Decision Support by Learning-On-Demand

Klaus P. Jantke[1], Martin Memmel[2], Oleg Rostanin[1], Bernhard Thalheim[3], Bernd Tschiedel[3]

[1] German Research Center for Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3, D-66123 Saarbrücken
[2] Computer Science Institute, University of Kaiserslautern,
PostBox 3049, D-67653 Kaiserslautern
[3] Computer Science Institute, Brandenburg University of Technology at Cottbus,
PostBox 101344, D-03013 Cottbus
jantke,rostanin@dfki.de memmel@informatik.uni-kl.de
tschied,thalheim@informatik.tu-cottbus.de

Abstract. Decision-support systems provide a large functionality supporting the decision making process. They allow to draw conclusions from data provided by users. In order to be applied they require however that the user is knowledgeable and knows how to interpret and apply the solutions provided. Often people need, however, an additional insight into the foundation of decision support systems. The Data Mining Tutor (DaMiT) has been developed in order to overcome this problem. The learning process is based on data provided by DaMiT and data provided by users. User learn with the DaMiT system what kind of decisions can be made, what kind of conclusions might be drawn, and what kind of decisions should not be drawn due to the data quality.

1 Introduction

Decision support systems (DSS) are classically designed to serve the management level of organizations. They help managers make decisions that are semi-structured, unique, or rapidly changing, and are not easily specified in advance. Therefore, DSS have to be responsive enough to run several times in a day in order to correspond to changing conditions. While DSS use internal information from databases, they often bring in information from external sources, such as current stock prices or web databases. It is claimed that users of DSS can work with them explicitly with a variety of models to analyze them directly; these systems explicitly include user-friendly software, are interactive and provide a service to change assumptions and to include new data. DSS offer users flexibility, adaptability, and a quick response. They allow users to initiate and to control the input and output. They operate with little or no assistance from professional programmers. DSS use sophisticated analysis and modeling tools.

This large list of requirements cannot satisfy all cases appearing in practice although DSS tend to be very large systems with a very large set of functions. At the same time problems appear at run-time. In order to solve such problems and to map the problems to the solutions provided by DSS, users must be educated very well and must understand the functions provided by the system. Therefore, users are usually overwhelmed by the system, they are not able to understand what the result of an analysis is and what is maybe misunderstood[1]. We call this gap the knowledge gap.

Instead of using huge systems with results which might be not interpretable or are not understood and with results which are partially incorrect, users should know what they are doing. Users of DSS are application engineers working in the area where the data have been collected. The DaMiT system educates users to such extent that they can evaluate whether the results of decision support systems are generating correct results and whether results will lead to correct conclusions. In this case, our approach leads to better analysis results and higher confidence into the anaylsis:
If in practice a problem occurs which might lead to decisions or to changes in strategy or practice, engineers should be supported by a system that allows them to understand the associations in the data, to understand the algorithmic analysis of the data and to understand what kind of conclusions may be drawn out of the analysis.
We call this approach the decision-learning approach.

In a large project integrating research groups in ten German universities and application groups in half-dozen German software companies we developed a system that supports decision learning

- by learning basic and advanced topics on data mining on demand,
- by applying data mining algorithms to test data for training of users, and
- by finally applying these data mining algorithms to data of the application engineer.

Using our system the user becomes knowledgeable and will be able to correctly interpret the results of data mining in his/her data. Based on the correct interpretation of the results generated by a data mining algorithm on the users

[1] In the Cottbus Innovationskolleg data have been collected on the open coal mining area. The size of these data reached almost 15 TB. The engineers wanted, for instance, to know whether watering mines will influence the local climate. The results generated by one of the most powerful decision support system led to the conclusion that watering will lead to an average temperature decrease by half degree centigrade per year over a longer period. Therefore, a proposal has been raised not to water the coal mines but rather to green these mines. This proposal can only be satisfied with huge investments. At the same time it became clear that something must be wrong in the analysis. The environmental scientists were not able to clarify which error had been made and how to correct this strange result. After careful correction of errors in the data, after some research on statistical functions [STY99] could clarify that the only change on the local climate will be a drift away from the current continental climate (hot summers, cold winters, low rain rates) to a climate also observed in other areas of Eastern Germany.

data the user can draw the right conclusions and knows the deficiencies and problems within the solution.

A large number of tutoring systems are available in the market. Most of these tutoring systems are based on the paradigms of instructional learning. The learner follows a sequence of steps and may proceed further depending on the success of exercise solution. We did not find any system that is able to integrate users data and to interactively work with algorithms. The last requirement is far too high to be satisfied in general. We can show, however, that such systems can be built by restricting the knowledge area and the functionality that is provided.

The size of the paper restricts the topics to be tackled in this paper. We thus concentrate on two topics: A general introduction into the DaMiT system and a discussion of scenarios of systems utilization. We do not want to provide a general introduction into the data mining research neither give an insight into data mining algorithms nor into research on general decision support systems. Our main aim in this paper is to show how user can be supported to understand the impact of solutions for their decision problems.

2 Intelligent System Support for Decision Making

Decision making requires to be knowledgeable on the outcome of solutions and to forecast the impact of the decision. This task is very difficult to solve. A simpler task is the support of the generation of right solutions for the data provided. This task can be solved if the user knows how algorithms used for solution generation are computing on the data, which problems might create partially incorrect or missing data and what is the statistical support of the solution generated.

The DaMiT system targets on the last problem. It provides an intelligent way to learn data mining theory, to learn to handle data by data mining algorithms and to interpret results of the algorithms in data ming.

2.1 Data Mining for Decision Support

Many authors understand data mining as the engineering discipline of digging for the golden nuggets of knowledge hiding in knowledge sources (cf. [HKMW01]). This is, for sure, the wrong perspective. It is misleading in the sense of suggesting that a user just needs to get access to the right tools and then, in wielding the tools, gets through to the nuggets.

But even in the successful case, the nuggets are not *found*, they are *generated*. Data Mining is a version (or a subprocedure) of knowledge discovery (cf. [Br02], and knowledge discovery deals more with building the knowledge than with finding it. This general approach is nicely expressed in Albert Einstein's letter to Karl Popper in 1935 on the occasion of publishing [Pop35].

In other words, data mining is a creative process of building models that turn out to be useful in some sense. Perhaps, the insights derived from such models can be economically exploited, or they lead to scientific and engineering ideas towards innovative results. When models have been generated and is sitting on your computer screen, it is rather difficult to decide whether it is a *useful* one or not. There is an overwhelming collection of data mining approaches and techniques (cf. [KlZy02]), and you can never be sure in advance which one will be the best and most successful. Data mining ist very much both an art and a science.

Consequently, data mining is a typical area that calls for human-computer interaction or, even more explicit, human-computer co-operation. What we do need is a co-operation in which both partners are bringing in their respective strength.

Taking this consideration seriously, data mining turns out to be a core discipline of artificial intelligence (AI). In the authors' opinion, AI mainly deals with the development of computers from tools to assistants. Information and communication technology, in general, does reach an ever increasing number of users. It is unrealistic to expect all those users to become IT-specialists. The opposite should take place. Users should not be expected to adapt to computer systems, but computer systems should be flexible, adaptive and even proactive to behave like proper assistants to the human users.

What we have called above the decision-learning approach does call very explicitly for computer assistance. The system DaMiT underlying the present publication is such an interactive assistant. In co-operation with the DaMiT system a user can *study the science and experience the art of data mining.* This dichotomy is worth a closer look.

When you are really dealing with data mining problems, you try to build a model (a decision tree, for example, to classify your costumers in those who will react to your mailing action planned and those who will not) on the data you have. Your computer, especially the data mining software on your computer, does suggest complex models. In case you are not yet satisfied with the computer's proposal and, therefore, provide other data or tune certain parameters, your computer responds with just another, usually more complex, model. In playing the game, you are facing a flood of models which are, in fact, a result of your creative co-operation with your machine. It is not easy to distinguish whether you are more experiencing the art or practicing the science of data mining. It is always worth to rethink what has been done so far. Sometimes, an inspection of the underlying assumptions, of properties of the algorithms invoked and the like might help. Your activities towards building a decision support model might benefit from some dovetailing with learning activities: learning on demand.

In complicated decision support situations, a straightforward model construction will rarely work. We have to reconsider decision support.

Furthermore, it is worth to explicate that the DaMiT system goes far beyond the limits of conventional e-learning systems. Experiencing the art of data mining does require excessive opportunities for *doing* data mining. Thus, learning by doing plays a central role in the DaMiT approach. On the one hand, the DaMiT systems appears as an e-learning system with a large variety of proper domain functionality offered. On the other hand, it may be seen as a portal to a variety of data mining functionalities that assists the user with an unusually intense learning support. With DaMiT, your decision support activity is supported by learning on demand.

2.2 Learning Elements and Learning Units

Instructive teaching is based on modules. Interactive or cooperative learning must be based on learning objects of lower granularity. Moreover, learning objects must be adaptable to the learner based on the profile of the learner, to the language of the learner, to the content currently available, and to the environment of the client. This kind of adaption cannot be supported if learning objects are of the granularity of modules. Therefore, DaMiT is based on a separation and decomposition of learning objects into learning elements that are of lowest granularity and the construction of learning units:

- *Learning units* are the main objects to be transferred to the learner. We distinguish between *provided* and *derived* learning units. the first are provided by the author. The second may be provided by an author on the cut-and-paste interface. In most cases they are derived on the basis of *filters*. Filters are for instance: the repetition filter (used for quick repetition of units until a certain point), the definitions filter (used to cut only parts which are not useful for definitions), the examples filter (used for assembling examples), the quick reminder filter (used for generation of prefixes that will remind the user at a given point of utilization of the unit on main elements of the provided learning unit, e.g., assume that a user resumes the execution of a unit at a point s then the quick reminder filter applicable in this case generates as a prefix those elements which are necessary to understand before continuing at a given point (quickReminder($e_1; ....e_{s-1}$); $e_s; e_{s+1}, ...e_k$). Other filters may be defined in a similar fashion, e.g. proposition-and-theorems filter [TAT03].
- *Learning elements* are components of learning units. They can be associated to learning units as displayed in Figure 1. In this case the learning unit is expressed by a sequence of elements $e_1; e_2; ...; e_k$.
  Learning elements are associated to other learning elements, e.g. by prerequisites, precedence, extends associations. They are characterized through meta-data similar to the learning objects (LOM) standard [LOM02] such as authors, restrictions and requirements for utilization of leaning elements, educational characteristics, copyright. In general, a learning unit can be specified by an algebraic expression in the website specification language SiteLang discussed below.
- *Derived learning units* can be defined by any expression on learning elements. Particularly, some elements are defined by subexpressions of the main learning units.

2.3 DaMiT Systems Architecture

The DaMiT system separates authoring tools and the delivery system. The solution generalizes data warehouse architectures and is based on an input engine, an content management system, and and output engine:

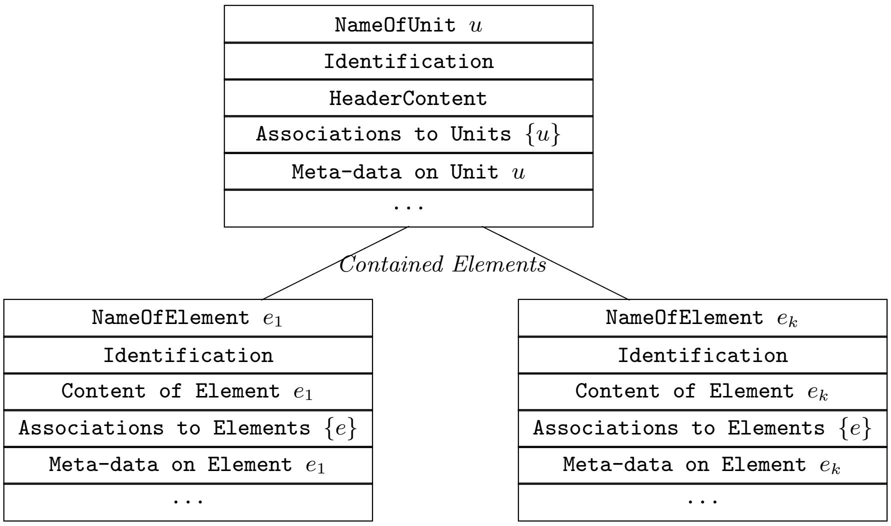

Fig. 1. Learning Units and Learning Elements

DaMiT Input System: The DaMiT input system supports authoring for the delivery of content, supports input of content from foreign databases and supports integration of content provided by other systems. The system uses a variety of learning objects:

Lectures are provided on the basis of LaTex texts. The learning units and elements are characterized

- by direct characterization such as learning level, difficulty, learning style, knowledge level of the learner, prerequisites, goals for learning for the learner, language, version,
- by association among the learning objects and classification of the objects based on the DaMiT topic and ontology maps and the glossary,
- by meta-information such as general learning object information, didactic scenarios into which the learning object might be integrated, pedagogical information,
- by specific information on the learning object, e.g. identification information, authors, usage information, status, source, references and links,
- by utilization information such as accessibility, payment restrictions, security information, rights, technical restrictions and requirements for utilization.

The learning objects characterization extends the LOM standard.

Foreign databases can be docked to the DaMiT engine on the basis of interfaces and integration tools. The interface specification is based on a number of supported formats such as ARFF (Attribute-Relation File Format) used in the WeKa data mining system, CSV (Comma Separated Values) often used in spreadsheet programs, DAT (Tabulator separated data) used in SPSS, DATA, NAMES, ALL (used in C4.5 and the MLC++ library, PMML (Predictive Model Markup Language) used for exchange of XML data and additionally formatted on the basis of JXM data.

Foreign content is integrated on the basis of XML exchange facilities.

Acquisition tools: The DaMiT system is developing dynamically based on the work of users with the system. The system is used in the Data Mining Cup [DMC] which is a competition of users based on the facilities provided by the DaMiT system. The results of the competition are integrated into the DaMiT system and enhance the learning space.

Gate subsystem: In order to support the storage engine, all data are transferred to DB2 data. The gate subsystem transfers the data, the learning objects to the content management system.

DaMiT content management system: The DaMiT contents are stored in the relational database management system DB2. DB2 provides tools for handling deeply structured data and for extracting data from XML documents and for injecting data into XML documents.

Storage is based on the database structure that models all meta-data of learning objects, on the topic and ontology maps and on the functionality provided by the DaMiT engine to the user.

User management is an integral part of the system. Users have their profile, their style of usage and their history of using the learning units in the system. To support automatic adaptation of the system to the user, information on the user, the profile, the learning history and the user's payment is stored.

Data extractors, database mining toolbox: The DaMiT system is automatically adapting to the user and the specific needs.

Learning units generator: This component of the system generates the learning unit from the learning elements based on the specification of the unit, based on the specific profile of the user, based on the learning environment of the user and based on the technical environment currently in use.

XML document generator: A learning unit is represented by a suite of XML documents. The generation of the suite of XML documents is based on the onion approach [ThD01] used for website generation. It provides XSLT-based facilities [KlM03] for generation of the environment, e.g. the navigation, the search and display facilities. This approach is based filtering and stepwise extension of the learning units depending on the learner, the list of tasks under consideration, the learning history, the computational environment of the learner, and the payment profile [TAT03].

Play-out system: The play-out sub-system records the actions of the user and coordinates the access of different users. It is also used for the display of the DaMiT system content in the same outfit.

Payment engine: Some of the lectures can only be used if the learner pays for their use. Therefore, the DaMiT system encloses a payment management component. If the learner pays for a unit or for the data then these data are delivered to the workspace of the user

Workspace engine: Each user owns his workspace for the data he/she wants to use and owns, for the learning units, the results of the learning steps, and for the exercises the user is completing.

3 Decision Support Situations

It is well known to everybody but not supported by most tutoring systems that learners have very different skills, very different background knowledge, very different approaches to learning and want to use learning systems whenever this is really necessary. Such automatic adaptation is difficult to achieve.

In [Cau00] it has been shown how knowledge-imparting web documents can automatically be composed based on the information

- of prerequisites of learning elements,
- of targets for learning,
- of the knowledge and skills profile of learners and
- the environment of the learner.

However, this approach has left some open questions, e.g. concept structuring, document chapter structure, document generation on the basis of sparse content dependency structures. The content modeling in DaMiT is based on ideas similar to [Cau00], while the content generation is realized by means of a user-determined lesson structure. We developed an architecture of learning elements and learning units and focused on supporting scenario-driven learning in a flexible way based on the website specification language SiteLang. It is also worth considering how the DaMiT solution can be enhanced by didactics-driven, automatic aggregation of content objects analogously to [Cau00].

3.1 The Website Specification Language SiteLang

SiteLang [ThD01] is based on a theory of media objects [ScT00b], on entity-relationship models [Tha00], and on the integration of media objects into story boards [ScT00b]. Interaction description can be based on notions developed in linguistics and in movie business. The *story* of interaction with the information system is the intrigue or plot of a narrative work or an account of events. Stories can be played in different *scenarios*. The *story space* consists of a well-integrated set of scenarios. We restrict ourselves to stories which can be handled by casual web-users and which do not need specific education before the site can be used.

A *scenario* is a run through a system by a cooperating set of actors. A scenario is composed of scenes. Each scene belongs to a general activity. Each scenario has its history which can be accumulated and used for the representation of the *history* of the part of the scenario visited so far. This history is used for the *escort information* [Tha00] attached to the media objects of the scenes of the scenario. The *scene* of scenario is associated with a consistent *expression of dialogue steps*.

Dialogue scenes are represented by frameboxes. We represent dialogue steps by ellipses. The transitions among dialogue steps are represented by curves.

3.2 Story Boarding For Decision Support

The story board of the DaMiT system has been been developed by applying the website specification language sketched above. The story space can be represented as displayed in the following graph in Figure 2. The login scene clarifies

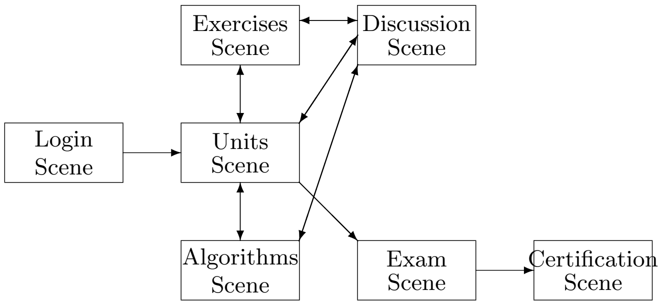

Fig. 2. The Story Space of the DaMiT System

– which learner (identified by name or anonymous) with
– which profile (knowledge and skills level, history of use, group of users)
– and which portfolio (environment context, payment and tasks to be performed)

is going to use the system.

3.3 A Typical Scenario

Let us now use the DaMiT system step by step for decision support. In this case the following scenario is applicable. We run the system and discuss the scenes and dialogue steps in these scenes.

1. The practitioner has read the surveys on available algorithms, their application areas, their limitations and the requirements to data which can be analyzed by the algorithms. After that the QuDA component developed by [GY02] has been chosen. A number of algorithms within this DaMiT component is applicable.
2. The practitioner learns first those units which are necessary to be understood for the application of the QuDA algorithms and for the interpretation of results obtained by applying one of the algorithms to the practitioners data.
3. Now the practitioner enters the algorithms application scene. He/she chooses first the algorithm to be applied. It is possible to obtain a quick reminder. Furthermore, the learner may first look over demonstrations and illustrations for the chosen algorithm.

4. Now the practitioner selects the data for the algorithm and configures the data in his/her system for utilization by the algorithm.
5. Furthermore, the practitioner executes the algorithm and obtains results.
6. Finally, the results are explained to the practitioner. He/she may now continue by selecting another method or algorithm and obtain additional insight into the data to be analyzed.

As data mining is a creative process of interactivly generating more or less complex models, there is no a priori criterion of success. Several runs through the steps above migth be necessary.
The representation of learning steps is developed in detail based on the SiteLang framework. After the learner has obtained the information on available learning units and has collected the program for learning he may enter the algorithms scene after completing the units scene as shown in Figure 2. The algorithms scene allows the learner to use the hand on his/her hands and to understand the impact of the algorithm application to the data. The algorithms scene is displayed in Figure 3. The implementation of the dialogue step which is used

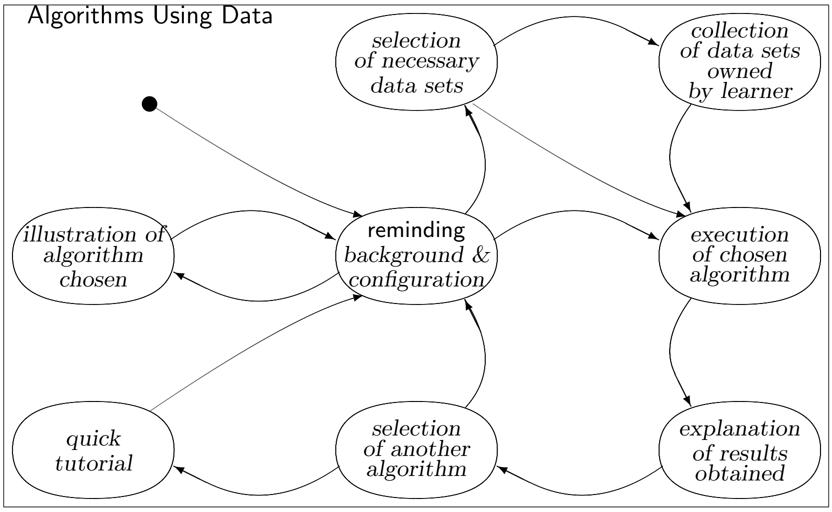

Fig. 3. Dialogue Scene for Application of Algorithms To Data

for reminding the background and algorithms selection in the decision support scenario is displayed in Figure 4. Now the learner may proceed on the data he /she has on hand. In this case the learner obtains results as displayed in Figure 5.

4 Conclusion

The DaMiT systems supports users to become knowledgeable in data mining. Users can learn first the fundamentals of data mining, then learn how to cope

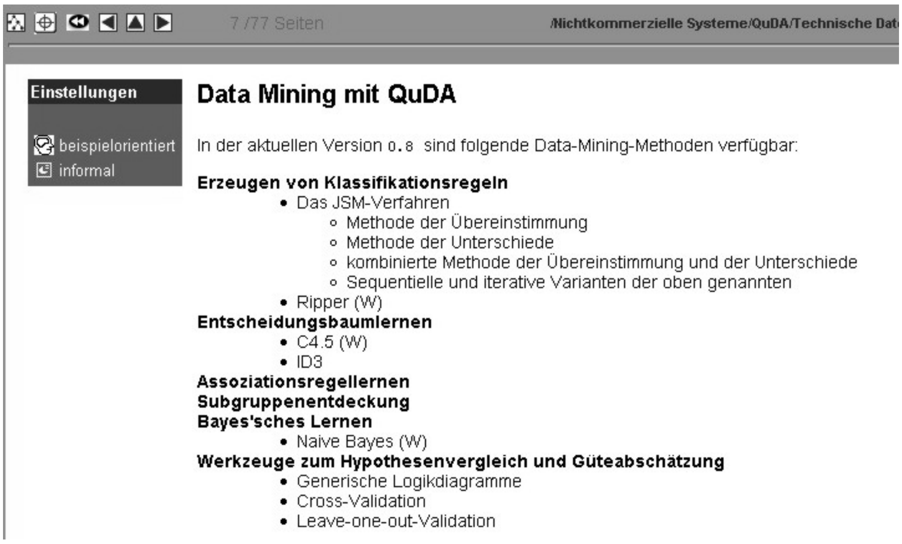

Fig. 4. Selection on Available Methods and Algorithms in the QuDA Section of DaMiT

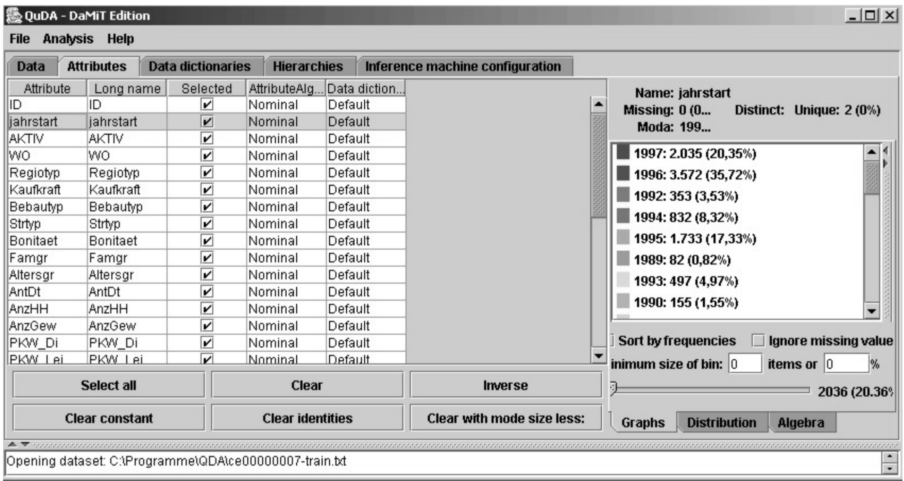

Fig. 5. Execution of Chosen Algorithms on Learners Data in the QuDA Section of DaMiT

with data and algorithms and finally learn how to proceed with their data and how to interpret results generated by data mining algorithms on their data.

Thus, the DaMiT system can be used as an add-on to decision support systems. It can also be used stand-alone for decision making as long as data mining algorithms are used. In this paper we gave an insight into the system by discussing the architecture of the system and the story boarding of the systems use.

References

[Br02] M. A. Bramer, Knowledge discovery and data mining. IEEE Professional Applications of Computing Series 1, The Institution of Electrical Engineers Publishers, 1999.

[Cau00] J. Caumanns, Automatisierte Komposition von wissensvermittelnden Dokumenten fr das World Wide Web. PhD, BTU Cottbus, Faculty of Mathematics, Natural Sciences and Computer Science, Cottbus, 2000.

[DaM01] DaMiT. The intelligent data mining tutorial workbench. http://DaMiT.dfki.de

[DMC] Data Mining Cup. http://www.data-mining-cup.de

[GY02] P. Grigoriev and S. Yevtushenko, JSM - Reasoning as a data mining tool - Experiments on UCI datasets. Proceedings of the 7th Russian National Conference on Artificial Intelligence, CAI-2000 PhysMathLit Publisher, 37–43/2002.

[HKMW01] H. Hippner, U. Küsters, M. Meyer, and K. Wilde, Handbuch Data Mining im Marketing. Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, 2001.

[KlM03] M. Klettke and H. Meyer, XML and databases. dpunkt.Verlag, Heidelberg 2003.

[KlZy02] W. Klösgen and J.M. Zytkow, Handbook of data mining and knowledge discovery. Oxford University Press, 2002

[LOM02] LOM. IEEE Learning Technology Standards Committee's (LTSC) LOM Working Group. Learning Object Meta-Data (LOM) http://www.imsproject.org

[Pop35] K. R. Popper, Logik der Forschung. Tübingen, 1934.

[RusN03] St. Russell and P. Norvig, Artificial intelligence - A modern approach. Prentice Hall, Upper Saddle River, 2003.

[RTT02] O. Rostanin, B. Tschiedel, and B. Thalheim, Szenario-basiertes e-Learning für adaptive Inhaltspräsentation. Proc. LIT'2002, infix Publishers, 2002, 330- 338

[ScT00b] K.-D. Schewe and B. Thalheim, Modeling Interaction and Media Objects. Proc. NLDB' 2000, LNCS 1959, 2002, 313-324.

[StD02] J. Strutz and G. Degel, Offene Übungsaufgaben und Praktika im e-Learning. Einbindung, Auswertung und Bewertung im Tutorsystem DaMiT. Proc. LIT'2002, infix Publishers, 2002, 410 - 420

[STY99] O. Seleznev, B. Thalheim, and S. Yigitbasi, Statisitical evaluation methods for dynamics in environmental data. Preprint Computer Science Institute of Brandenburg University of Technology at Cottbus, I-07/1999.

[TAT03] B. Thalheim, A. Binemann-Zdanowicz, B. Tschiedel, Content modeling for e-learning services. Proc. SCI'2003, Orlando, July, 2003.

[Tha00] B. Thalheim, Entity-relationship modeling – Foundations of database technology. Springer, Berlin, 2000.
See also http://www.informatik.tu-cottbus.de/∼thalheim/HERM.htm

[ThD01] B. Thalheim and A. Düsterhöft, SiteLang: Conceptual modeling of internet site. Proc. ER'2001, LNCS 2224, Springer, 2001, 179-192.