International Journal of Computer Science and Applications,

Vol. 10, No. 1, pp. 63 – 77, 2013

SOFTWARE ARCHITECTURES OF APPLICATIONS USED FOR ENHANCING ON-LINE EDUCATIONAL ENVIRONMENTS

MARIAN CRISTIAN MIHĂESCU

Department of Computer Science and Information Technology, University of Craiova,
Craiova, Dolj, Romania
mihaescu@software.ucv.ro
http://software.ucv.ro/~cmihaescu/

This paper presents two prototype software architectures that are used for building applications that are used for enhancing the on-line educational environments. The software architectures are based on the concept of educational data/knowledge flow. The presented software architectures are designed with the goal of setting up an infrastructure whose implementation will offer functionalities for on-line educational environments. The core functional goal of the two architectures is professional student search. This goal is achieved by employing two classification mechanisms: a decisions tree based classifier and a Bayesian classifier. The final business goal of the architectures is searching for a tutor student among all other colleagues and searching for students to which messages are to be sent in an advanced messaging system. Such functionalities offered by a service type application that runs along the on-line educational environment is intended to enhance the educational quality of the on-line educational environment that is offered to students.

Keywords: on-line educational system, educational data mining, learner classification

1. Introduction

This paper addresses the problem of designing software architecture for building applications used for enhancing on-line educational environments. Enhancing is obtained by providing platform side intelligent and advanced functionalities in the shape of a professional search systems, recommender systems or learning path builder systems. Such enhancements are obtained when several conditions are met: the courses have a well-structured hierarchical nature, necessary experiences (i.e., activities) are stored in a structured way and a feasible data analysis process is set up. All these conditions need to put together within a proper designed software architecture that meets all the requirements for such applications.

Each of the above presented prerequisites is equal important. The first prerequisite is concerned with having a proper infrastructure setup. This means that the on-line educational environment has a well refined hierarchical structure in which all learning assets (e.g., disciplines, chapters, quizzes, etc.) are properly defined. For example, at chapter level there needs to be defined a concept map [Novak and Canas (2006)] with which all other learning assets are associated. That is why, each quiz question associated to a chapter is also associated with a concept.

The second prerequisite is concerned having enough data for building a data analysis process. The data represents experiences had by the involved partied during usage of the on-line educational system. From this point of view, several aspects are equally important: the number of features representing a user, the total number recorded activities and the overall quantity of recorded data. These three conditions make sure that analyzed data is statistically relevant and thus may be successfully used in a data analysis process. As a general key aspect the data needs as many conditionally independent attributes as possible and as much data as it can be obtained. Still, these aspects are not key aspects since more data and more feature do not necessarily mean an increase in the accuracy of data analysis process. This concern is mainly addressed by the effective debugging of the data analysis process. All available data represents the input dataset used for building the user model of the student. The user model represents the baseline model against which all current students are characterized. Thus, the data provided by the currently existing students in the one that is actually used as "query" for the currently existing learner model.

The third prerequisite is concerned with setting up an adequate data analysis process. From this point of view the following issues issues need to be addressed: setting up a proper requirement, choosing the proper input dataset in terms of parameters number, types and values, choosing the right algorithm that produces usable and interpretable results and setting up a proper data/knowledge flow. This paper presents two sample data flows that respond to two applications from the area of professional search systems. The first example is a tutor search system for on-line educational environments and the second example is an advanced messaging system based also on a similar student search system. Both tools integrate machine learning algorithms for building a learner's model and provide advanced functionalities for served on-line educational environment.

In general, there are two types of algorithms: unsupervised and supervised. The unsupervised algorithms (e.g., clustering, regression, etc.) are used to discover patterns in data. The supervised algorithms (e.g. classification, decision trees, SVM, neural networks, etc.) are used to classify new items, which in educational applications may be users (e.g., students, professors, etc.) or educational assets (e.g., courses, documents, quizzes, concepts, etc.). The final key aspect of the data analysis process is evaluation of the quality of the data analysis process. This evaluation gives confidence in using the final results and may provide important information regarding necessary actions needed to improve the accuracy of the results provided by the data analysis process. The continuous improvement in quality of the data analysis process is the key aspect in having a truly machine learning based data analysis. Although the application domain is not quite critical, as for example in medical applications, the concern for proper evaluation of obtained knowledge is a key aspect of the designed system.

The design of eLeTK (e-Learning Enhancer Toolkit) is modular in packages such that continuous development is feasible. The main packages are: input loader, filters, data processing, evaluation, output builder and configuration. Each package contains classes that implement specific business logic such that they may be put together to form a data/knowledge flow.

The proposed design of eLeTK makes it suitable for building systems that run along on-line educational environments and enhance their educational purposes. From a software systems point of view, a setup of eLeTK works as a service for an on-line educational environment in an attempt to offer the intelligent character. The designed tools have the goal of implementing the mental model of a professor gathering the activities that are performed in classical face-to-face educational environments. The need for such tools is important for enhancing the learning experience of students and for providing the necessary knowledge about the students to professors in an environment where only data about performed activities is available. From this perspective it becomes obvious the need for advanced data processing techniques that need to be properly set within a software in order to produce desired knowledge.

2. Related Work

In the last decades there has been a lot of effort in the new domain of EDM (Educational Data Mining). In [Romero and Ventura (2007)] there are presented many tools that were designed and implemented for this domain. The main issues that distinguish educational data mining from other domains where such state of the art algorithms are used are related to domain specific data, domain specific objectives and goals, custom adaptation of classical data analysis techniques and custom interpretation and visualization of results.

The general approach of such systems is based on the some core processes. Firstly, it is assumed that responsible academics and educators design, plan, build and maintain an educational environment. Second, students interact with the educational environment thus producing interaction data and the educational environment has the ability to store all the performed activities in a format that is well structured and may be processed in order to provide needed data. Thus, the activity data along with educational environmental data (e.g., course information, academic data, quizzes, etc.) represent one input in the educational data-mining tool, which in our case is eLeTK. The obtained data-mining tool is used to show discovered knowledge, recommendations or learning paths to students or other involved parties.

There are many general data analysis tools which are not designed to work with data from a specific domain. Among such tools there are DBMiner, Clementine, Intelligent Miner, Weka, etc. [Klosgen et.al. (2002)]. The main drawback of these tools in the context of EDM is that they cannot be used in educational contexts by students or professors. In a most optimistic case, these tools are used by experienced data analysts with a good background in a specific on-line educational environment. In this way there are performed an off-line data analysis which may produce knowledge regarding analyzed data or recommendations for involved parties. From this point of view, eLeTK represents a new layer between a general data analysis tool and a specific on-line educational environment. The main goal of the proposed software architecture is to provide integration of three main pieces: the underlying software packages or libraries implementing of machine learning/data mining algorithms, the activity data provided by an on-line educational environment and the on-line educational environment itself.

So far, in the above presented general context there were created tools oriented towards educators [Romero et.al.(2004), Merceron and Yacef (2004)] and tools oriented towards academics responsible and administrators [Grob et.al. (2004), Urbancic et.al. (2002)]. These tools perform tasks as associations, pattern analysis, classification, clustering, text mining, statistics and visualization.

In the area of on-line educational environments, there are two types of systems: classical learning content management systems and intelligent web-based educational systems. Some examples of commercial learning content management systems are Blackboard, Virtual-U, WebCT, TopClass, etc. and some example of free LCMS are Moodle, Ilias, Claroline, aTutor, etc. [Paulsen (2003)]. On the other hand, some examples of intelligent educational systems are SQL-Tutor, German Tutor, ActiveMath, VC-Prolog-Tutor, AHA!, InterBook, KBS-Hyperbook, WebCOBALT [Brusilovsky and Peylo (2003)].

One of the outputs of all web-based educational systems is the activity data performed by their users. These web-based education systems can normally record the student's accesses in web logs that provide a raw trace of the learners' navigation on the site. There are several types of logs [Srivastava et.al. (2000)] and there are also AI techniques for monitoring student learning process [Camacho et.al. (2008)].

The main tasks that are generally implemented are data preprocessing (e.g., data cleaning, user identification, session identification, transaction identification, data integration) [Koutri at.al. (2005), Zorilla et.al. (2005)], data analysis (e.g., decision tree construction, rule induction, artificial neural networks, instance-based learning, Bayesian learning, logic programming, statistical algorithms, etc.) [Klosgen et.al. (2002)] or web mining (e.g., clustering, classification, outlier detection [Klosgen et.al. (2002)], association rule mining, sequential pattern mining [Agraval and Sirkant (1995)], text mining [Grobelnik et.al.(2002)]).

One of the main drawbacks of this approach is that is fully data centered. Thus, the change regarding user preferences [Burlea et.al. (2011)] aspects are neglected. Still, from a pedagogical point of view this approach may be considered one hundred percent objective which is usually regarded as an advantage.

A plus in the domain is brought by custom usage of a hierarchical way of structuring e-Learning content. The conceptual-visual dynamic schemes (CVD-schemes) are the marked oriented graphs introduced in cognitonics domain [Fomichov (1994), Fomichov (2006)] for inventing effective teaching analogies. Such graphs establish a correspondence between the components of a piece of theoretical material to be studied and the components of a well-known or just created by the teacher but bright fragment of the inner world's picture of the learner.

3. Employed Infrastructure and Methods

3.1. *The e-Learning Environment*

Tesys [Burdescu and Mihaescu (2006)] on-line educational environment is primarily a classical collaborative software system in which all involved parties (e.g., administrators, professors and students) perform their main responsibilities. Administrators are responsible for managing the general infrastructure as curriculum of learning programs (i.e., the studied disciplines and the assigned professors), users, etc. Some examples of actions that are currently performed by administrators are enrolling students to needed learning programs, giving them grants to take the failed exams, passing them into the next year of study, communicating with students and professors.

An important aspect regarding the functionality of the e-Learning regards the types of activities the students are performing. Some of the currently implemented activities are login, logout, taking tests and exams, communicating with other students or with professors, etc. All these activities represent a repository of experiences that are very valuable in a data analysis process.

Currently, the Tesys e-Learning platform has five running instances each with one up to five study programs. For example, one setup manages four different study programs with duration of three years where more than 100 professors and almost 1000 students are currently active. This setup manages the following learning assets: 120 courses, almost 1000 chapters, almost 5000 quiz questions, almost 1000 taken quizzes and exams and almost 10,000 sent messages. All performed activity is recorded into log files and in a database. In this way, for a student there may be computed many features describing the performed activity.

In general the features are of two types: one regards indicators of the quiz activities and one regards the time spent on different activities. Some of the features from the first category are: *positivCount* – the number of correctly answered questions; *correctPercent* – the percentage of correctly answered questions from the total number of questions; *totalTries* – the total number of tries (answered questions); *avgTries* – medium number of tries per question. Some of the features from the second category are *avgQuestionTime* – on average, how long (in minutes) it takes for a student to answer a question; *totalTime* – total time spent on testing.

3.2. *Data Analysis Techniques and Weka*

The data analysis [Han et.al. (2011)] techniques fall in general three categories: unsupervised, supervised and rule based. The most common unsupervised method is clustering and may be of several types: partitioning, EM, hierarchical, fuzzy, etc. The main supervised methods relate to classification algorithms: Decision Trees, Bayesian Networks, Vector Space Classification, CART. The most common algorithm for building association rules is Apriori.

All these algorithms are mature and proved their effectiveness in different domains outperforming classical statistical data analysis procedures.

Weka [Witten and Frank (2005)] is a java software implementation of a large set of machine learning and data mining algorithms. The implementations are generic and therefore data coming from any application domain (e.g., e-commerce, bioinformatics, e-Learning, etc.) may be analyzed. From this point of view, the main issue regards designing of new algorithms, implementing them into Weka (or similar libraries) in an attempt of improving the time and space complexity. Still, the main goal is to integrate of-the-shelf implementations of algorithms into custom designed applications. From this perspective, the need for integration framework or software architecture is obvious. The main goal of the proposed architectures is to provide a framework in which all the needed pieces (data and software) are put together in such a way to provide relevant knowledge.

This paper presents the integration framework of two classification algorithms: decision trees and Bayesian classifier. In Weka, J48 [Loh (2008)] is the implementation of the C4.5[Quinlan (1993)] algorithm, a data analysis procedure which generates a decision tree in order to classify new data. Weka also provides implementation for Bayes algorithm [Lowd (2005)] which is also used for classification purposes but in a different context, i.e. to classify new items, which in educational applications are students.

4. eLeTK Architecture

4.1. *General Requirements*

The main issues related to existing libraries implementing machine learning and data mining algorithms relate to their generic approach. Existing libraries like Weka or Mahaout [Apache Mahout Project, (2013)] implement algorithms in a generic way without any particularity for the domain from which data may come. That is why, it is relatively difficult to integrate and accommodate domain specific machine-learning problems and even more difficult to integrate into existing production environments that produce the data that need to be analyzed. These are the main issues that eLeTK wants to solve by becoming a layer between the production environments (i.e., on line educational systems), the data that is to be analyzed and the specific requirements. Intuitively, eLeTK may be regarded like a software application mainly designed to be used by educational domain experts with some background in data analysis in order to produce complete data workflows that solve educational problems.

Designing and developing a software system that addresses the above presented issues in the domain of Educational Data Mining is an interdisciplinary problem related to the following equally important areas: Machine-Learning/Data Mining/Information Retrieval, Software Engineering and Human Computer Interaction.

The constraints from the algorithms point of view regards the diversity of algorithms, the wide range of data models, the difficulty of obtaining interoperability among models and difficulty of integration into specific information systems which represent the main source of data. For example, Weka workbench is quite weak regarding the interoperability among different data modeling methodologies.

The main general activities that need to be managed regard obtaining raw activity data and representing items, performing a data analysis process, management of obtained data models, assessing the performance of the obtained models and presentation into an understandable format to end-users. These processes need robust and scalable data management and flexible integration architecture. The management of the processes should be performed in a user friendly manner by educational experts with core background understanding of data analysis methods and capabilities. The main goal of the architecture is to loose pressure from the data management activities and to provide a flexible way of defining data processes that solve educational problems.

The overall data processing activity is regarded as a data workflow. The processing pipeline of the data workflow always starts from raw data provided by the information system which is represented in our particular situation by the on-line educational system. During the data analysis, the data may take different shapes such as structured as needed by the data analysis engine, as a data model, knowledge or other type of representation. The activities that may be performed upon different types of data are streaming, loading, modeling, moving, recording, replicating or crawling. All these activities may be performed also as a scheduled job, without administrator's presence. Each activity may be defined as a data workflow in which there are specified all the necessary activities, such as data preprocessing and translation, data consistency verification, data quality check or data validation. Data workflow may also be merged, copied, crawled or recovered.

One of the most important core information about eLeTK is the Data Resource Registry (DSR) which records metadata about performed activities, available services, data, models, knowledge or workflows. DSR manages the core assets handled by an instantiation of eLeTK.

An important activity within the data workflow is the performance assessment. This activity builds trust into eLeTK and makes users (e.g., administrators, end-users, etc.) confident that the obtained knowledge may be used with confidence and in a safely manner. The first step in assessment is the error analysis, which needs error metrics (e.g., precision, recall, F-Measure, Sum of squared errors, etc.) that are in close relation with the employed data analysis technique. An important activity is the debugging of the machine learning process. This takes care of learning curves, convergence problems and provides a deep insight of the data analysis process such that it is obtained a clear feedback regarding necessary actions that may bring further improvements. The most common actions are to collect more data, try a smaller/larger number of parameters for items representation, avoid over fitting/bias or add polynomial features. Regarding this aspect, plotting learning curves, cost functions or other specific quality indicators may give also important indications regarding what actions may be tried when debugging the data analysis process.

Another important activity that is performed within eLeTK is the evaluation of the obtained model/hypothesis. This is accomplished by using training and testing sets in a 10-fold cross validation scheme. A model selection procedure may be also created by using a validation set and by computing validation errors for each challenger model that is taken into consideration.

4.2. *Data Analysis Workflow Design*

The data analysis workflow (DAW) design meets certain requirements that make sure the business goals of eLeTK are fulfilled. The DAW implements a simple data pipeline model. This means that creating DAWs from the end user perspective is an activity that takes care of setting up raw source data, specification of the preprocessing activities, algorithmic pipeline, model representation, validation methodology, output representation and visualization. Each DAW has the goal of minimizing data relocations by having a preliminary estimation of job size in time and memory. An important part of the workflow is represented by the scheduler system that specifies the frequency by which a DAW is executed.

The data analysis workflow is a pipeline which has the following main layers/components: data layer, loader layer, modelers, performance assessors, viewers, fault handlers and configuration management.

The design of the DAW is created such that end-user simplicity and operational are ensured. This issue mainly implemented by custom wizards that allow quick and effective DAW creation.

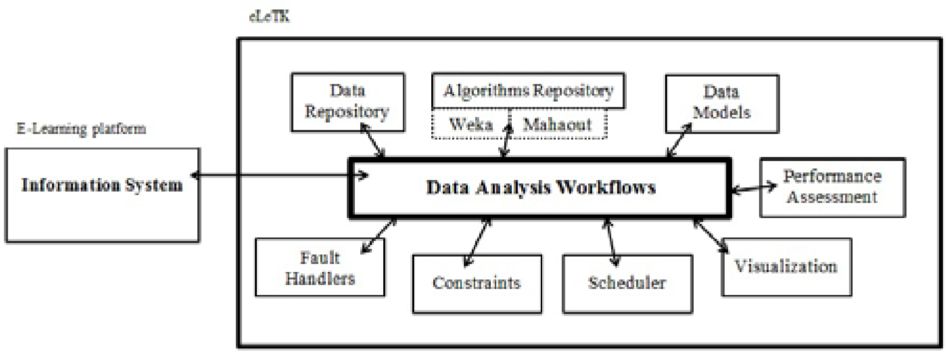

Fig. 1. eLeTK System's Architecture Overview

The Data Repository represents the place where all raw data collected from the information system is collected. The Algorithms Repository is represented by Weka or other library that implements necessary algorithm or extensions of already implemented algorithms. The Data Models manages the obtained knowledge. It may be represented by a set of clusters, rules or classifiers. The Performance Assessment module manages different schemes implementing various errors metrics. The implemented error metrics may be used in conjunction with corresponding algorithms that are managed by the Algorithms Repository. The Fault Handlers implement the verifications regarding correct data integration, data availability and data integrity. The Constraints module manages the specific context variables in which a workflow runs. The constraints regard quality metrics thresholds, number of clusters, rules selection strategies expressed at highest level available for eLeTK users. The Scheduler module is responsible for the way each DAW runs. Setting up a scheduled job is the last activity that needs to be performed after the data analysis pipeline is set up for a new DAW. A scheduled job may be run at request or at certain intervals of time. Visualization module is available only for development and debugging purposes.

4.3. *Detailed Data Workflow in eLeTK*

The software architecture of eLeTK follows the above presented general system specifications. It ensures the proper integration of different modules is such a way that development (e.g. adding new data models or new fault handlers) is performed in a productive manner.

Another property of the software architecture is modifiability. This means that steps that make up a DAW can be added/edited/deleted in a reasonable way.

The software architecture integrates a logging mechanism such that error analysis and debugging may be performed with rapid discovery of faults.

There are two types of users for the eLeTK system. One is represented by the system administrators. Their main job is to set up the DAWs by specifying raw data sources, data modelers, performance assessors, fault handlers and a schedule. The administrators need to be experienced data analysts such that created DAWs have a good structure, be reliable and provide usable and high quality knowledge. The second type of users is represented by end-users. They are represented by educational experts that act on behalf of on-line educational system (e.g., instructors, professors, course managers, teaching assistants etc.) and they are regarded as end-users. From this perspective they represent the main beneficiaries of the eLeTK system.

Within the software architecture there is implemented a Configuration Manager which is responsible adding/editing/deleting raw data sources, available algorithms and data models, current constraints, etc. The Configuration manager is also responsible for checking the health of existing DAWs and launch reliable fault recovery processes that make sure that data consistency is preserved. In this context, a performance watchdog is very necessary tool in order to obtain quick information regarding failures and heavy data workloads.

At client side there is an applet whose goal is to provide interaction with the involved parties: students, professors, managers, etc. The main goal of the applet is to communicate with the server application in a client-server manner. For example, if the applet is to be used by a student it sends the id of the student along with other needed data (i.e., command, parameters, etc.) and receives the decision tree containing the current students and the class of the student. From functional point of view, the applet displays the current model (i.e. the decision tree) and the items placed in leaf nodes. From the data management point of view it represents a merge between the data model and the instances (i.e., students, materials, etc.) accommodated within the model.

In this architecture, the current students are the ones which are enrolled in the current year of study. The *repositoryOfStudents.xml* contains the former students who performed their activities within the on-line educational environment. This data file contains the experiences against which the current students are classified.

On the server side there are two data repositories. The *repositoryOfStudents.xml* is used for building the data model (i.e. the decision tree) and *currentStudents.xml* containing the so far performed activities by the current users.

The below figure represents a detailed data workflow that may be used for classification purposes.

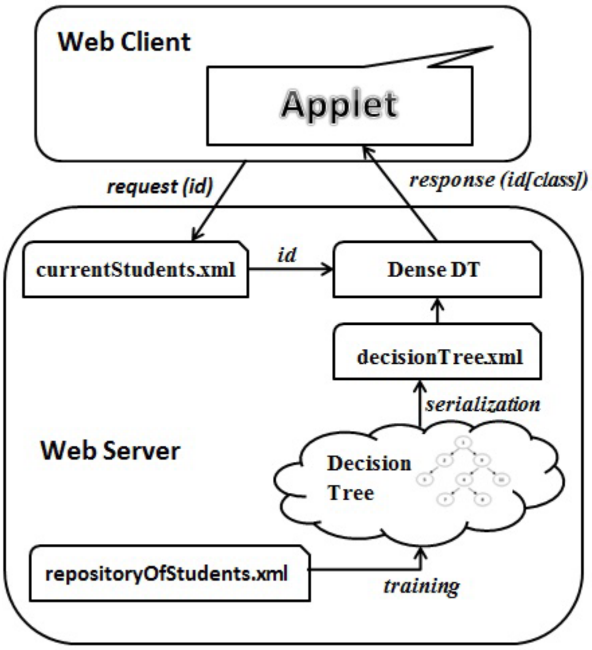

Fig. 2. Detailed Data Wokflow in eLeTK System's Architecture

5. Sample Setup and Usage Scenario

The first step is to perform eLeTK setup. The final goal of a setup procedure is to create a valid DAW. This requires a minimum configuration of a raw data source, a data model, a constraints setup and a set of visualization rules.

The raw data source may be the database of the information system. If this is the case, the credentials of the database (e.g., URL, username, password) must be provided and a scheduled job may be defined to bring the records into the Data Repository in a structured way (e.g., XML). Another scheduled is to create a data file according with the format required by the algorithm and used implementation. Once the training and testing data is available the data model may be created.

One usage scenario of a classification model may be used for determining colleagues (i.e. students from *currentStudents.xml*) that may serve as tutors for a certain student. This usage scenario implies the existence of the following mechanisms. A model building mechanism that takes the activity data from *repositoryOfStudents.xml* file and builds the data model. The obtained model needs to be serialized to an xml file for further use (e.g., display) at client side. Another important mechanism is the classification of a student against in the current data model. This mechanism obtains the *actual class* for the student which needs a tutor. Finally, there is needed mechanism for determining the *target class* for the student. The target class will contain students that are most likely to have enough knowledge for becoming tutors.

In the next sample setup is supposed that a student that is classified as *low* or *average* will be advised to request for advice tutors that were classified as *average* respectively *high*.

Here is a sample *arff* file that may be used by algorithms implemented in Weka workbench.

```
@relation activity
@attribute avgTimeSpent numeric
@attribute correctPercentOfAnswers numeric
@attribute class {low, average, high}
@data
50,65,low
10,25,low
100,65,average
250,95,high
…
```

The data model may be represented by a classifier. Thus, an in memory classifier is available for classifying new data. Some of the most common classification algorithms implemented by Weka are Decision Trees (e.g., ID3, J48, CART), Naïve Bayes or even advanced algorithms such as Support Vector Machines.

The constraints setup contains the thresholds for specific error metrics (e.g., precision, recall, FMeasure, etc.) that allow a reliable classification. In this way eLeTK provides also an indication regarding the confidence in obtained results. The constraints may also be specific to information system users such as students or professors. For example, a student may require the classification of available resources such that his target class is high. Another example, from the perspective of professors is to set up a higher level for FMeasure value regarding the classification of students that may start studying the next chapter. In this way, the effectiveness of the e-Learning system is increased.

Here is a sample decision tree obtained from the above sample dataset.

```
J48 pruned tree
------------------
correctPercentOfAnswers <= 70
|   avgTimeSpent <= 50: low (3.0)
|   avgTimeSpent > 50: average (7.0)
correctPercentOfAnswers > 70: high (6.0)
Number of Leaves  :     3
Size of the tree :          5
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances         14              87.5    %
Incorrectly Classified Instances        2              12.5    %
Kappa statistic                       0.7987
Mean absolute error                    0.0833
Root mean squared error                0.2887
Relative absolute error               18.9356 %
Root relative squared error            60.602  %
```

The above presented decision tree represents the text format for the in-memory data model which may be used when needed to classify a new student. This model is serialized to an xml file which is thereafter used to build a DOM tree containing the actual instances from *currentStdents.xml* file.

The visualization rules provide the real and final expression of what is behind a certain assignment of an item into a certain class. Taking into consideration the above scenario one output of the visualization module may have the following structure:

Dear Student ***John Doe***,

The eLeTK classified you as ***average*** and advices you as tutors the following colleagues:

Samuel Hart
Dick Hartley
Bert Dickens

In a similar way, a classification of students may offer important feedback to professors. In a second scenario, for example, a professor may query the current situation of enrolled students and discover that here are students which spent lots amounts of time in certain activities (e.g., study and/or have quiz activity only regarding some concepts) although their classification is still bad. In this situation, eLeTK provides important feedback regarding an classes of students for which custom messages may be delivered. A sample feedback for the professor may have the following form:

Dear Professor **John Doe**,
The following students: **Samuel Hart, Dick Hartley, Bert Dickens** are in class ***average*** and may need assistance regarding the following concepts:
Graph representations
Connectivity

This is the way in which DAW implements an advanced messaging system that may be used by professors in an on-line educational environment.

In this manner, there may be created a wide range of Data Analysis Workflows, each of them solving a particular educational data mining problem.

6. Conclusions and Future Work

This paper presents the design of a software system, called eLeTK, whose aim is to run along on-line educational systems. The main purpose of eLeTK is to offer feedback for users (e.g., students or professors) of the on-line educational system in an attempt to enhance the educational proficiency. The intuition of such a system is that it represents a substitute for the knowledge that a real professor acquires in a face-to-face educational systems.

The main advantages of eLeTK is that it has access to all performed activity of students and that the data analysis procedures are hundred percent objective. In classical, face-to-face educational systems the main issue regards the ability of the professor to effectively and objectively assess and guide the student.

The main target of eLeTK is to reach a high level of accuracy and thus give the classical on-line educational system the possibility to act as high quality traditional learning environment.

The software architecture is a modular one and is built around the idea of Data Analysis Workflow. A DAW represents a data processing pipeline which is custom designed to accommodate data coming from on-line educational systems.

From a general perspective, eLeTK may work as a service virtually for any on-line educational system. The main requirements for the on-line educational system are to offer access to activity data, to have a custom setup procedure according with provided data, to create required DAW and to accommodate feedback offered by eLeTK.

The future works regard the development of a quality assessment procedure of the offered feedback. This assessment procedure is supposed to offer valuable feedback that may be used in continuous improvement of eLeTK.

Future works also regard continuous improvements of existing modules by integration of more advanced state of the art algorithms, performance assessment procedures, visualization capabilities of results and other new features.

Future work regards increasing the number of features. So far, the prototype version uses only several general features. The system may be adapted to more fine grained features thus taking into account documents, concepts or even quizzes. As generalization, quizzes may also be assigned to propositions that are crated from existing concepts. On the other hand, other classification algorithms with an adapted system design regarding feature types and values may be designed. Other future works regard the usage of other classification algorithms and usage of even more features, such as number of messages sent/read on the forum and their length.

Acknowledgments

This work was supported by the strategic grant POSDRU/89/1.5/S/61968, Project ID61968 (2009), co-financed by the European Social Fund within the Sectorial Operational Program Human Resources Development 2007 – 2013.

References

Agrawal R., Srikant R (1995). *Mining sequential patterns*, In Eleventh International conference on data engineering (pp. 3–14). Taipei, Taiwan, IEEE Computer Society Press.

Apache Mahout Project (last retrieved March, 2013), http://mahout.apache.org/

Brusilovsky P. and Peylo C. (2003). *Adaptive and intelligent web-based educational systems*, International Journal of Artificial Intelligence in Education, 13, 156–169.

Burdescu D.D, Mihăescu M.C. (2006). *Tesys: e-Learning Application Built on a Web Platform*, International Joint Conference on e-Business and Telecommunications -International Conference on e-Business (ICE-B), pp.:315-318.

Burlea Schiopoiu A., Badica A., Radu C. (2011). *The evolution of e-learning platform TESYS user preferences during the training processes*, In Proceedings of ECEL2011, 11-12 Novembre Brighton, UK, pp.754-761.

Camacho D., Ortigosa A., Pulido E., Moreno M. (2008). *AI techniques for Monitoring Student Learning Process,* In Advances in E-Learning: Experiences and Methodologies. A book edited by Francisco J. Garcia-Peñalbo. Publisher: Information Science Reference-IGI Global, USA. Chapter 9, pp. 149-172.

Fomichov V. and Fomichova O. (1994). *The Theory of Dynamic Conceptual Mappings and its significance for Education, Cognitive Science, and Artificial Intelligence*, Informatica. An Intern. Journal of Computing and Informatics (Slovenia), 18 (2).

Fomichov V. and Fomichova O. (2006). *Cognitonics as a New Science and Its Significance for Informatics and Information Society*, Special Issue on Developing Creativity and Broad Mental Outlook in the Information Society (Guest Editor Vladimir Fomichov), Informatica. (Slovenia), 30 (4).

Grob H., Bensberg F., Kaderali F. (2004). *Controlling open source intermediaries – a web log mining approach*, In Proceedings of the 26th international conference on information technology interfaces pp. 233–242.

Grobelnik M., Mladenic D., Jermol M. (2002). *Exploiting text mining in publishing and education*, In Proceedings of the ICML-2002 workshop on data mining lessons learned (pp. 34–39).

Han J., Kamber M. and Pei J. (2011). *Data Mining: Concepts and Techniques, 3rd ed.*, The Morgan Kaufmann Series in Data Management Systems.

Klosgen W., Zytkow J. (2002). *Handbook of data mining and knowledge discovery*, New York, Oxford University Press.

Koutri M., Avouris N., Daskalaki S. (2005). *A survey on web usage mining techniques for web-based adaptive hypermedia systems*, Adaptable and Adaptive Hypermedia Systems, IRM Press, pp. 125-149, Hershey.

Loh W-Y (2008). *Classication and Regression Tree Methods*, Wiley, 2008.

Lowd D., Domingos P. (2005). *Naive Bayes Models for Probability Estimation*. ICML '05 Proceedings of the 22nd international conference on Machine learning.

Merceron A., Yacef K. (2004). *Mining student data captured from a web-based tutoring tool: Initial exploration and results*, Journal of Interactive Learning Research, 15(4), pp. 319–346.

Novak, J.D. and Cañas, A. J. (2006). *The Theory Underlying Concept Maps and How to Construct and Use Them*, Technical Report IHMC CmapTools.

Paulsen M. (2003). *Online education and learning management systems*, Bekkestua: NKI Forlaget.

Quinlan J.R. (1993). *C4.5: Programs for Machine Learning*, San Mateo, CA.

Romero, C. and Ventura S. (2007). *Educational Data Mining: A Survey from 1995 to 2005*, Expert Systems with Applications, 33(1), pp. 135-146.

Romero C, Ventura S., Bra P.D. (2004). *Knowledge discovery with genetic programming for providing feedback to courseware author*, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, 14(5), pp. 425–464.

Srivastava, J., Cooley, R., Deshpande, M., & Tan, P (2000). *Web usage mining: Discovery and applications of usage patterns from web data*, SIGKDD Explorations, 1(2), 12–23.

Urbancic T., Skrjanc M., Flach P. (2002). *Web-based analysis of data mining and decision support education*, AI Communications, 15, 199–204.

Witten, I. H. & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)*, Morgan Kaufmann.

Zorrilla M. E., Menasalvas E., Marin D., Mora E., Segovia J. (2005). *Web usage mining project for improving web-based learning sites*, In Web mining workshop, Cataluna.