

An End User Development Model to Augment Usability of Rule Association Mining Systems

Elisa Albergaria, Fernando Mourão, Raquel Prates, Wagner Meira Jr.

Department of Computer Science, Federal University of Minas Gerais, Brazil

{elisa, fhmourao, rprates, meira}@dcc.ufmg.br

Abstract: One of the main challenges to a broader use of association rules data mining systems is their usability. In this paper we propose the End User Development Conceptual Model aimed at enabling the user to customize the interface of rule mining systems and create domain and problem specific queries. To do so, the user must be an expert user, both in the domain and system use (which usually requires knowledge of data mining technical concepts). The goal of the expert user is to create an abstract interface level that will allow a domain expert, with no knowledge of data mining, to use the system in specific problem situations. Thus, expert users can be perceived as co-designers of the system. An initial assessment of the model's usefulness and implementation feasibility was made.

Keywords: End-user development, semiotic engineering, data mining, rule association, customization.

1 Introduction

One of the outcomes of the Internet and the continuous growth of technology access is the creation of large volumes of data. The data mining field started over a decade ago (Webb, 2007) aiming at enabling people to extract useful knowledge from these large volumes of data. Different data mining techniques and methods that support knowledge discovery in distinct situations have been proposed.

Although many data mining (DM) systems have been developed and are in use, most of them still require the user to understand many aspects regarding the techniques or algorithms being used. Thus, one of the current challenges of the field is to decrease the complexity involved in using the systems, improving their usability (Kriegel, 2007) and broadening the use of such systems. In this paper, we present a model that enables at use time the creation of an abstract interface level that will allow domain experts who do not have data mining technical knowledge to make use and benefit from such systems. In order to provide a proof of concept for

Please use the following format when citing this chapter:

Albergaria, E., et al., 2008, in IFIP International Federation for Information Processing, Volume 272; *Human-Computer Interaction Symposium*; Peter Forbrig, Fabio Paternò, Annelise Mark Pejtersen; (Boston: Springer), pp. 161–174.

the model we focus on the usability challenges posed by one of the most popular frequent pattern mining techniques (Hipp, 2000), Association Rules (AR) mining systems.

In a nutshell, association rule (or just rule) mining generates sets of items in the form $X \rightarrow Y$ (Agrawal, 1994). Different interest measures are used to identify potentially relevant rules, such as confidence (conditional probability) or support (frequency of occurrence). For instance, a rule such as *Milk, Cereal* \rightarrow *Coffee* (80.00, 50.00) means that 50% (support) of all purchases in this database include milk, cereal and coffee. It also means that, from all purchases that include milk and cereal, 80% (confidence) also include coffee.

Historically, DM systems have evolved from systems that supported just one step of the knowledge discovery process such as clustering or visualization (1st generation systems) to suites that support several steps of the process (e.g. WEKA (2008)), known as 2nd generation systems (Goldschmidt, 2005) (Piatetsky, 1999). Although these systems are currently very popular, they pose a great challenge to users. This challenge is due to their need for users to understand DM technical aspects, such as which is the best algorithm, the meaning of parameters, as well as their impact on the knowledge discovery. Particularly, in AR data mining systems the user must engage in a complex parameter setting (Kriegel, 2007) that has a direct impact on the quality of the knowledge discovered. Since these concepts are not part of users' domain, learning them represents a high cost and challenge to users (Albergaria, 2006).

Recently, researchers have pointed out the need to create systems that are easier to use (Han, 2007) (Kriegel, 2007). Although there have been proposals in that direction, little has been done towards decreasing the necessary technical knowledge. The works that focus on the issue (Kirkland, 1999) have the disadvantage of limiting the data mining system to a specific problem. Our solution proposes a model that decreases the cost of learning technical concepts without limiting the power of AR mining systems.

In this paper we present the End User Development Conceptual Model (EDeM) that allows users to create domain specific solutions to be added to 2nd generation AR mining systems. In order to create customizations users must also have the required DM technical knowledge. These customizations will then allow users who do not have this technical knowledge to also make use of the system.

In section 2 we present other works aimed at improving usability of DM systems. Then we present theoretical framework that has grounded our work. Section 4 presents the proposed model (EDeM), its components and the preliminary evaluation conducted. Finally, we discuss this work's contribution and next steps in the research.

2 Related Works

Some researchers of the data mining community have recently stated that one of their biggest challenges is to increase usability of DM systems (Kriegel, 2007). The main challenges in using AR mining systems involve having to learn DM technical concepts necessary to define parameters to execute the pattern search. This involves fine tuning an extensive number of parameters, interacting with a large volume of resulting rules, selecting relevant rules and interpreting them (Albergaria, 2006) (Hofmann, 2000) (Kriegel, 2007) (Mei, 2006). Efforts in dealing with one or more of these usability challenges have already been made.

Many authors have focused on supporting users in exploring large volumes of rules and identifying relevant rules by focusing on improving rule visualization or on reducing the number of rules. In an attempt to improve visualization different techniques and novel representations have been proposed (Han, 1996) (Rainsford, 2000) (Wong, 1999). To reduce the number of rules the main approaches have been either to generalize them by use of taxonomy (Domingues, 2005) or by pruning rules that are not interesting according to a specific measure (Srikant, 1997). Other works have focused on supporting users in understanding and interpreting a rule. In this direction visual metaphors (Hofmann, 2000) and semantic annotations have been proposed (Mei, 2006).

All these proposals are advances in improving the usability in rule mining systems. However, they still require users to learn DM technical concepts in order to interact with the system. Some works have as a goal supporting users in learning these concepts. In that direction, expert systems (4th generation systems) have been proposed to support users in decision making during the DM process (Goldschmidt, 2002). As users are guided to understand the process, they gradually learn the concepts.

There have been some works that have aimed at abstracting the required technical DM knowledge from users. To do so, they have created a DM solution that focuses on specific contexts (3rd generation systems). For instance, the Advanced-Detection System (ADS) detects fraud/violative behavior in the Nasdaq Stock Market according to NASD regulation (Kirkland, 1999). These systems are able to offer users DM systems that do not require the learning of technical concepts. However, they can only be used to solve a specific problem. Should the users' problem evolve or change, or the user decide on a new approach, the system would no longer be useful (or a new version of the system would have to be developed).

In this paper we take an end user development approach to propose a solution that allows users themselves to create an abstract level to the system that could be used by other users without having to learn DM concepts. This solution allows for a new abstract interface level, without limiting the systems' applicability.

3 Theoretical Framework

One of the biggest challenges to develop usable systems is that requirements and contexts identified at design time often change at use time (Fischer, 2007). A proposed solution is to include End User Development (EUD) into the systems, that is, to allow users to adapt software to situational and novel uses that emerge at use time. EUD solutions vary from offering users opportunities to customize the systems all the way to including (re)programming components in the system (de Souza, 2005) (de Souza, 2006) (Fischer, 2004) (Fischer, 2007).

Fischer (2004) has defined meta-design as the design that allows users to become system's co-designers and has argued that this approach should be adopted by system developers as an EUD solution. In this paper we take this approach and propose the End User Development Conceptual Model (EDeM) that allows users to create domain specific solutions to be added to 2nd generation AR systems. This model is grounded on the semiotic engineering theory of HCI.

Semiotic engineering theory (de Souza, 2005) perceives an interface as a one-way meta-communication artifact in which the designer conveys to users who the system is aimed at, what problems they may solve with it and how to interact with the system to do so. As the user interacts with the system he understands the range of functions that can be performed, and range of contexts it may be used within. Semiotic engineering argues that the message should also present to him the rationale and design principles that have been followed in creating the system. In EUD systems this message should also communicate to users what parts of the system can be changed and how. As the users become co-designers they also become co-authors of the message being conveyed through the system.

4 End User Development Conceptual Model - EDeM

As we have pointed out, 2nd generation AR systems require users to learn technical concepts in order to engage in pattern finding activities (Albergaria, 2006) (Kriegel, 2007). Although some users are willing to learn them to be able to interact with these systems, others perceive it as just too high of a cost. Thus, the End User Development Conceptual Model (EDeM) presents an architecture model for an EUD module to be built and added to 2nd generation AR systems. The goal of this EUD module is to empower users to create extensions that allow for new interactions possibilities that do not require technical knowledge. To do so it offers users mechanisms to define context and problem specific mining tasks and create a new abstract interaction level in which to activate them directly.

The solution we have proposed was conceived based on a strategy pointed out to us in an interview with a 2nd generation system user that aimed at auditing Government expenses. He commented that he acted as the "miner" for his team, that is, he used the system to identify relevant patterns that could solve their

problem, and then presented to his team what indicators they could use to identify expenses that should be audited.

Based on this strategy, EDeM considers two possible roles a user could take: expert or final user. The expert user (U_{Exp}) is an expert on the domain and understands the technical concepts required to interact with 2nd generation systems. Whereas the final user (U_F) may be a domain expert, but is not willing to endure the cost of learning all these concepts. In EDeM U_{Exp} s act as co-designers creating an abstract interface level for the U_F that is problem specific.

EDeM is composed of three components: the generator that allows the U_{Exp} to create the User Interface Abstract Language; the knowledge base that contains the U_{Exp} 's design rationale for the customization made; and the User Interface Abstract Language (UIAL) that is the resulting interface language with which the U_F will interact. Figure 1 depicts EDeM components and the communication among them. We next explain in detail each of these components and how they relate to each other.

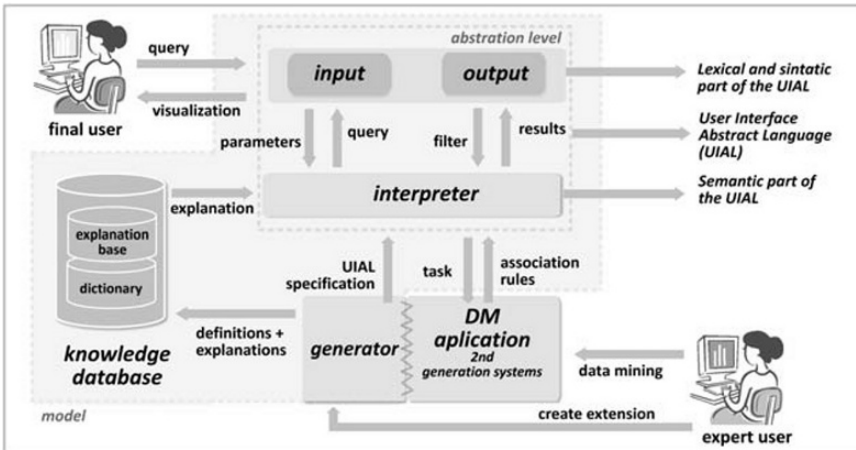


Figure 1 EDeM – End User Development Conceptual Model

4.1 User Interface Abstract Language

The User Interface Abstract Language (UIAL) is created by the U_{Exp} to the U_F . The UIAL comprises one or more problem specific queries that are an abstraction of a DM task. To create a query, U_{Exp} s define what parameters should be considered in the DM process, whether their values are fixed or will be defined by U_F s. U_{Exp} s must also specify how the query, as well as its results, will be presented to U_F s.

In order to quickly illustrate a potential UIAL, suppose that users worked for the agency responsible for auditing Government Purchases. U_F 's responsibilities included identifying if any suppliers had been favored. Hence, the U_{Exp} could create a query as part of the UIAL such as: “During the year of <DEFINE YEAR>, did any supplier win <OP [>]> <DEFINE PERCENTAGE>% of the bids to pur-

chase product <DEFINE PRODUCT>?”. In this query U_F s would assign values to YEAR, PERCENTAGE and PRODUCT, which are all well known concepts in their domain. U_{Exp} s would also define the relation of these values to parameters (e.g. PERCENTAGE would be the value assigned to confidence), as well as set the values to other necessary parameters that are not mentioned at the UIAL level, such as support or AR algorithm selection. Using this query, a U_F could then ask the system “*During the year of 2008, did any supplier win more than 50% of the bids to purchase product toner?*”. The results could be shown to him in a textual form, such “*ETA Inc. was the supplier in 55,19% of the bids to purchase toner.*”

Note that, as any language, the UIAL must have lexical, syntactic and semantic components. The lexical elements comprise the text of the query and of the results, as well as the parameters (and their values) that will be made available to U_F s. The syntax is the possible combinations of the lexical elements at the UIAL. Finally, the semantic part is the behavior of the query, that is, the DM task that is executed once the query is activated.

In EDeM the UIAL is comprised by two subcomponents: the presentation layer and the interpreter. The presentation layer contains all the interface elements U_F s will interact with. The interpreter is responsible for the communication between the UIAL’s presentation layer and the 2nd generation AR system. To do so it translates the specific query submitted by the U_F into a DM task that can be performed by the DM system. Once the DM system generates its result, that is the association rules, it is transmitted to the interpreter which then creates the final output to be presented to the U_F , based on the specifications made by the U_{Exp} .

In the semiotic engineering theory perspective, the UIAL is a meta-communication mainly from the U_{Exp} to the U_F in which the U_{Exp} communicates his views on what are (some) of the problems that are relevant to the U_F that may be solved by the underlying DM system and how to interact with the system. Notice that the U_{Exp} may be the main author of this message, but he is not the only one. The EUD module designers define the set of possible UIAL elements from which the U_{Exp} may be able to choose. Also, the designers of the 2nd generation DM system specify the set of possible mining tasks available to U_{Exp} s to associate to queries. Therefore, the UIAL is actually a message composed by the U_{Exp} and designers of the DM system and its EUD module.

4.2 Generator

The generator is the component in EDeM with which U_{Exp} s interact in order to create the UIAL (presentation layer and interpreter). The generator requires U_{Exp} s to define which elements will be presented to the U_F and how, from which it creates the presentation layer. The interpreter subcomponent is generated based on the definition of the DM task associated to each query at the UIAL.

In order to define the DM task associated to a query U_{Exp} s must decide which parameters are to be considered in the DM process, which ones have their values fixed at design time and which at use time. Although defining DM tasks involves mainly parameter setting the decisions involved in this setting may be complex (Albergaria, 2006) (Kriegel, 2007). EDeM does not intend to support U_{Exp} s in

modeling a problem into a DM task, but rather, making this modeling available to other users who do not have the required technical knowledge to do the modeling themselves.

4.3 Knowledge Base

The purpose of the knowledge base is to allow U_{Exp} to register their rationale for the abstraction they are creating. The knowledge base has two subcomponents: an explanation base and a dictionary. The explanation base stores all clarifications registered by U_{Exp} regarding their decisions. The explanations are classified in two different levels, the ones that are to be made available to U_{FS} , and the others that are more technical and intended to the U_{Exp} himself (or others U_{Exp} s).

For instance, associated to the query “*During the year of <DEFINE YEAR>, did any supplier win <OP [>]> <DEFINE PERCENTAGE>% of the bids to purchase product <DEFINE PRODUCT>?*” the U_{Exp} could associate the following explanation: “*This query allows you to explore the Government Purchase database and identify whether for a specific year and product any suppliers were favored. The underlying hypothesis is that no one supplier should win all the bids. In our experimentation we have found that those who win more than 40% of the times could already be considered candidates of having been favored. So the query considers the minimum PERCENTAGE value of 40%, but you could increase it if you would like.*”. This explanation would allow U_{FS} to understand the (intended) meaning of the query, as well as why there is a minimum value defined for PERCENTAGE. Thus, from a semiotic engineering perspective, this explanation is crucial to improve the quality of the U_{Exp} to U_{F} (designer to user) communication through the system.

An example of a more technical explanation would be to clarify why a specific fixed value has been chosen for a parameter (e.g. support = 0.27). For instance, “*This value was defined by testing different values in the Government Purchase Database and was considered a relevant value because ...*”. Notice that this explanation registers the rationale for a decision regarding the modeling of the problem into a DM task. Since support is not shown at the UIAL it is not intended to U_{FS} but may be essential to document the *ad hoc* value chosen.

The other subcomponent of the knowledge base is the dictionary. The dictionary registers which elements of the underlying 2nd generation system interface language will be used in the UIAL, and how they will be translated to the UIAL. For instance, <PERCENTAGE> that appeared in the query shown above would have an entry in the dictionary such as the one depicted in Figure 2. The lexicon is defined by the U_{Exp} and contains its description in the UIAL (may have a variable, as well as the fixed text). The semantics is the element it represents in the 2nd generation interface. Finally, the U_{Exp} may also enter an explanation related to the element.

The dictionary has two main functions in the model. The first one is to lead the U_{Exp} to think about what elements of the DM system he believes should be part of the UIAL, as well as how they should be represented. The other is to support the U_{Exp} in maintaining the consistency among the elements that are shown in the in-

put and output of UIAL, since these elements may be defined at distinct times in its creation process.

<p>Lexicon: <PERCENTAGE> % of the bids</p> <p>Semantics: Confidence</p> <p>Explanation: In this query confidence represents...</p>

Figure 2 Example of dictionary entrance.

Although the knowledge base component is not required to create or execute queries, it could have a major impact on how people use them. Semiotic engineering (de Souza, 2005) perceives it as an essential component. Inasmuch as it requires U_{ExpS} to add metalinguistic elements (i.e. elements that explain other elements/aspects) to the UIAL. These metalinguistic elements are a privileged part of the U_{Exp} to U_F communication, since they allow U_{ExpS} to send a direct message about their intentions or decisions. Also the dictionary acts in the model as an epistemic tool (i.e. a tool that increases the problem-solver's understanding of the problem (de Souza, 2005)) to U_{ExpS} for it leads them to reflect about decisions being made regarding the UIAL.

5 Preliminary Evaluations

The solution proposed by EDeM was based on two main premises: (1) that expert users could use it to generate useful queries to final users who did not have DM technical knowledge; (2) that an EUD module based on its proposed architecture could be built for an existing 2nd generation AR system. Thus, our first assessment effort focused on collecting indicators about these two aspects: usefulness of the model and implementation feasibility.

5.1 Usefulness

Assessing potential usefulness of EDeM involved evaluating whether U_{ExpS} could actually create an abstract level that was domain dependent and relevant to U_{FS} . An initial evaluation in that direction was done using scenarios (Carroll, 2000) and was carried out in two steps. The first one involved taking an existing DM task created in a 2nd generation DM system and defining queries that could potentially be useful to U_{FS} . The next step was to verify whether other U_{ExpS} would be able to create abstractions that were relevant queries in different domains.

The first step was done by taking an existing problem that had been modeled as a DM task and creating an abstract level that described how a U_F would input the query and get the responses to it. The problem chosen was that of a real client of the DM research group at UFMG (Tamandua, 2008) the State Government Auditing Department, which is responsible for auditing public state agencies' bidding

process and purchases. In 2006, rule mining tasks using a 2nd generation DM system, the Anteatr (Guedes, 2006), had been defined in order to identify potential frauds in the purchasing database. The focus was on supplier favoring, (i.e. deciding in favor of a supplier by means not stated in the law). Originally, the mining task had been defined by experts of the DM group. Using this work as foundation, a scenario describing in detail how an abstract level based on the model could be created to give access to U_{FS} to the issues they were interested in was created. (Examples used in the previous section are excerpts of this scenario.) The scenario showed how the model could support the creation of queries that would be interesting to intended U_{FS} . Queries considered interesting were those that conveyed results described in the original final report.

The next step was to verify if other U_{ExpS} would be able to create relevant queries for different domains. This step of the evaluation was conducted as part of a class project for the DM course at the undergraduate level. In the project developed by the DM class (2nd semester, 2007), students had to model a real problem as a rule association mining task, and also create a model of the abstract level that would be presented to a final user. In other words, define questions final users would be interested in asking, as well as possible domain specific responses that could be (automatically) generated from the resulting association rules. Thus, projects required students to interact with a real user to define real problems to be modeled, as well as to have access to the database containing the necessary information.

Projects were usually developed in groups of 2 students. Out of 12 projects that were handed in, 8 (assessed as having achieved initial goals) were considered for the model evaluation¹. Projects were done in 3 different domains: quality of test questions on the University's admission test (6), building electricity expenditure monitoring (1), and crime rates in the city (1).

All eight groups were able to create a successful abstract level (input and output). By successful we mean an abstraction that was problem specific and did not depend on understanding the underlying technical concepts. One group went beyond what was requested and actually implemented the query on top of Weka (2008) an open source DM system. In their project, students were also required to explain their proposed queries, as well as their modeling of the problem as a DM task. One interesting outcome was that even though the dictionary was not presented to them, most students included a mapping between representations used in the queries and elements of the DM system interface.

Both steps of the evaluation generated positive indicators that U_{ExpS} could create useful abstract level queries that would be relevant to U_{FS} . Thus, the next step was to evaluate the feasibility of implementing a EUD module using EDem that could be added to an existing 2nd generation DM system.

¹ After the semester was over students were asked if they authorized their work to be used for evaluating the model.

5.2 Implementation Feasibility

In order to evaluate whether an EUD module based on EDeM could be implemented involved actually implementing the model and analyzing the cost of adding it to a 2nd generation DM system. The system chosen was the Anteater (Guedes, 2006) (Tamandua, 2008), developed at Computer Science Department at UFMG, and to which architecture and code we had access to. Anteater is a DM platform that aims at providing scalable and efficient DM services (Guedes, 2006). The Anteater offers rule mining techniques to its users, but new frequent pattern finding techniques are also being developed to be added to the system. It has been used in projects with the Brazilian Government in different domains such as auditing, health and public safety.

The prototype to be added to the Anteater system was designed based on EDeM. Users have a choice of interfaces: query interface, extension creation interface, or original Anteater interface. In order to create the UIAL, U_{Exp} s must define the query to be shown to U_F s; decide the values of parameters to be considered in the AR mining task; and specify aspects of the final presentation of the resulting association rules. The query input interface is always textual and the U_{Exp} defines the text to appear and which attributes (previously selected from the database) the U_F will be able to specify values for. When defining the query text to be presented, the U_{Exp} may also enter an explanation for it to be made available to the U_F . Figure 3 (A) and (B) show the screenshot of the extension creating interface for defining input query interface and its result at the UIAL, respectively.

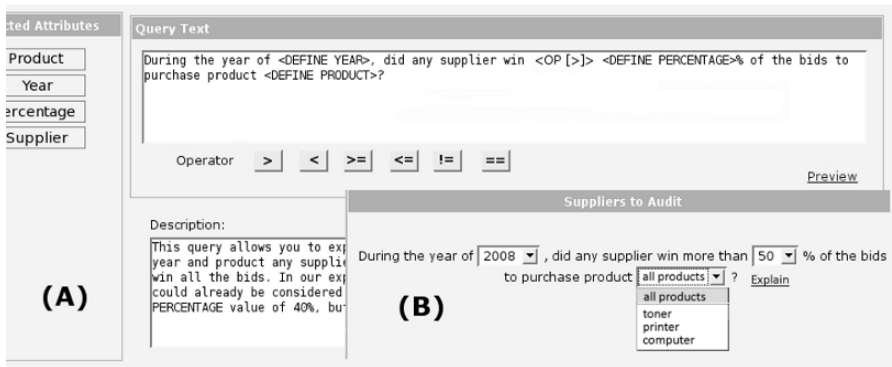


Figure 3 (A) Extension creating interface for defining query input; (B) UIAL input query interface.

The U_{Exp} also defines how the association rules should be translated to appear at the UIAL (Figure 4(A)) to the U_F (Figure 4(B)) and by which parameter it should be ordered (filters for attributes defined in the query are available to U_F s).

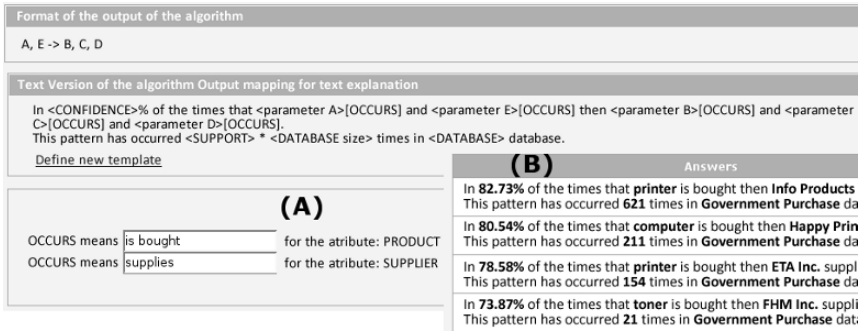


Figure 4 (A) Extension creating interface for defining query output; (B) UIAL results interface.

The prototype is being implemented (in Java and AJAX) as a proof of concept of EDeM to be added on to the Anteater system (implemented in Java and JSF). The Anteater has a distributed architecture based on major points of the knowledge discovery problem. Thus it contains 3 servers (database, data mining, and visualization) that contain a specific set of functionalities to be offered to users as services. All interaction during the processing of a user request happens over the Web, based on a user interface that controls the access to individual services. These services are in a 4th component, the application server.

EDeM required one more server to be implemented to contain its components, namely the generator, interpreter and the knowledge base. The interface module that controls the interaction with both expert and final users was added to Anteater’s application server. In this case, it made sense to keep together all the user-system interaction modules.

Even though the prototype is not fully functional yet, it has already shown that it is possible to create an EUD module to the Anteater. Most of the extension interfaces are already implemented and most of the UIAL generation works. Although the communication with the Anteater has not been completed yet, it has already been proven possible. The UIAL input query interface generates a list of parameters and their assigned values that are necessary to execute an AR mining task. Also the high level output is created from the association rule structure generated by Anteater (only textual form is implemented at this point, but table format and graphical presentations have been planned as alternatives). The main reason why the communication among the prototype and Anteater has not been completed is because rather than hard coding it into the system, we are investigating the possibility of using a markup language (namely, PMML) to transfer the information between them. This would make it easier to evaluate the prototype with other 2nd generation rule mining systems in the future.

6 Discussion and Next Steps

This paper has presented the End User Development Conceptual Model (EDeM) that allows for a U_{Exp} to act as a co-designer of a 2nd generation AR system. The U_{Exp} is empowered to create a novel interaction abstract level that is problem and context dependent and that enables U_{Fs} (who do not have knowledge of DM technical concepts) to interact with DM systems and benefit from them. Differently from 3rd generation DM systems, this solution does not limit the system's applicability. U_{ExpS} can easily change or add new queries to the UIAL.

EDeM is founded on semiotic engineering theory which argues that designing is a communicative act. Thus, the model supports U_{ExpS} in creating their messages to U_{Fs} by providing them with a knowledge base (KB). The KB allows U_{ExpS} to register explanations and intentions regarding queries created, potentially improving quality of the U_{Exp} to U_{F} communication. It is also possible for U_{ExpS} to enter their rationale for the task modeling into the KB, supporting their reflection on values chosen, as well as documentation. In this direction, the dictionary also supports U_{ExpS} reflection on which elements of the underlying DM system to represent in the UIAL and how. Moreover, it helps maintain consistency of how DM system elements are represented in different steps of the UIAL creation process.

Different aspects of EUD systems may be analyzed to provide indicators about them. Fischer (2004) proposes that EUD languages should be considered according to their cost of learning to the user and scope. The EUD language proposed by EDeM has a low learning cost to U_{ExpS} (who already know the 2nd generation DM system's interface language) and a low scope (since it is only useful for creating extensions for DM systems).

In that same direction de Souza and Barbosa (2006), in a semiotic analysis, show that EUD involve distinct signification systems for users and systems. Systems have a symbol manipulation perspective and should be considered in lexical, syntactic and semantic terms. Users have a communicative perspective and manipulate expressions, contents and intents. They point out that system and user perspectives do not have a one-to-one mapping. Thus, it is interesting to analyze the extension provided by EDeM according to these perspectives.

For the DM systems the extensions are meaning preserving, inasmuch as they can only manipulate lexical and syntactic levels (creating queries that combine parameters and their values). However, the systems cannot be semantically extended that is, no new functionalities may be added to them. In terms of U_{ExpS} communicative perspective the extensions are intent-preserving. As U_{ExpS} create a query they add a new expression to the system, associated to a specific content. Since the content and intent (what to achieve with the mining task) were already possible in DM systems, adding the queries has a rhetorical effect in the U_{ExpS} perspective. In analyzing the model, we must also consider its effect for U_{Fs} . They may not be directly involved in creating extensions but are the intended consumers for them. In their perspective extensions are full-fledged design that offers them new intent, content and expression, since prior to them they had no access to the system.

It is easy to see that creating extensions has a low cost for U_{ExpS} , but a very high benefit for U_{FS} . As U_{FS} understand the exploring possibilities of DM systems they may become willing to learn the technical concepts involved. In that direction EDeM may also be useful, since the explanations provided by the U_{ExpS} may support U_{FS} in understanding the DM tasks associated to the queries, and gradually learning the technical concepts.

Although EDeM offers U_{ExpS} support to create the query and reflect about it, as well as register their rationale, it is not able to guarantee the quality of the extension created. The final extension may have a low quality for a number of reasons varying from poor choices of elements for the UIAL to a bad modeling of the domain problem into a mining task. This problem is not introduced by EDeM, since 2nd generation AR systems are also not able to provide any indicators on the quality DM tasks created by users.

The development of the prototype is currently being completed. Once it is fully functional and coupled to the Anteater, evaluation involving users (U_{ExpS} and U_{FS}) will be performed. This assessment will provide us with indicators on the prototype itself, as well as on the underlying End User Development Conceptual Model. Although association rules mining is one of the most popular frequent pattern technique, there are a number of other relevant ones. Thus, we intend to investigate the model's applicability to other frequent pattern techniques and data mining methodologies. It would also be interesting to evaluate the cost of coupling the prototype to other 2nd generation AR mining systems, (e.g. Weka (2008)).

Acknowledgments The authors thank CAPES, CNPq, Fapemig and Finep for their support to their research and to the Anteater project.

References

- Agrawal, R., and Srikant, R. Fast Algorithms for Mining Association Rules. In *Proc. of the 20th Int'l Conference on Very Large Databases*, Santiago, Chile, September (1994).
- Albergaria, E., Prates, R., Almir, F., Rocha, L. and Meira Jr., W. Characterizing interaction challenges in data mining systems. *Procs of IHC*, 40-49, (2006) (*In Portuguese*).
- Carroll, J. M. *Making Use: Scenario-Based Design of HCI*. MIT Press (2000).
- de Souza, C. S. *The semiotic engineering of HCI*. MIT Press, Cambridge (2005).
- de Souza, C. , Barbosa, S. A semiotic framing for end-user development. In: Lieberman, H., Paternò, F.; Wulf, V.. (Org.). *End User Development*. Springer, 401-426 (2006).
- Domingues, M. and Rezende, S. Using taxonomies to facilitate the analysis of the association rules. *Procs of the 2nd Int'l Workshop on Knowledge Discovery and Ontologies*, 59-66 (2005).
- Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A. G., and Mehandjiev, N. Meta-Design: A Manifesto for End-User Development. *Communications of the ACM*, 47, 9, 33-37 (2004).
- Fischer, G. Meta-Design: Expanding Boundaries and Redistributing Control in Design. *Proceedings of the Interact'2007 Conference*, Rio de Janeiro, Brazil, September, (2007).
- Goldschmidt, R.; Passos, E.; Vellasco, M. An Action Plan Definition Assistant in KDD Process. *Procs of the 2nd International Conf. on Artificial Intelligence and Applications*, (2002).

- Goldschmidt, R. *Data Mining – A Practical Guide*. Editora Campos (2005). (In Portuguese)
- Guedes Neto, D., Meira Jr., W., Ferreira, R. A. C. Anteaater: A Service-Oriented Architecture for High-Performance Data Mining. *IEEE Internet computing*, 10, 36-43 (2006).
- Han, J., Y. Fu, W. Wang, J. Chiang, W. Gong, K. Koperski, D. Li, Y. Lu, A. Rajan, N. Stefanovic, B. Xia, and O. R. Zaïane. DBMiner: a system for mining knowledge in large relational databases. In *Proceedings of KDD.*, 250-255 (1996).
- Han, J., Cheng, H., Xin, D., and Yan, X. Frequent pattern mining: current status and future directions. *Data Min. Knowl. Discov.* 15, 1, 55-86 (2007).
- Hipp, U Gütntzer, G Nakhaeizadeh.. Algorithms for Association Rule Mining - A General Survey and Comparison. *ACM SIGKDD Explorations Newsletter*, 58-64 (2000).
- Hofmann, H. Siebes, A. and Wilhelm, A. Visualizing Association Rules with Interactive Mosaic Plots, *Proc. Int'l Conf. Knowledge Discovery and Data Mining (KDD)*. 227-235, (2000).
- Kirkland, J.D.; Senator, T. E; Hayden, J. J.; Dybala, T.; Goldberg H. G.; Shyr, P.. "The NASD Regulation Advanced-Detection System," *AI Magazine*, 20, 1, pp. 55–67 (1999).
- Kriegel, H., Borgwardt, K. M., Kröger, P., Pryakhin, A., Schubert, M., and Zimek, A. Future trends in data mining. *Data Min. Knowl. Discov.* 15, 1 (Aug.), 87-97 (2007).
- Mei, Q.,Xin, D., Cheng, H. , Han, J., Zhai, C. Generating Semantic Annotations for Frequent Patterns with Context Analysis. *Proceedings of KDD*, 337-346 (2006).
- Piatetsky-Shapiro, G. The data-mining industry coming of age. *IEEE IS*, 14, 6, 32–34 (1999).
- Rainsford, C. and Roddick, J. Visualization of temporal interval association rules. In *Procs. of the 2nd. Int'l Conf. on Intelligent Data Engineering and Automated Learning*, 91-96 (2000).
- Srikant, R. and Agrawal, R.. Mining generalized association rules. *Future Generation Computer Systems* 13 (2–3), 161–180 (1997).
- Tamandua (Anteaater) <http://tamandua.speed.dcc.ufmg.br>. Accessed in February (2008).
- Webb, G. I. Editorial. *Data Min. Knowl. Discov.* 15, 1 (Aug. 2007), 1-2.
- Weka, <http://www.cs.waikato.ac.nz/ml/weka/>. Accessed in February (2008).
- Wong, P. C., Whitney, P. and Thomas, J.. Visualizing Association Rules for Text Mining. In *Procs of IEEE Information Visualization*, IEEE CS Press (1999).