system. The command manager, which forms an

[1] Intersystem communication deals with the management of the possible interactions and data transfer of different data mining techniques in a uniform way

interface between the *Data Mining Engine Layer* and the *User Layer*, is being developed using XML DTDs, as indicated later in section 5. The specific data mining "engines" are implemented using Delphi. It should be mentioned that the "engines" correspond to specific data mining algorithms.

4 The User Interface

The visual interface is designed based on the goal of supporting the user in the entire data mining process. Moreover, the interaction environment is consistent, uniform and flexible. The interface employs various visual strategies that can effectively enable the user to exploit his/her powerful visual capabilities with a view to discovering knowledge through metarules, association rules and clustering.

Toward describing the system features, we consider a communications company that provides various services such as Web-access services, telephone services, etc. The company has a main office and a number of service centers. The main office principally deals with strategic and administrative issues. In fact, the company offers its services through the service centers. The company plans to introduce some special offers. The marketing director is expected to recommend the type of service to be featured and the customers to be targeted. Assume that the marketing head decides to mine some information using the Data Mining system. In this case, we may view him as the user of the system.

The marketing director might want to identify regions that had relatively good general service sales in the last one month. They might want to use that information further to propose some specific service and customers that might be worth consideration in the offer. The recommendation could also include another service that normally does best with the proposed service.

4.1 Identifying Regions With Good Sales: Using the Clustering Environment

Understanding how different regions have been doing can be resourceful in making marketing decisions. The task can be accomplished through the clustering environment.

As a user, one starts by specifying a target dataset. The specification relies on two intuitive interaction spaces namely the specification space and the target space. This may be seen in all the figures illustrating the visual interface (e.g. in the top-left part of Fig. 2). The specification space provides the mechanisms, tools and resources necessary for visually building the set of task relevant data. The target space holds or hosts the relations that are part of the task relevant dataset. The latter space may be envisaged as a container for the constructed target dataset. The two spaces are backed with drag and drop mechanisms and tools. Moreover, the two spaces are complementary in the manner in which they support the user. Therefore, the user operates by appropriately moving between the two components. Since the interface supports drag-and-drop mechanisms,

the user may intuitively move elements (such as relations and relation items) from one component to the other as appropriate.

In this task, the marketing executive is mainly interested in customers and services (i.e. based on relations *Customer* and *Service* or on relation *CustServ*). The company already has geographical information pertaining to customer addresses. For instance, their *loci* with respect to the main office. The user may construct a relation in which the attributes of interest are *CustID*, *CustX*, *CustY*, *CustAmt* and *ServID*.

The Data Mining system provides an interaction environment with various input widgets through which the user can specify parameters characterizing a clustering task. For instance:

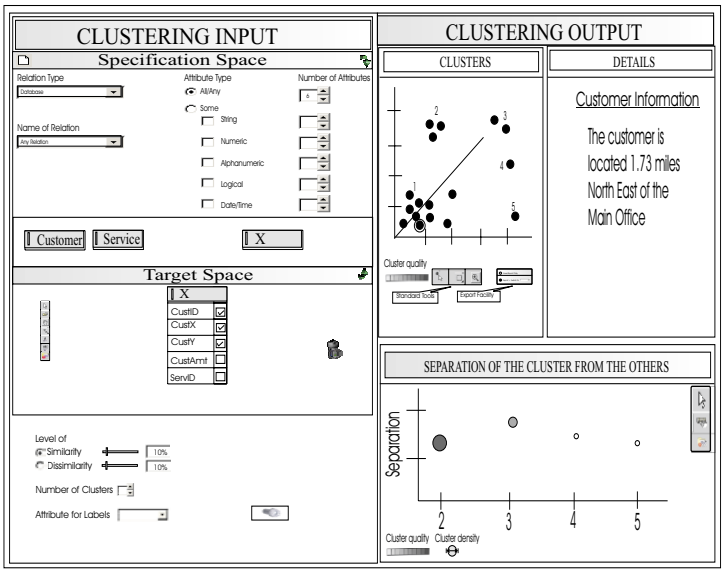

Fig. 2. The Clustering Environment

- Radio buttons for the specification of homogeneity or separation measures. Each of the two measures is presented on an ordinal scale with a slider control.
- A palette-based mechanism for specifying an attribute value.
- A "checking" mechanism for specifying attributes that will determine the partitioning of the target dataset. In Fig. 2, the attributes *CustID*, *CustX* and *CustY* have been selected ("checked").
- A combo-box for specifying an attribute to be used in labeling clusters (In Fig. 2, the labeling attribute has not been specified. In such a case, the system would default to numbering the clusters serially).

When through with the parameter specification, the marketing head may instruct the system to partition the target dataset by clicking the "torch" icon. The system performs the clustering and displays the results.

As regards clustering, the system offers two main visualization mechanisms: *Clusters + Details ("Overview + Detail")* and *Separation View*.

Clusters + Details ("Overview + Detail")
This visualization displays clusters on a scatter plot, and also presents details that correspond to a selected cluster or cluster object. The former display corresponds to an "Overview" window whereas the latter corresponds to a "Detail" window.

Cases are mapped to points on the scatter plot, with each point taking some x, y (and if appropriate z) values. The system uses two alternative encoding approaches with regard to the scatter plot display. In the first approach, objects that belong to the same cluster are optionally enclosed in a bounded region. Each such region is shaded with some grayscale level that reflects the accuracy level of the represented cluster. In the approach, outliers are drawn positioned outside the bounded regions. A currently selected cluster, cluster object, or outlier is drawn with an outline around it. The points themselves may be encoded to reflect some other aspects (e.g. by coloring). The top-right part of Fig. 2 shows a visualization in which the scatter plot is generated using this first approach and in which the user has opted to have no enclosures. The values on the x-axis correspond to *CustX*, those on the y-axis correspond to *CustY* and the z-axis values correspond to *CustID*. In the second approach, objects that belong to the same cluster are drawn using the same color saturation level. The color saturation level also represents the accuracy of the represented cluster. In this approach, outliers are all drawn using one color. The outlier color is reserved for that purpose and therefore is not used to map any other aspect. The system determines the approach to use based on the distribution of cluster objects. The "Detail" window effects the exposition of a selected cluster or point. The interface relies on a system-driven mechanism which determines an appropriate presentation style for the details of the selected entity.

Separation View
Measures of accuracy are useful in many ways (e.g. for interpretability and evaluation purposes). The system currently supports a display based on a separation function. The visualization is a graph depicting how far the selected cluster (or the cluster containing the currently selected point) or outlier is from the other clusters and outliers e.g. the bottom-right part of Fig. 2 shows how far the cluster containing the outlined object is from the other clusters and outliers. The value of separation is mapped to the y-axis. A circle encodes a cluster or an outlier. The circles are arranged along the x-axis. The density of the cluster or outlier is bound to the size of the circle. The grayscale level of a circle represents the quality of the represented cluster or outlier.

From the *Clusters + Details* and *Separation View* visualizations, regions that are close to the main office depict a lot of sales. With regard to the anticipated offer, a marketing strategy might put a lot of emphasis on people and service centers that are close to the main office.

The marketing executive might want to gain more knowledge from those interesting regions. For instance he might want to identify some specific service and customers within those particular regions. The task would entail establishing data relationships. The analysis can be done through the metaquery environment.

However, it is important to observe that the task is based on some particular subset of data which is not equivalent to the currently defined set of target data. In other words, the user intends to use some output from one task (clustering) as an input to another task (metaquerying).

The interface enables the user to select points or clusters of interest through the use of the *Standard Tools* toolbox. The marketing director may then turn to the *Export Facility*. The resource would enable him to specify whether he would want to just save the specified output or to save and switch to another task with that output as the input to the new task. In the latter case, the system switches to the new environment with the output appearing in the Target Space.

In the ongoing example, the relation *ClustOutl* in Figure 3 represents the clustering output that has become metaquerying input.

4.2 Establishing Data Relationships: Using the Metaquery Environment

The marketing director will need to analyze the relationships that exist among services, customers, and centers. The analysis would help him to determine the service to feature and potential customers. The metaquery environment can be helpful in carrying out the analysis. Such relationships can be mined by exploiting the relations $CustCent$, $CustServ$, and $ServCent$. Assume that the user is interested in the following attributes: $CustCent.CustID$, $CustCent.CentID$, $CustServ.CustID$, $CustServ.ServID$, $ServCent.ServID$ and $ServCent.CentID$. Therefore the marketing director needs to specify the three relations with the foregoing attribute constraints toward constructing the target dataset. It should be mentioned that the metaquery analysis will be restricted to only the tuples contained in the data that was "imported" from the clustering task (tuples in the relation *ClustOutl*), which is already in the target space.

In the environment, the user may define links/"joins"[2] manually (*Manual Joins*) or have the system automatically do that (*Automatic Joins*). Assume that the marketing director chooses the latter option. The system links the attributes as follows:

– $CustCent.CustID$ with $CustServ.CustID$

[2] As far as our designing of metaqueries is concerned, "joins" do not refer to table joins. A "join" refers to a link between attributes that is aimed at generating a consequent pattern

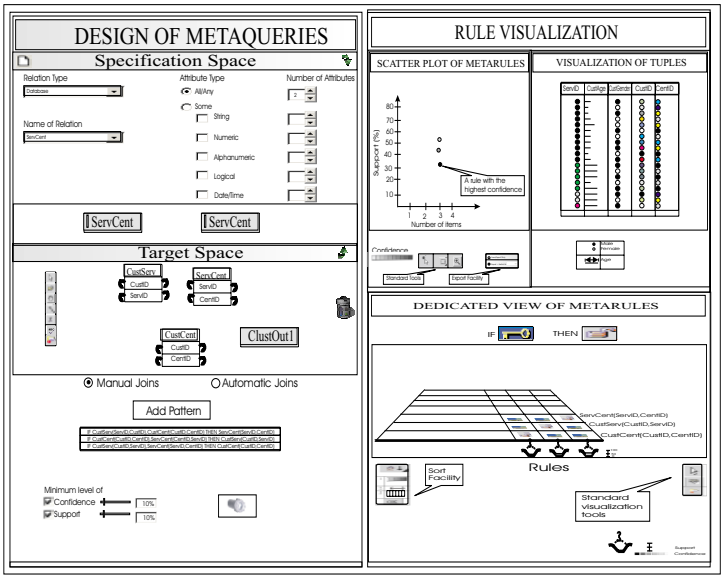

Fig. 3. The Metaquery Environment

– $CustCent.CentID$ with $ServCent.CentID$
– $ServCent.ServID$ with $CustServ.ServID$

Letting X be a representation for $CustID$, Y for $CentID$, and Z for $ServID$, and allowing reordering of attributes, the system generates the following transitive "combinations" (which are actually metaqueries):

1. $CustCent(X,Y), CustServ(X,Z), ServCent(Z,Y)$
2. $CustServ(X,Z), CustCent(X,Y), ServCent(Y,Z)$
3. $ServCent(Z,Y), CustServ(Z,X), CustCent(X,Y)$

The system puts the patterns in a pool as seen in Figure 3. The director may also specify confidence and support values by using sliders or text-boxes. He may then instruct the system to search for specific rules from the target dataset that correspond to the metapatterns in the pool and that satisfy the specified parameters, by clicking the "torch" icon.

The Data Mining system provides various visualizations for the search results. For any rule, the aspects that are of principal interest include: measures of interestingness, relationship between the head and body, and details about the items participating in the rule. The system provides two main visualizations for the search results *Rules + Tuples* (Overview + Detail) and *Dedicated View*.

Rules + Tuples (Overview + Detail)
This visualization displays all the rules from the search operation, and also

presents tuples that correspond to some selected rule(s). The rules are displayed using a scatter plot. The interface invokes a system-driven mechanism which chooses an appropriate presentation style for the tuples.

The scatter plot may be envisaged as the "Overview" window and the tuples display as the "Detail" window. On the scatter plot, a rule is mapped to a point, the confidence of a rule to grayscale, the support of a rule to the y-axis, and the number of items in a rule to the x-axis. Consider that the metaquery search based on the ongoing example returns the following results:

1. $CustCent(CustID, CentID) \leftarrow$
   $CustServ(CustID, ServID), ServCent(ServID, CentID)$
   Support = 33.33% and Confidence = 100%
2. $CustServ(CustID, ServID) \leftarrow$
   $CustCent(CustID, CentID), ServCent(CentID, ServID)$
   Support = 44.44% and Confidence = 75%
3. $ServCent(ServID, CentID) \leftarrow$
   $CustServ(ServID, CustID), CustCent(CustID, CentID)$
   Support = 55.56% and Confidence = 60%

In Fig. 3, there is a *Rules + Tuples* visualization of the foregoing results. In the visualization, the marketing director has selected the rule with the highest confidence for exposition. The tuples window depicts some interesting trends. The service represented by black circles had the highest demand. The marketing head may interact with the display, for instance by using the exposition tools or by pointing the circle(s), thereby getting to know that the interesting service is *WWW-Access*. The display also depicts that, virtually all the customers who requested the *WWW-Access* service are young.

Consequently, the marketing director may wish to consider *WWW-Access* service for the anticipated offer. Moreover, he has fairly substantial information regarding the customers to target: those living *close* to the main office and who are of *young* age.

Dedicated View
Like the foregoing visualization, the *Dedicated View* enables the user to visualize all the rules from the search operation. However, in this case, rules are visualized in a more elaborate manner. The *Dedicated View* displays: the confidence and support values of each rule, the relationship between the head and the body of each rule, and the individual items/components that make up each rule. The visualization uses a simple 2D floor with a perspective view. The floor has rows and columns. A rule is represented by a column on the floor. The rule is made up of the components which have entries in the column. The rows represent the items (such as attributes). Associated with each column/rule is a "bucket". The gray value of the contents of the "bucket" represents the confidence value of the rule. The level of the contents of the "bucket" is bound to the support value of the rule. The handle of the "bucket" can be used to select the corresponding rule. Rule items that form the antecedent are each represented using a "key" icon,

whereas those that form the consequent are each represented using a "padlock" icon. The visualization can be seen in the bottom-right part of Figure 3.

4.3 Market-Basket Analysis: Using the Association Rule Environment

The marketing director might also intend to find out another service that is frequently requested every time *WWW-Access* service is requested. Such knowledge would be instrumental in making some marketing decisions. For instance in designing advertisements that capture the two products. The analysis can be realized by switching to the association rule environment. It is worth mentioning that if the user would be interested in switching to the the new environment with the previous output as the new input, he could use the *Export Facility* that was described at the end of section 4.1. One of the distinct features in the

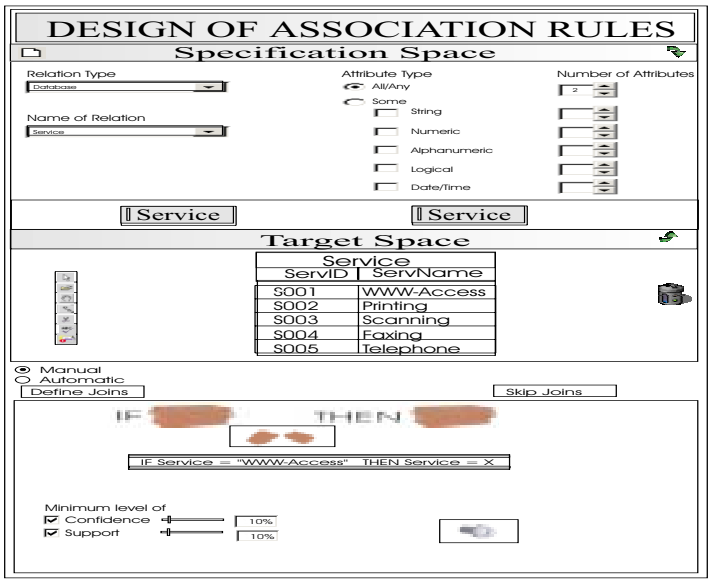

Fig. 4. Basket-based Construction of Association Rules

association rule environment is the provision of "market baskets". It is interesting to note that the interface allows the marketing director to formulate the quest without having to understand the transaction details. For this task, it is enough for him to just have the *Service* relation and constrain it to attributes *ServID* and *ServName*. The resultant relation is seen in the target data space of Fig. 4. Toward specifying the structure of the association rule of interest, the marketing director would drag and drop the tuple *Service = "WWW-Access"*

into the first "basket". He may leave the second "basket" empty as a generic service entry that the system will later instantiate with various relevant service entries. Figure 4 shows part of the association rule environment. In the figure, the marketing director already has put the item into the *IF* "basket" and has emptied the "baskets" into the pool. The user may specify threshold levels and then instruct the system to carry out a search based on the foregoing inputs. The system returns association rules that satisfy the specified parameters.

The association rules are visualized using the same mechanisms that are used for visualizing metarules. By observing the association rule having outstanding measures of interestingness in the visualization, the marketing director may be able to determine the best service to associate with *WWW-Access*.

5 Mapping

Defining a mapping between the abstract components of a system and the corresponding visual ones is beneficial in many ways. For instance, such a definition facilitates data exchange, process automation, data storage and capturing of semantics. It is on the basis of the foregoing understanding that we have embarked on an effort to develop a mapping language for the Data Mining system[2]. At the moment, we have realized some preliminary definitions for metaqueries and clustering.

With regard to metaqueries, we have developed an initial version of MIF (Metarules Interchange Format). MIF may be envisaged as a two-way link for exchanging metarule-based information between any applications that deal with metarules. In the context of our Data Mining system, MIF would facilitate communication between the Visual Interface and the metaquery engine, Metaquery Evaluation Engine (MEE). The Visual Interface generates XML input documents, written in MIFIn format. A MIFIn document specifies the inputs (or contents of a request) for a metaquery task. The MEE produces XML output documents, written in MIFOut format. A MIFOut document contains the answer to a metaquery request. The document can be displayed by the Visual Interface.

The Data Mining Group has recently developed an industrial XML-based standard for the exchange of results between mining applications named PMML, an acronym for *Predictive Model Markup Language*[4]. PMML 2.0 is almost entirely satisfactory with regard to the description of output from a clustering task. However, the specification has no sufficient provision for the description of input to a clustering task. We propose an XML DTD that specifies input to a clustering task. As for output, we have carried out an evaluation of PMML 2.0 and consequently defined an update that supports clustering output description.

6 Usability

To determine the usability of interfaces, it is necessary to subject them to rigorous evaluation tests. It should be pointed out that our system primarily targets

users who are acquainted with data mining. This user audience is specialized and it may be reasonable to consider them as expert users. It would arguably be easier to design an interface for a specific type of users than for a mixed audience. Nonetheless, the need to carry out usability tests remains. As a way of getting started, we carried out usability heuristics. The term "usability heuristics" refers to a more informal evaluation where the interface is assessed in terms of more generic features. This informal evaluation presents reasonably concise and generic principles that apply to virtually any kind of user interface. In the following discussion, we analyze how some of the principles have been applied in the design of the Data Mining system.

- The interface dialogue should be simple and natural. Moreover, the interface design should be based on the user's language/terms. In general, there should be an effective mapping between the interface and the user's conceptual model. In our system, the interface primarily uses data mining terms. It is worth recalling that our target audience comprises users who are conversant with data mining concepts. Furthermore, the provision of "hooks" and "chains" for linking attributes, "baskets" for designing association rules, "drag and drop" mechanisms, "buckets" for measures of interestingness, and "keys" and "padlocks" for antecedents and consequents are part of getting effective mappings between the interface and the user's conceptual model.
- The interface should shift the user's mental workload from the cognitive processes to the perceptual processes. Our Data Mining interface supplies various mechanisms to support the shift. For practically all inputs, the user does not have to supply the units of measurement. Moreover, the system offers interaction controls (e.g. sliders) for helping the user get familiar with the range of valid values and also for helping him/her input within the range. Furthermore, visually presenting query parameters (e.g. data relations) minimizes the possibility of making mistakes while formulating a query.
- There should be consistent usage and placement of interface design elements. Consistency builds confidence in using the system and also facilitates exploratory learning of the system. In our interface, the same information is presented in the same location on all the screens.
- The system should provide continuous and valuable feedback. One of the mechanisms our Data Mining system uses to provide feedback is realized when the user puts some item into the "baskets" or empties the "baskets". The "baskets" respond to reflect the insertion or the removal.
- There is a need to provide shortcuts especially for frequently used operations. In the Data Mining interface, there are various shortcut mechanisms, for instance double clicking and single key press commands.
- There are many situations that could potentially lead to errors. Adopting an interface design that prevents error situations from occurring would be of great benefit. In fact, the need for error prevention mechanisms arises before (but does not eliminate) the need to provide valuable error messages. Our Data Mining interface offers mechanisms to prevent invalid inputs (e.g. specification by selection, specification through sliders). It also provides some

status indicators (e.g. when an item is put in a "basket", the status of the "basket" changes to indicate containment).

Moreover, on informal user tests, we performed some informal user tests on a previous version of the prototype with data mining experts from the universities of Bologna [13] and Calabria [14]. We got encouraging results from the tests and even suggestions on how to improve the interface. For instance, the data mining experts suggested that the interface should provide an optional interaction environment specifically designed for the expert user and still leave the user with the freedom to switch between the two.

At present, we are designing formal usability experiments for the current version of the prototype. Consequent results would be instrumental in determining further relevant interface improvements and modifications.

7 Future Work and Conclusion

There are plans to incorporate an optional visual environment designed for the expert user. Based on the understanding that similarity queries can significantly improve the interactivity of a data mining process, we are planning to incorporate such support. Moreover, there are plans to incorporate visualization systems/tools into the system for the exploration of data mining results. One of the visualization systems that we are intending to use is the DARE system [15] [16]. However, at present, the incorporation has not been done yet. Work on formalization and usability studies is still going on. At the moment, there is a partial prototype of the Data Mining system. The complete implementation of the same is underway. In this paper, the need for a framework that supports the entire discovery process has been discussed. The paper has also highlighted the pivotal role that visualization plays in such a framework. A Data Mining system with a visual environment that is aimed at supporting the user in the pursuit of knowledge has been described.

References

1. Fayyad, U., Grinstein, G. G., Wierse, A. (eds.): Information Visualization in Data Mining and Knowledge Discovery. Morgan Kaufmann Publishers (2002)
2. Catarci, T., Ciaccia, P., Curci, V., Kimani, S., Ianni, G., Lodi, S., Palopoli, L., Patella, M., Santucci, G., Sartori, C.: Visual Data Mining System Architecture. Technical Report D3.R2 - D2I, Integration, Warehousing, and Mining of Heterogeneous Data Sources, Italian MIUR Project, `http://www.dis.uniroma1.it/~lembo/D2I` (2001)
3. Domshlak, C., Gershkovich, D., Gudes, E., Liusternik, N., Meisels, A., Rosen, T., Shimony, S. E.: FlexiMine - A Flexible Platform for KDD Research and Application Construction. Technical Report FC9804, BenGurion University (1998)
4. The Data Mining Group: PMML 2.0 - Predictive Model Markup Language `http://www.dmg.org/pmmlspecs_v2/pmml_v2_0.html`
5. SGI `http://www.sgi.com`

6. Accrue Software Inc. http://www.accrue.com
7. SGI http://www.sgi.com
8. SPSS: Clementince http://www.spss.com/clementine
9. IBM: Quest http://www.almaden.ibm.com/cs/quest
10. Swayne, D. F., Cook, Buja, A.: XGobi - Interactive dynamic graphics in the X Window System with a link to S, Proceedings of the Section on Statistical Graphics. American Statistical Association, 1992.
11. GGobi Data Visualization System http://www.ggobi.org
12. Eudaptics Software gmbh http://www.eudaptics.com/technology/index.html
13. http://www-db.deis.unibo.it
14. http://www.deis.unical.it
15. Catarci, T., Santucci, G., Costabile, M. F., Cruz, I. F.: Foundations of the DARE system for Drawing Adequate Representations, Proceedings of the International Symposium on Database Applications in Non-Traditional Environments. IEEE Press, 1999.
16. Catarci, T., Santucci, G.: The prototype of the DARE System, Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data. ACM Press.
17. Baldonado, M., Chang, C.-C.K., Gravano, L., Paepcke, A.: The Stanford Digital Library Metadata Architecture. Int. J. Digit. Libr. **1** (1997) 108–121