A Model-View-Controller architecture for Knowledge Discovery

M. Castellano[1], N. Pastore[2], F. Arcieri [2], V. Summo[2]
& G. Bellone de Grecis[2]
[1]*Dipartimento di Elettrotecnica ed Elettronica, Politecnico di Bari, Italy*
[2]*Global Value, ACG srl, an IBM and FIAT Company, Italy*

Abstract

In this paper we present a flexible mining architecture able to define and validate a process, generally applicable in different e-business sectors, for providing new added value e-Knowledge services. The architecture is designed on the Model-View-Controller pattern in order to get a clear separation of the component functionalities and covers the whole process of Knowledge Discovery in Databases (KDD) and in Text (KDT) for the extraction of patterns starting from structured and unstructured data. When a service request comes to the system, this is received by a Controller that will call one or more Miners to provide the results. The Miners represent the Model of the system and, by using a Kernel, dynamically activate either the KDD or the KDT process, depending on the typology of the service. As View, the system makes use of the CWM standard for the representation of metadata about models and results of the mining processes. The Kernel includes two different Focuses, for the selection of structured and unstructured data. The results of the Focuses are passed to a unique Pattern Extraction step, where Web and Data Mining algorithms are collected for the analysis. Finally, an Evaluation step interprets the utility of the extracted patterns. The proposed solutions can be suited in a distributed mining environment, where a set of services are managed and made available as a means of meeting the diverse needs of the e-business world.
Keywords: data mining architecture, web mining, MVC pattern.

Data Mining V, A. Zanasi, N. F. F. Ebecken & C. A. Brebbia (Editors)

1 Introduction

Data mining has become useful over the past decade in business to gain more information, to have a better understanding of running a business, and to find new ways and ideas to extrapolate a business to other markets. Its analytical power to find new business areas and opportunities does not need to be proven anymore to both the business and data-mining analysts. Today, data mining is no longer thought of as a set of stand-alone techniques, far from the business applications; over the last three years, integrating data mining with mainstream applications has become an important part of e-commerce applications. Recent research issues are dealing with the integration of mining technologies with relational databases and their business-oriented applications in a more flexible and modular way. The goal is to find new and interesting patterns in the data and somehow use the gained knowledge in business decisions. The integration of mining results into the operational business is usually done in an ad-hoc manner. With the integration of mining, the focus shifts toward the deployment of mining in business applications. Even if the purpose of the model is to increase knowledge of the data, the knowledge gained needs to be organized and presented in a way that the end user can use it. Depending on the business requirements, the deployment phase can be as simple as generating a report or as complex as implementing a repeatable data mining process. In order to achieve this purpose, data mining systems have to overcome unique challenges; they need to combine access to diverse and distributed data sources with the large computational power required for many mining tasks. High performance of a data mining system and the design of its architecture are not only a question of choosing the proper algorithms and efficient implementations of them. In the assisting the process of understanding how to decide which techniques are applicable and how to set about the whole process of mining own businesses, our contribution presents, as an alternative, an appropriate and flexible architecture able to meet the diverse e-Knowledge needs of e-business by providing and managing e-Knowledge services based on data and web mining techniques. The real aim of the proposed system is to define and validate a sequence of steps that can be followed for all mining activities, such a process that is generally applicable in different e-business sectors, and describes how to identify interesting and new patterns by covering the whole process of the knowledge discovery in a distributed and heterogeneous environment. The purpose of this paper is not to present new algorithms or other techniques, but to show how to adopt them in a flexible mining architecture in order to provide e-Knowledge services.

2 Backgrounds

2.1 Systems for Knowledge Discovery processes

Knowledge Discovery is an interdisciplinary area focusing upon methodologies for extracting useful knowledge from data. The ongoing rapid growth of online

Data Mining V, A. Zanasi, N. F. F. Ebecken & C. A. Brebbia (Editors)

data due to the Internet and the widespread use of databases have created an immense need for discovering methodologies. The challenge of extracting knowledge from data draws upon research in statistics, databases, pattern recognition, machine learning, data visualization, optimization, and high-performance computing, to deliver advanced business intelligence and web discovery solutions [10,12]. The realization of a general purpose, fully automated, knowledge discovery system is difficult to obtain and in last years many issues of research have focused on ways of manually applying traditional machine-learning and discovery methods to data stored in databases. Recently, a deep attention has been moving towards more fully automated approaches. The Knowledge Discovery in Databases (KDD) Model proposed by Piatetsky-Shapiro, Matheus and Chan [3] represents a starting point for our solution. They defined a system as a collection of the following components:

- *Controller*, to control the invocation and parameterization of other components
- *Database Interface*, to generate and process database queries
- *Knowledge Base*, to contain domain specific information
- *Focus*, to determine which portions of data to analyze
- *Pattern Extraction*, to collect pattern-extraction algorithms
- *Evaluation*, to evaluate the interestingness and utility of extracted patterns

Their model represents an abstraction of what usually occurs in KDD systems and is helpful for contrasting the tradeoffs between autonomy and versatility.
In this paper, we will start from this solution in order to design a more complex and complete architecture for the providing of new added value e-Knowledge service.

2.2 The Model-View-Controller design pattern

Several problems can arise when applications contain a mixture of data access code, business logic code, and presentation code. Such applications are difficult to maintain, because interdependencies between all of the components cause strong ripple effects whenever a change is made anywhere. High coupling makes classes difficult or impossible to reuse because they depend on so many other classes. Adding new data views often requires re-implementing or cutting and pasting business logic code, which then requires maintenance in multiple places.
The Model-View-Controller design pattern solves these problems by decoupling data access, business logic, and data presentation and user interaction. It has its roots in Smalltalk [9], where it was originally applied to map the traditional input, processing, and output tasks to the graphical user interaction model. Anyway, it is a high-level pattern in that it concerns itself with the global architecture of a program and tries to provide a classification of the different kinds of objects that make up an application. According to the pattern, there are three types of objects: *Model* objects, *View* objects, and *Controller* objects. The pattern defines the roles that these types of objects play in the application. Each


Data Mining V, A. Zanasi, N. F. F. Ebecken & C. A. Brebbia (Editors)

of them is separated from the others by abstract boundaries, and communicates with objects of the other types across those boundaries.

The Model object represents enterprise data and the business rules that govern access to and updates of this data. Often the model serves as a software approximation to a real-world process, so simple real-world modeling techniques apply when defining the model.

The View object renders the contents of a model. It accesses enterprise data through the model and specifies how that data should be presented. It is the view's responsibility to maintain consistency in its presentation when the model changes. This can be achieved by using a push model, where the view registers itself with the model for change notifications, or a pull model, where the view is responsible for calling the model when it needs to retrieve the most current data. The Controller object translates interactions with the view into actions to be performed by the model. In a stand-alone GUI client, user interactions could be button clicks or menu selections, whereas in a Web application, they appear as GET and POST HTTP requests. The actions performed by the model include activating business processes or changing the state of the model. Based on the user interactions and the outcome of the model actions, the controller responds by selecting an appropriate view.

3 Mining architecture prerequisites

Nowadays, knowledge discovery systems face challenging problems from real world databases which tend to be dynamic, incomplete, redundant, noisy, sparse and very large. There is no universal strategy to configure an optimal data mining environment; as reference, we can consider the architectures proposed by Chattratichat et al. [1] and Krishnaswamy et al. [5]. The architecture that we propose in this paper has been designed in a way to follow some prerequisites in order to support mining elaborations in a heterogeneous and distributed system.

As our architecture wants to be an e-Knowledge services provider, the first requirement is that each service has to be built as a module, independent from other ones. The possibility to add one or more service in the system must be guaranteed every time without doing any substantial change, so to have full control and a complete supervision.

Another important consideration is the flexibility. Usually users have different business goals in mind when they need to discover hidden knowledge in their data. Hence, the architecture should be flexible in supporting various data and web mining techniques and algorithms [13]. This can be achieved by providing a clear separation between the process logic and the e-Knowledge services that the architecture has to provide. A mining system should also incorporate optimization strategies and a high performance especially for large data sets in order to enable mining elaborations with an acceptable response times.

In general, factors like the inclusion of services, algorithms, changes on system topology, different user privileges or system load restrictions have a deep impact in the overall performance of the system. Important decision like task

Data Mining V, A. Zanasi, N. F. F. Ebecken & C. A. Brebbia (Editors)

scheduling, resource management or partition schema should be tuned again almost every time one of these factors is modified [11].

4 The Knowledge Discovery architecture

In this paper we propose a Knowledge Discovery architecture as a modular and flexible structure. The purpose is to generate and make available e-Knowledge services by integrating data and web mining techniques, by extracting, selecting, processing and modeling huge amount of data in a distributed and heterogeneous content environment of informative resources, coming from both the unstructured WWW and databases or other structured sources. The paper assumes that the step of the information retrieval in the web mining process is done, so that the start up is represented by the step of the information extraction, where all data have already been stored in the knowledge repository. The system represents a distributed data and web mining tool, where a set of services are managed and made available through a controller. As system, we intend a collection of components and each one with a well defined task. In the next Figure 1 the architecture of the Knowledge Discovery System is showed.

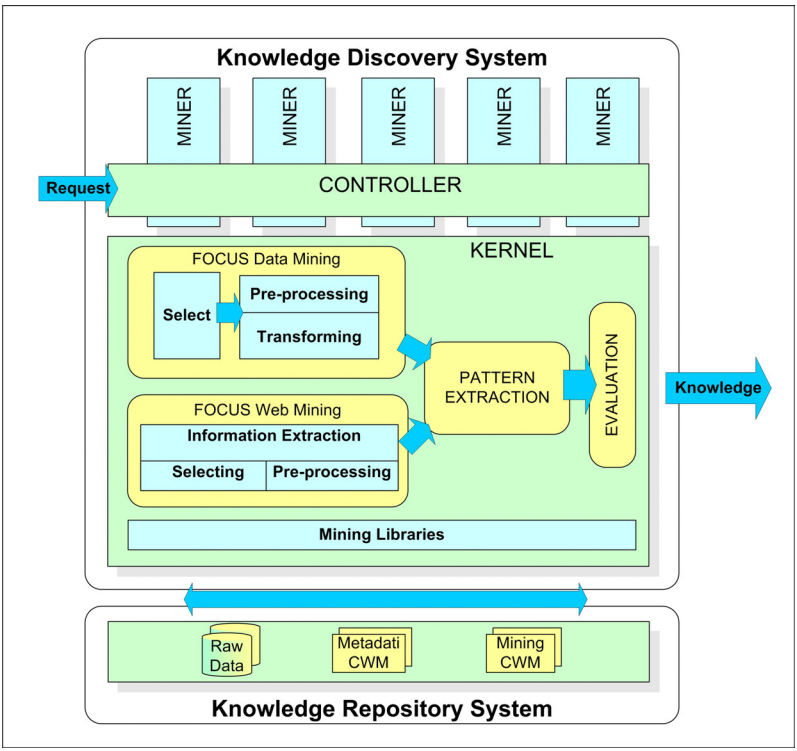

Figure 1: The Knowledge Discovery architecture.

Data Mining V, A. Zanasi, N. F. F. Ebecken & C. A. Brebbia (Editors)

The architecture is mainly composed by:

- a Controller, that receives e-Knowledge services requests and then activates the business logic of the correspondent service.
- a set of Miners, that represent the business logic of the e-Knowledge services. One or more miners can realize one of them.
- a Kernel, that represents the business logic of the Knowledge Discovery in Databases (KDD) and in Text (KDT) processes, by holding data mining and web mining algorithms.
- a Knowledge Repository system, that represents the Business Information System.

When a client requires an e-Knowledge service, it is received by the Controller that will call one or more Miners to provide the results.

Miners stay at the level where information is produced and represent operations that realize a service. They can work either by loading from the Knowledge Repository system the mining models associated to the required e-Knowledge service or by activating a training process of Knowledge Discovery in Databases or in Text in the Kernel according to the typology of the service. Miners are building blocks that can be used to build a complex application. In this case, the application is represented by an e-Knowledge service and it may be formed by one or more Miners.

The Kernel represents the core of the system and covers the knowledge discovery process [2] by working on data and by building new mining models. The Kernel follows the process of discovering knowledge starting from raw data and involving iterations of three main stages: data preparation, data and web mining, and results analysis. A variety of software tools already exists and is available for each of the stages, but they suffer from two big problems, weak interoperability among them and inability to substitute one tool with another for the same application, where by application we mean a major task in each stage of the knowledge discovery process, such as dirty data cleaning, data integration, explorative analysis, principal components analysis, clustering, classification and so on. An integrated architecture that takes the user and the various e-Knowledge services through all three stages of the knowledge discovery process in a unique workflow is the aim of the proposed system. The Kernel is mainly composed by the following components:

- Data and Web Mining Focus
- Pattern Extraction
- Evaluation
- Mining Libraries

Focuses are useful in order to determine which data are useful to provide the service and because there could be the necessity to have detailed information about the structure of data, table or database to be accessed, to know the most appropriate fields for the required service, to know the required input for the next

Data Mining V, A. Zanasi, N. F. F. Ebecken & C. A. Brebbia (Editors)

step of the Pattern Extraction. Data Mining Focus holds the phases of selecting, pre-processing and transforming, while Web Mining Focus holds the phase of the Information Extraction. The results of the Focuses are then passed to a unique Pattern Extraction step [4], where a set of Web Content, Structure and Usage Mining techniques [6,7,8], together with other Data Mining algorithms, are collected for the analysis. Finally, an Evaluation step interprets the utility and the carefulness of the extracted patterns. This last step is important because only a part of the extracted patterns is really of interest and observes the knowledge and the objectives of the service required by the user. The Kernel activity is supported by the Mining Libraries that represent a set of Data and Web Mining algorithms, able to solve each stage previously described.

The Knowledge Repository system represents the Business Information System of the whole architecture and its functions are those of:

- Repository for raw data
- Repository for knowledge metadata
- Repository for mining model

For the sharing of the service results, the storing of data and metadata and the representation of mining models, the CWM standard [14] has been used in order to integrate the knowledge coming from different sources. The CWM lets the using of a number of sub-metamodels which represent common warehouse metadata in the following major areas of interest to data warehousing and business intelligence:

- Data Resources, that includes metamodels that represent object-oriented, relational, record, multidimensional, and XML data resources.
- Data Analysis, which includes metamodels that represent data transformations, OLAP, data mining, information visualization, and business nomenclature.
- Warehouse Management, which includes metamodels that represent warehouse processes and results of warehouse operations.

4.1 The Knowledge Discovery architecture based on the MVC design pattern

Many applications, especially the J2EE ones, can be described in terms of object. The application's functionality is separated across these objects, or functionalities, to provide separation of responsibility, reuse and improved scalability. The separation of objects may be a physical separation, where each one is located on a separated hardware resource, or purely logical. As shown in the next Figure 2, we design the proposed architecture of Knowledge Discovery on the Model-View-Controller design pattern:

The MVC architecture of the Knowledge pattern hinges on a clean separation of objects into one of three categories: a controller for handling events that affect

Data Mining V, A. Zanasi, N. F. F. Ebecken & C. A. Brebbia (Editors)

the model or views, a model for maintaining and managing data, views for displaying all or a portion of the data.

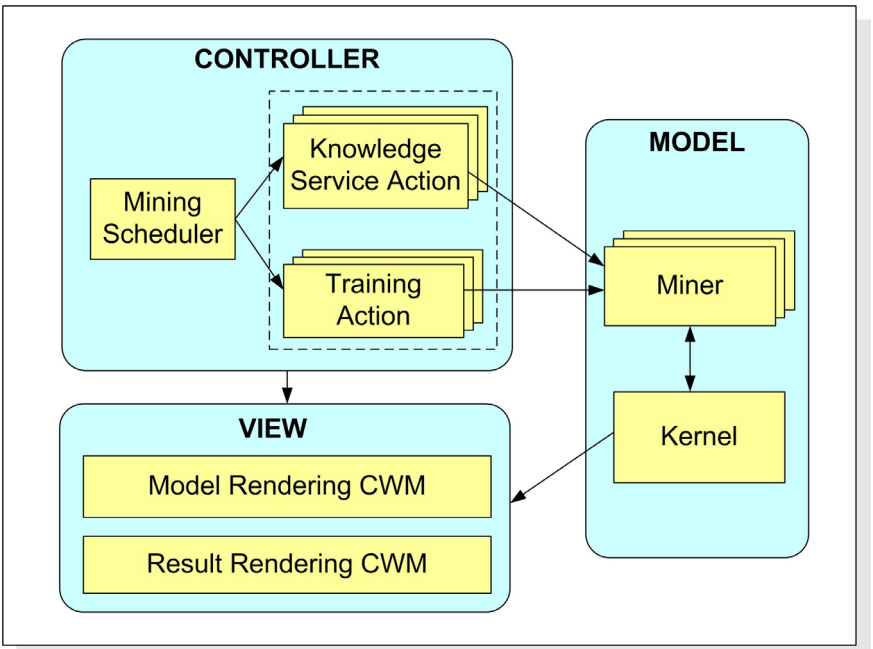

Figure 2: The Model-View-Controller architecture of the system.

When a client requires an e-Knowledge service, the request is received by the Controller through a Mining Scheduler, analyzed and then forwarded to an action. Each action corresponds to an e-Knowledge service and may call one or more Miners to get the results. There could be two kinds of action according to the typology of the request. They are:

- Knowledge Service Actions
- Training Actions

The Knowledge Service Actions are activated by the Mining Scheduler when there is a request of an e-Knowledge service. In this case, each action calls Miners that have the task to load one or more mining models, previously built by the Kernel, from the Knowledge Repository. The Training Actions are activated by the Mining Scheduler when the system has to re-train a mining model. Each Training Action calls the Miners that have the task to activate the process of Knowledge Discovery in Databases or in Text in the Kernel, according to the typology of the service. Miners, together to the Kernel, represent the business

logic of the service and for this reason they are located in the Model object of the MVC design pattern.

The View object has the task to render to clients results produced by the Model.

In the proposed architecture, there are two different kinds of results:

- Service Results, as output of a service request
- Mining Model Construction, as output of the Knowledge Discovery process.

In accordance to this, there are two different kinds of Views:

- Result Rendering CWM
- Model Rendering CWM

Both results will be represented through the CWM [] standard with the only difference that in the first case they are given back as result of an effective service request, in the second case they are used to extend the initial knowledge base through the building of new and more precise mining models.

5 Conclusion and remarks

In this work we have shown a decision support system through a clean separation of all its functionalities by adopting the Model-View-Controller design pattern. As remarkable results, the system realizes a decoupling between the e-Knowledge services that it is able to provide and the knowledge discovery processes adopted for the information mining and this is made possible by adopting the concept of reusable components as miners, which lets the creating of new services by using the same building blocks. Furthermore, to improve a clearer understanding of the contents and a more efficient sharing of knowledge, the use of the CWM standard has been considered. It allows not only to represent in a standard way the processing results as output of an effective service request, but also to model mining patterns for a better reusability of them. Finally, this work doesn't pretend to present new algorithms or new mining techniques, but just to show a model of knowledge discovery system able to be flexible and more efficient.

Acknowledgements

The authors would like to thank Mario Logrieco and Vito Belviso for the important roles that they played during the development of this work.

References

[1] J. Chattratichat, J. Darlington, Y. Guo, S. Hedvall, M. Kohler, and J. Syed: An Architecture for Distributed Enterprise Data Mining. In: 7th

International Conference on High Performance Computing and Networking Europe (1999).
[2] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy, Advances in Knowledge Discovery and Data mining, MIT Press, (1996).
[3] C.J. Matheus, P.K. Chan, G.P. Shapiro: Systems for Knowledge Discovery in Databases. In: IEEE TKDE Special Issue on Learning & Discovery in Knowledge-Based Databases, (1993).
[4] J. Han, M. Kamber: Data Mining Concepts and Techniques, Morgan Kaufman Publishers, (2001).
[5] S. Krishnaswamy, A. Zaslavsky, S.W. Loke, An Architecture to Support Distributed Data mining Services in E-Commerce Environments. In: Workshop on Advanced Issues of E-Commerce and Web/based Information Systems (2000).
[6] T. Nasukawa, T. Nagano: Text Analysis and Knowledge Mining System. In: IBM Systems Journal, Vol 40, No 4, (2001).
[7] O.R. Zaiane, M. Xin, J. Han: Discovering Web Access Patterns and Trends by Applying Olap and Data Mining Technology on Web Logs. In: Advances in Digital Libraries, (1998) 19-29.
[8] J. Srivastava, R. Cooley, M. Deshpande, P.N. Tan: Web Usage Mining: Discovery and Applications of Usage Patterns from Web. In: ACM SIGKDD, (January 2000).
[9] Smalltalk, http://www.smalltalk.org/.
[10] W.F. Cody, J.T. Kreulen, V. Krishna, W.S. Spangler: The Integration of Business Intelligence and Knowledge Management. In: IBM System Journal, Vol 41, No 4, (2002).
[11] J.M. Pena, E. Menasalvas: Towards Flexibility in a Distributed Mining Framework. In: DMKD Workshop (2001).
[12] L. Kurgan, K.J. Cios, M. Trombley, The WWW Based Data Mining Toolbox Architecture. In: Proceedings of the 6th International Conference on Neural Networks and Soft Computing, Poland, (2002) 855-860.
[13] S.K. Pal, P. Mitra: Web Mining in Soft Computing Framework: Relevance, State of the Art and Future Directions. In: IEEE Transactions on Neural Networks, Vol.13, No 5, (2002).
[14] Object Management Group: Common Warehouse Metamodel (CWM) Specification. OGM Press, Needham, Massachusetts, (2001).

Data Mining V, A. Zanasi, N. F. F. Ebecken & C. A. Brebbia (Editors)