1 User environment for Data Mining Technique
Application: DAMA Prototype

1.1 General Specification

<table>
<thead>
<tr>
<th>Tema</th>
<th>User Environment for Data Mining: Prototype</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>D3.P4</td>
</tr>
<tr>
<td>Date</td>
<td>September 30, 2002</td>
</tr>
<tr>
<td>Product Type</td>
<td>Software Application</td>
</tr>
<tr>
<td>Responsible Unit</td>
<td>DIS</td>
</tr>
<tr>
<td>Involved Units</td>
<td>DIS</td>
</tr>
<tr>
<td>Authors</td>
<td>Vincenzo Curci, Stephen Kimani, Giuseppe Santucci, Tiziana Catarci</td>
</tr>
</tbody>
</table>

Contact Author
On line Document
Development technology
Java, Swing, JDBC, SQL
Multiplatform item

1.2 DAMA Prototype Description

The prototype, from now on defined DAMA (DAta Mining Application), is a modular Java application that allows the user to interact with data on a relational DB using various Data Mining techniques. At the moment, the application supports, but is not limited to, Association Rules, Metaqueries and Clustering.

The prototype is based on an extensible architecture in order to facilitate the integration of DM technique implementations. DAMA does not implement the DM algorithms, but rather it defines an infrastructure to be adopted to interact with the DM algorithms. The algorithms are implemented by third parties in possibly any language. Clearly, a Java algorithm is better integrated in the GUI than other implementations, which require a more complex development of the interface layer. This interface layer is a set of Java language “interfaces” which must be implemented by the specific DM algorithm to be integrated in DAMA.

DAMA must be considered a prototypical framework to embed and chain DM techniques.

The GUI is structured in a similar manner and represents the core of DAMA. Again there is a general GUI framework which can embed specific GUI implementations of a set of Java interfaces to enable the user interact with the DM algorithms in the most friendly way. In other words, for every DM algorithm, two layers are necessary. The first one is an implementation of the Java abstraction layer to wrap the algorithm engine. The second one is an implementation of the GUI abstraction layer to get the specific interaction environment through which a user activates and controls the algorithm.

The two layers for every DM technique need to be isolated from each other in order to enhance the modularity of the framework. The isolation is realized through the adoption of the classical pattern Model(the Algorithm)-View(The Algorithm GUI)-Controller(The Algorithm GUI in co-operation with the general DAMA GUI and DAMA framework).

1.3 The General DAMA GUI

The general DAMA GUI is the container of the specific DM algorithm GUIs, which are loaded when found and made available to the system. The general GUI shows the list of all the DM techniques available.

Since the GUI acts on data from RDBMS, generic RDBMS data manipulation functions are provided. The functions are basically: DB login and working table selection. There are also functions to start/stop the DM algorithms and save/restore configurations.

DAMA GUI offers also general feedback services to give information on the status of the algorithm, the working DB and informational messages.
A specific DM technique is allocated the tabbed panes. Every technique uses 2 panes: one for configuration and the other for results. On the left hand side is a tree view that allows the user to follow dynamically the execution of a selected DM technique. During the execution, the tree gets filled with information in time. In the following figures, a possible scenario is shown.

![Figure 14. DAMA general GUI layout.](image)

![Figure 22. Empty tree](image)
Figura 3. The tree after user settings
Figura 44. The tree at the end of the discovery process (in this case for association rules).

The main commands can be accessed through the menu or the floating toolbar.

Figura 55. MenuBar and Toolbar.
1.3.1 DM Wizard

To simplify the user interaction, there is also a DM wizard. The following figures show a possible interaction scenario.

Figura 66. Step 1: Select the DM technique

Figura 72. Step 2: Select DB.

Button commands
- Open DB
- Save configuration
- Start the discovery process
- Stop the discovery process
- DM Wizard Activation
- Exit
- Help
Figura 8. Step 3: Select tables, columns and data to process

The last step corresponds to a panel which is specific to every DM technique as described in the rest of this document.
1.4 Association rules

The following figures show some screenshots of the specific GUI for association rules.

Figura 9. Association rules, Setting Panel: the user has selected two hypotheses of rules.

Figura 10. Result Data are shown in tabular format, using colors to enhance rankings.
Figura 11. Alternative histogram representation of result Data: every rule is a parallelepiped. The user can change the point of view with the mouse.
1.5 Clustering

The following figures show some screenshots of the specific GUI for clustering.

Figura 12. Clustering, Setting panel.
Figura 13. Clustering, Result Data: Clusters found.
Figura 14. Alternative graphical representation of clusters: every cluster has a different color. The user can change the point of view with the mouse.
1.6 Metaqueries

The following figures show some screenshots of the specific GUI for metaqueries.

Figura 15.15. Metaqueries, Setting panel.

Figura 16.16. Metaqueries, Join hypotheses are shown through colored boxes on the right of the table boxes.

The representation of metaquery results is identical to that of association rule results.