Web process and workflow path mining using the Multimethod approach

Jorge Cardoso*

Department of Mathematics and Engineering,
University of Madeira, 9050 390 Funchal, Portugal
E-mail: jcardoso@uma.pt
*Corresponding author

Mitja Lenič

System Software Laboratory,
University of Maribor, FERI, Smetanova 17,
2000 Maribor, Slovenia
E-mail: mitja.lenic@uni-mb.si

Abstract: Workflow Management Systems (WfMS) provide a fundamental technological infrastructure to define and manage business processes efficiently. WfMS logs contain valuable data that can be used to discover and extract knowledge about the execution of workflows and processes. One piece of important and useful information that can be discovered is related to the prediction of the path that will be followed during the execution of a workflow. We call this type of discovery, path mining. In this paper, we present and describe how path mining can be achieved using different data mining techniques including the Multimethod approach.

Keywords: web processes; workflows; path mining; Multimethod approach; quality of service; business process management systems; data mining.

Reference to this paper should be made as follows: Cardoso, J. and Lenič, M. (2006) 'Web process and workflow path mining using the Multimethod approach', *Int. J. Business Intelligence and Data Mining*, Vol. 1, No. 3, pp.304–328.

Biographical notes: Jorge Cardoso joined the University of Madeira (Portugal) in March 2003. He previously gave lectures at University of Georgia (USA) and at the Instituto Politécnico de Leiria (Portugal). He received his PhD in Computer Science from the University of Georgia in 2002. While at the University of Georgia, he was part of the LSDIS Lab. where he did extensive research on workflow management systems. In 1999, he worked at the Boeing Company. He was the coorganiser and cochair of the First and Second International Workshop on Semantic and Dynamic Web Processes. He has published several refereed papers in the areas of workflow management systems, semantic web, and related fields.

Mitja Lenič's work research interests are concentrated on intelligent systems, programming languages, complex systems, chaos theory, and software quality and metrics. His research results were implemented in SQUFOL project financed by the EU. He has written many technical and research papers published in recognised international journals and major conferences and coauthored various textbooks.

1 Introduction

E-commerce and e-services have been growing at a very fast pace. The web, coupled with e-commerce and e-services, is enabling a new networked economy (Sheth et al., 1999). The scope of activities that processes span has moved from intra-enterprise workflows, predefined inter-enterprise and business-to-business processes, to dynamically defined web processes among cooperating organisations.

Business Process Management Systems (BPMS) provide a fundamental technological infrastructure to define and manage business processes efficiently. A BPMS represents a single, unified modelling, integration, and execution environment that can be applied to the implementation of business processes (Smith and Fingar, 2003). These systems can manage web processes and workflows orchestrating web services and activities respectively.

Currently, organisations use BPMS, such as Workflow Management Systems (WfMS), to support a wide range of distinct applications (CAPA, 1997; Kang et al., 1999; Q-Link, 2002; Anyanwu et al., 2003; Hall et al., 2003; Luo et al., 2003), such as insurance claims, bank loans, bioinformatics experiments, healthcare procedures, telecommunication services, military plans, and school administration. Applications can be more oriented to support or enhance existing processes, to increase competitive advantage, to reduce costs, and also to manage critical infrastructures. According to the Aberdeen Group's estimates, spending in the Business Process Management software sector (which includes workflow systems) reached $2.26 billion in 2001 (Cowley, 2002).

With the emergence of web services, WfMS have become essential to support, manage, and enact web processes, both between companies and within the firm itself (Sheth et al., 1999). WfMS are systems capable of both generating and collecting considerable amounts of data describing the execution of business processes.

The data generated from the execution of processes are rich with concealed information that can be used for making intelligent business decisions (Agrawal et al., 1998; Weijters and van der Aalst, 2001; Grigoria et al., 2004). This data is stored in a process log system that records all of the events for all of the processes being executed by the enactment service. Therefore, organisations are increasingly placing emphasis on category of knowledge as the key to providing them with a competitive edge and supporting the strategic decision making process.

One important and useful piece of knowledge to discover and extract from BPMS logs is the set of implicit rules that govern the path of web services or workflow activities followed during the execution of a process. We call this discovery, path mining (Cardoso, 2005). Path mining is vital to systems, applications, and algorithms that need to carry out path prediction procedures to processes. It allows for intelligent analysis by using data mining algorithms on BPMS logs. The knowledge obtained allows the understanding of the causes of specific behaviours, such as the execution of certain paths in a process instance. The crucial goal of path mining is prediction – and predictive data mining is the most common type of data mining and one that has the most direct business applications. Path mining generates a number of scientific and practical challenges. For example, which paths can be discovered and how much data is needed to provide useful information.

In web processes and workflows for e-commerce applications, suppliers and customers define a binding agreement or contract between the parties involved, specifying Quality of Service (QoS) items such as products or services to be delivered, deadlines, quality of products, and cost of services (Cardoso et al., 2002, 2004; Gillmann et al., 2002). The management of QoS metrics directly impacts the success of organisations participating in e-commerce. Therefore, when services or products are created or managed using web processes and workflows, the underlying management engine must accept the specifications and be able to estimate, monitor, and control the QoS rendered to customers.

The concept of path mining can be used effectively in many business applications to estimate the QoS of web processes and workflows (Cardoso et al., 2004) since the estimation requires the prediction of paths. Path mining allows better and more accurate predictive algorithms to be built, making it possible to compute the QoS for processes automatically based on the QoS of atomic activities.

This paper presents a very systematic way to achieve process path mining, illustrating the fundamental concepts of path mining in practice. We apply different data mining techniques, including the Multimethod approach (Lenic and Kokol, 2002), J48 (Weka, 2004), Naïve Bayes, Sequential Minimal Optimisation (Platt, 1999), and MultiBoost (Webb, 2000) to conveniently extract knowledge related to paths. The experimental results presented in this paper are very encouraging and open a new door for business process intelligence (BPI) (Grigoria et al., 2004) and process mining research.

This paper is structured as follows. Section 2 presents background information about business processes, workflows, and WfMS to bring in some basic knowledge to readers. Section 3 introduces the concept of process path mining and described the Multimethod approach. Section 4 discusses web processes and their relationship with workflows. We describe the basic elements that compose a process and list the most prominent process specification languages. We illustrate a scenario that will be used throughout this paper to exemplify the applicability of path mining. In Section 5 we present the procedure and the main steps involved in process path mining. Section 6 describes the fundamental aspects of the Multimethod approach and what makes it unique compared to other data mining methods. Section 7 presents the experiments that have been conducted and the results obtained. Section 8 presents the related work that has been made in the research area of process mining, process path mining, and data mining. Finally, Section 9 presents our conclusions.

2 Business processes, workflows, and WfMS

A business process is a set of activities that represent all the alternative methods of performing the work needed to achieve a business objective. The set of structured activities produce a specific service or product for a particular customer or customers. An activity, or task, is a specific action carried out in the context of a business process that occurs over time and that has recognisable results. Activities take one or more kinds of input and create an output that is of value to the customer. Activities use resources to produce products and services.

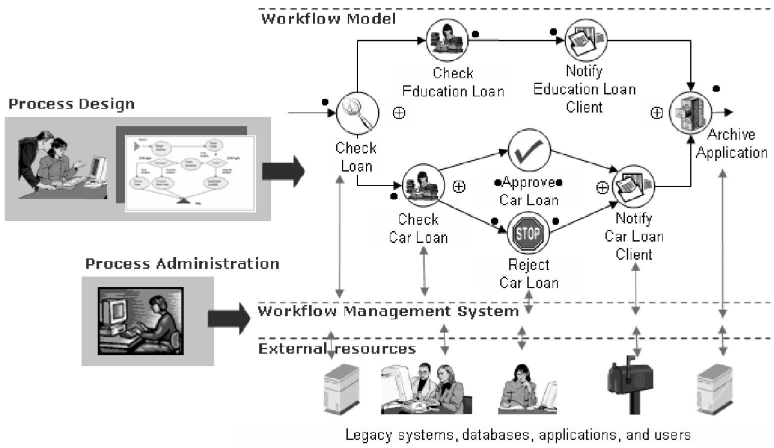
Business processes exist in all the departments of an organisation, such as human resources, finance, legal, engineering and operations, sales and marketing, and customer service department. Examples of business processes include loan application evaluation, travel expense report editing, and insurance claim processing.

Workflows and Workflow Management Systems provide an execution environment for controlling, monitoring, and deploying the code necessary to automate a business process. Workflows are the automation of business processes and they can provide the following benefits to an organisation:

- real-time business performance monitoring
- remote process management
- enhanced performance and reduce bottlenecks
- cuts operations costs
- reduce paper
- increases customer satisfaction.

A workflow is implemented in accordance with a business process specification and execution paradigm. Figure 1 shows the main components of a WfMS environment. Under a WfMS, a workflow model is first created to specify organisational business processes, and then workflow instances are created to carry out the actual steps described in the workflow model.

Figure 1 A workflow and its integration with external resources



During the execution, the workflow instances can access legacy systems, databases, applications, and can interact with users. A workflow can automatically start processes without user participation and make intelligent decisions based on information captured.

Workflow systems have been installed and deployed successfully in a wide spectrum of organisations. Most workflow management systems, both products and research prototypes, are rather monolithic and aim at providing fully-fledged support for the widest possible application spectrum. The same WfMS can be deployed in various domains, such as bio-informatics, healthcare, telecommunications, military, and school administration.

3 Process path mining and the Multimethod approach

3.1 Introduction to process path mining

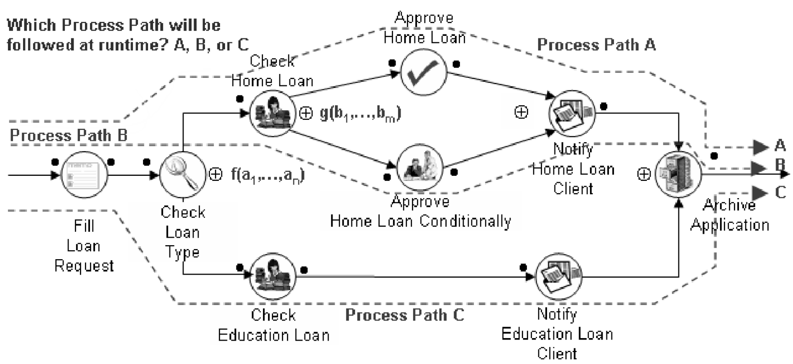
A process definition is used to specify the activities (i.e., workflow activities of web services) to be executed and the order in which this is supposed to be done. For that, conditions are usually specified in the definition, which correspond to causal dependencies between the activities. The WfMC (2002) identifies the following four basic types of relationship between the activities:

- *sequential*: several activities are executed in sequence
- *parallel*: two or more activity instances are executed in parallel
- *conditional*: one branch is selected and taken from multiple alternative workflow branches
- *iteration*: a workflow activity cycle involving the repetitive execution of one or more workflow activities.

During the execution of workflow processes, logical expressions may be evaluated by the workflow engine to decide the sequence of activity execution within a process. These expressions are evaluated against a set of workflow variables, which are specified in the workflow definition and whose values can be altered during execution of the workflow. Figure 2 shows two logical expressions, namely $f(a_1, \ldots, a_n)$ and $g(b_1, \ldots, b_m)$. These expressions are commonly referred to as transition conditions.

Process path mining is the procedure to predict the sequence of activity, within a process, that will be executed at runtime by the workflow engine. The sequence of activity if directly related to the workflow variables. For example, let us consider the process illustrated in the Figure 2. During the execution of this process three different paths can be followed to complete the process: path A, path B, and path C. Predicting the path that will be followed is of an enormous importance for business process management systems. If a path can be correctly predicted, it is possible to predict the time a process will need to be completed, the cost for executing a process, and to predict the reliability (Cardoso et al., 2004) of a process (since some paths may be more likely to contain activities that will fail at runtime).

Figure 2 Process path prediction



3.2 The Multimethod approach

To carry out path mining two main techniques can be used: stochastic analysis and data mining. Both techniques use the data stored in a BPMS log. These data is generated by the BPMS enactment service when executing process instances.

Stochastic analysis involves determining the probabilities of a specific path of a process (i.e., a sequence of activities) to be followed at runtime. This technique can be explored using the SWR algorithm that we have previously developed (Cardoso, 2002). Data mining can also be used. In this paper we explore the use of data mining techniques to predict the path of a process that will be followed at runtime. The data mining algorithms that we will study include J48 (an implementation of C4.5), Naïve Bayes, Sequential Minimal Optimisation, and Multiboost. This last algorithm has been developed by one of the authors of this paper and proves to outperform the other algorithms based on single methods.

4 Web processes and workflows

Workflows have been mainly known for their implementation and deployment within organisational boundaries. Web processes can be viewed as an evolution of workflows since they enable application integration within and across organisational boundaries. Therefore, workflows can be made available as web processes.

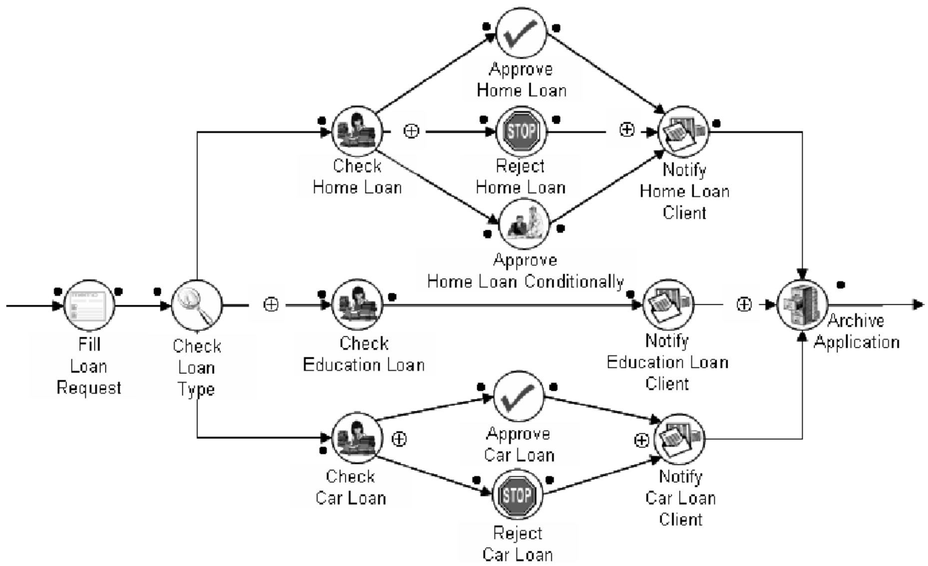
Web processes and workflows represent the automation of a business process, in whole or part, during which documents, information or activities are passed from one participant to another, according to a set of rules, and are made up of elements which comprise transitions, logic conditions, data flows, parallel and conditional building blocks, starting and ending points, splits, and joins. Web processes can be viewed as workflows that manage web services instead of tasks (Cardoso and Sheth, 2003) and these workflow management systems can also be used to enact web processes. For example, a web process can be invoked by an outside vendor submitting a request for a price quote. In this scenario, a web process can control the flow of a set of web services that make up an application. The correspondence between web processes and workflows is extremely important because it allows the use of earlier studies, research, and development in the area of workflow management systems to be applied to web processes (Cardoso et al., 2004). The similarity of web processes and workflows allows us to conclude that the path mining technique described in this paper can be applied to both types of processes. Hereafter, we will use the term 'process' to refer to web processes and workflows. We will also use the terminology defined by the WfMC (the Workflow Management Coalition (WfMC, 2002)).

4.1 Processes' basic elements

A process is composed of activities (Web services or tasks) and transitions. Activities are represented using circles and transitions are represented using arrows. Transitions express dependencies between activities. Activities with more than one outgoing transition can be classified as an and-split or xor-split. And-split web services enable all their outgoing transitions after completing their execution. Xor-split activities enable only one outgoing transition after completing their execution. And-split activities are represented with a '•'

and xor-split activities are represented with a '⊕'. An activity with more than one incoming transition can be classified as an and-join or xor-join. And-join activities start their execution when all their incoming transitions are enabled. Xor-join activities are executed as soon as one of the incoming transitions is enabled. As with and-split and xor-split activities, and-join activities and xor-join activities are represented with the symbol '•' and '⊕', respectively. An example of a process is shown in Figure 4.

Figure 4 The loan process



4.2 Business process scenario

In this section, we describe a scenario that will be used throughout the paper to explain and illustrate the use of data mining techniques to predict path executions.

A major bank has realised that to be competitive and efficient it must adopt a new and modern information system infrastructure. Therefore, a first step was taken in that direction with the adoption of a workflow management system to support its business processes. Since the bank supplies several services to its customers, the adoption of a WfMS has enabled the logic of bank processes to be captured in web processes schema. As a result, all the services available to customers are stored and executed under the supervision of the workflow system. One of the services supplied by the bank is the loan process depicted in Figure 1.

The web process is composed of fourteen web services. The Fill Loan Request Web service allows clients to request a loan from the bank. In this step, the client is asked to fill in an electronic form with personal information and data describing the condition of the loan being requested.

The second web service, Check Loan Type, determines the type of loan a client has requested and, based on the type, forwards the request to one of three web services: Check Home Loan, Check Educational Home Loan, or Check Car Loan.

A loan request can be either accepted or rejected. In the case of a home loan, however, the loan can also be approved conditionally. The web services in charge of accepting a particular type of loan are Approve Home Loan, Approve Educational Loan, and Approve Car Loan. The web services responsible for rejecting a loan are Reject Car Loan, Reject Educational Loan, and Reject Car Loan. The web service Approve Home Loan Conditionally, as the name suggests, approves a home loan under a set of conditions.

When the result of a loan application is known, it is e-mailed to the client. Finally, the Archive Application web service creates a report and stores the loan application data in a database record.

A complete explanation of the process is described in Cardoso (2005).

4.3 Process quality of service

One important missing requirement for workflow systems and BPMS is the management of QoS. Organisations operating in modern markets, such as e-commerce activities and distributed services interactions require QoS management. Appropriate control of quality leads to the creation of quality products and services; these, in turn, fulfill customer expectations and achieve customer satisfaction.

4.3.1 QoS model

Quality of service can be characterised according to various dimensions. We have investigated related work to decide which dimensions would be relevant to compose our QoS model. Our research targeted two distinct areas: operations management for organisations and quality of service for software systems. The study of these two areas is important, since BPMS are widely used to model organisational business processes, and workflow systems are themselves software systems.

Based on previous studies and experience in the business process and workflow domain, a QoS model composed of the following dimensions has been constructed (Cardoso et al., 2004): *time*, *cost*, and *reliability*. QoS specifications are set for activity definitions. Based on this information, QoS metrics are computed for processes.

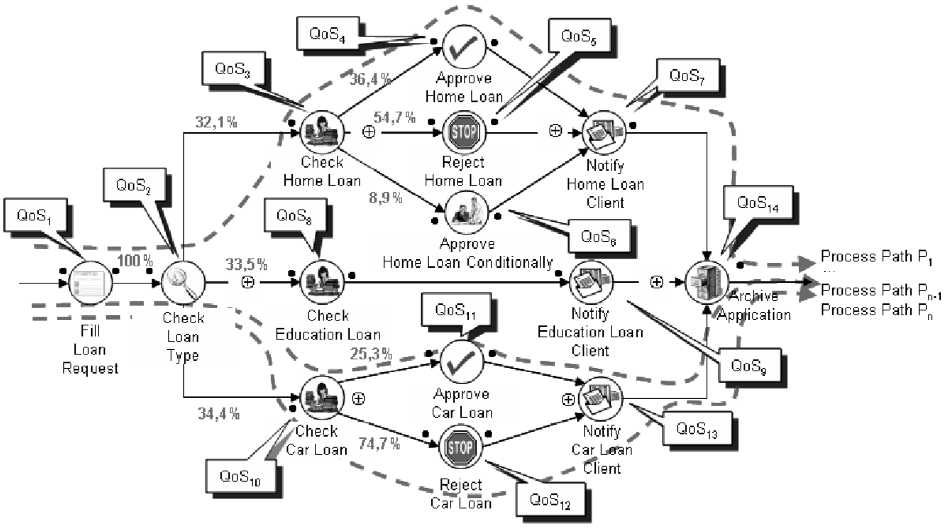
4.3.2 Creation of QoS estimates

In order to allow the analysis and computation of process QoS, it is necessary to initialise activities QoS metrics and also initialise stochastic information which indicates the probability of transitions being fired at runtime. Once activities and transitions have their estimates set, algorithms (e.g., the SWR Cardoso, 2002; Cardoso et al., 2004; algorithm) and mechanisms, such as simulation (Chandrasekaran et al., 2002; Miller et al., 2002), can be applied to compute overall process QoS.

- *Creation of QoS estimates for activities*. The specification of QoS metrics for activities is made at design time and re-computed at runtime, when activities are executed. During the graphical construction of a process, the business analyst and domain expert set QoS estimates for each activity. The estimates characterise the quality of service that the activities will exhibit at runtime (see Figure 2).

- *Creation of probabilities estimates for transitions*. In the same way we seed activities' QoS, we also need to seed process transitions. Initially, the designer sets the transition probabilities at design time. At runtime, the transitions' probabilities are re-computed. The method used to re-compute the transitions' probabilities follows the same lines of the method used to re-compute activities' QoS. The creation of probabilities estimates for transitions, to subsequently estimate processes' QoS, is based on a simple statistical technique (Cardoso, 2002; Cardoso et al., 2004). Figure 5 shows some of the paths that can be followed at runtime during the execution of a process. Each path has enabling probabilities associated with the transitions. We will see that the use of adequate data mining techniques yields better results for path mining compared to a simple statistical approach.

Figure 5 The loan process with QoS estimates for activities and probabilities estimates for transitions



5 Process path mining

The material presented in this section emphasises the use of data mining concepts and techniques for uncovering interesting process patterns hidden in large process logs. Our path mining technique is composed of three steps. In the first step, a process log is constructed. The process log structure must store activities' input and output parameters and the path followed during the execution of process instances. In the second step, we construct a process profile. This profile is composed of a set of attributes that will be analysed to establish a relationship between the attributes and the web services executed (i.e., the path followed during the execution of process instances). Finally, in the third step, we use data mining methods to determine the paths followed based on process profiles.

The method presented in the next sections is more suitable for administrative and production processes (McCready, 1992) compared to Ad-hoc and collaborative processes, since they are more repetitive and predictive.

5.1 Process log

During the execution of processes, events and messages generated by the enactment system are stored in a process log. Types of events that occur during process executions include the start and completion of each activity, the resource utilised, and any failure that occurred during activity or process execution. Typically, log systems are implemented using relational databases, transactional databases, or flat files. These data stores provide an adequate format on which path mining can be performed. The data includes real-time information describing the execution and behaviour of processes, web services, instances, transitions, and other elements such as runtime QoS metrics.

To perform path mining, current process logs need to be extended to store information indicating the values and the type of the input parameters passed to activities and the output parameters received from activities. Table 1 shows an extended process log which accommodates input/output values of activities parameters that have been generated at runtime. Each 'Parameter/Value' entry as a type, a parameter name, and a value (for example, string loan-type = 'car-loan').

Additionally, the process log needs to include path information: a path describing the activities that have been executed during the enactment of a process. This information can easily be stored in the log. For example, an extra field can be added to the log system to contain the information indicating the path followed. The path needs only to be associated with the entry corresponding to the last service of a process to be executed. For example, in the process log illustrated in Table 2, the service NotifyUser is the last activity of a process. The log has been extended in such a way that the NotifyUser record contains information about the path followed during the process execution.

5.2 Process profile

When beginning work on path mining, it is necessary to elaborate a profile for each process. A profile provides the input to machine learning and it is characterised by its values on a fixed, predefined set of attributes. The attributes correspond to the activity input/output parameters that have been stored previously in the process log. Path mining will be performed on these attributes.

The concept of profile has been exploited in other research areas. For example, operational profiles (Musa, 1993) have been used to test the reliability of programs. The idea is to test a program based on specific inputs. This can be achieved by the elaboration of an operational profile (Musa, 1999). The input space is partitioned into domains, and each input is associated with a probability of being selected during operational use. The probability is employed in the input domain to guide input generation. At runtime, programs have a probability associated with each input.

Table 1 Extended process log

		Conventional process log			*Process log extension*	
…	Process instance	Activity	Activity instance	…	Parameter/value	Path
…	LA04	RejectCarLoan	RCL03	…	int LoanNum = 14357; string loan-type = 'car-loan'	…
…	LA04	NotifyCLoanClient	NLC07	…	string e-mail = 'jf@uma.pt'	…
…	TR08	FillRequestTravel	FRT03	…	string City = 'Atlanta'; string Country = 'USA'; long BudgetCode = 193432	…
…	LA05	CheckLoanRequest	CLR05	…	double income = 12000; string Name = 'Eibe Frank'	…
…	TR09	NotifyUser	NU07	…	String e-mail = jf@uma.pt; String telef = '35129170023'	FillForm->CheckForm->Approve->Sign->Report
…	…	…	…	…	…	…

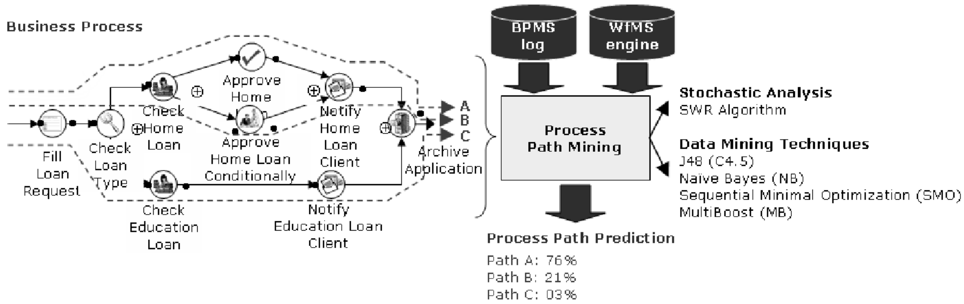
Table 2 Process instance profile and process path class

Process instance profile						*Class*
income	Loan_type	loan_amount	loan_years	Name	SSN	Path

5.3 Output of process path mining

The output of applying process path mining to a WfMS log and to a process in execution is a structure that predicts the probability that a given path will be followed during the remaining execution of the process. Figure 2 illustrates a very simple process for which three different paths can be followed at runtime, A, B, and C. Using path mining we can predict in an early execution of the process which path that will be most likely followed during the execution of the process. As an example, the path mining approach can indicate that path A will be most likely followed, with an accuracy of 76% (see Figure 3).

Figure 3 Process path mining



6 The Multimethod approach

The attributes present in a process profile trigger the execution of a specific set of activities. Therefore, for each profile previously constructed, we associate an additional attribute, the path attribute, indicating the path followed when the attributes of the profile have been assigned to specific values. In data mining, the path attribute is considered to be a class.

Once the profiles and a path attribute value for each profile have been determined, we can use data mining methods to establish a relationship between the profiles and the paths followed at runtime. A learning schema takes a set of classified profiles from which it is expected to learn a way of classifying unseen profiles.

In this section we explain in detailed the Multimethod approach since it will be used in our experiments and it is new and different from more traditional approaches.

6.1 Introduction

When we observe some event or when collecting data for analysis, we assume that there is some target concept present in the data. It has been always a dream of computer scientists to produce intelligent learning machine methods that could learn from past experience and recognise target concepts from observations. Of course we are still far from achieving this goal since we still don't fully understand the human learning process.

Despite this fact, many successful approaches have been developed in the field of data mining. One important component of data mining is machine learning, which is used to automatically extract knowledge from sampled and preprocessed data. The extracted knowledge is then post-processed to derive rules and patterns.

In the short history of data mining and machine learning many approaches that use different techniques have evolved. These approaches are based on the different interpretation of learning and exploit different laws to extract knowledge from data. The main differences are related to the way the learning process is implemented and to the way the extracted knowledge is represented. For example, neural networks attempt to model the human brain and in practice can produce highly accurate knowledge models. On other hand, the knowledge representation used, which involve connections between neurons, is hard to understand.

With the variety of existing domains it is almost impossible to develop a single machine learning algorithm that would work well in all the environments. If you consider single data mining methods it becomes clear that there is no clear winner (Wolpert and Macready, 1997), since learning algorithms and knowledge representations have an important impact on the performance of algorithms depending on the domain and application for which they are being used. Additionally, by considering the 'no free lunch theorem' we realise that there is no single method that would outperform all other methods in all the domains. To surpass this difficulty, we have constructed a new data mining technique called the Multimethod approach. To clarify the concepts behind the Multimethod approach we start by giving a short introduction of single and hybrid approaches.

6.2 Single method approaches

Single method approaches use a single knowledge representation to structure the discovered/extracted knowledge from data. The knowledge representation has a dramatic impact on the learning capabilities of a method. For example, if there is a target concept in the data that cannot be described by a knowledge representation it is impossible for a machine learning algorithm to find the target concept. On the other hand, by introducing a very complex knowledge representation, the number of possible solutions and the search space for the model (hypothesis) that describes the target concept is increased. Each knowledge representation, and thereby each method, has advantages and inherent limitations.

Decision trees (Quinlan, 1993), for example, are easily understandable to humans and can be constructed and used without the help of a computer, but they are not appropriate for discovering complex nonlinear concepts. Most connectivistics approaches that model cognitive abilities of the brain can extract complex relations, but unfortunately the knowledge representation is not easily understandable to humans and, therefore, they cannot be directly used for data mining. Evolutionary approaches to knowledge extraction pose an alternative, because they are not inherently limited to local solutions (Goldberg, 1989), but are computationally expensive and do not always guarantee good results.

There are many different approaches to knowledge representation, such as rules, rough-sets, case based reasoning, support vector machines, and different fuzzy methodologies. Unfortunately, they all subvert to some of the mentioned limitations.

6.3 Hybrid approaches

Hybrid approaches are based on the assumption that the synergetic combination of single models can produced a more powerful data mining approach, i.e., 'the whole is greater than the sum of the parts'. Each of the single methods has its advantages but also inherent limitations and disadvantages, which must be taken into account when using a particular method. Therefore, a sound solution is to combine one or more methods to overcome the disadvantages and limitations of a single method.

Hybrid systems are generally static and cannot change the structure of the methods they use. To be able to use hybrids with different knowledge representations, it is commonly required to transform one method knowledge model into another. Some transformations can be trivial, especially when converting knowledge from symbolic approaches. When the knowledge is not clearly represented, such as with neural networks (Todorovski and Dzeroski, 2000; McGarry et al., 2001), it is very likely that some concept elements may be lost during transformation because of the different hypothesis spaces in the knowledge representations. The Multimethod approach presented in the next section addresses precisely this problem.

6.4 The Multimethod approach

To address the problems previously described we have adopted and followed a set of established approaches from other research fields, especially from social sciences, that select and apply specific approaches based on the nature of the problem. We went a step forward with the idea that machine learning should dynamically create a hybrid approach that is the most appropriate for a particular problem. The Multimethod approach was introduced in Lenic and Kokol (2002). The goal was to achieve an improvement of the quality of knowledge extraction that is not inherently limited to sub-optimal solutions. The idea of combining different approaches had already been proposed (Dietterich, 2000), but all attempts to combine different methods have use a loose coupling approach. The most probable explanation for this fact is the ease of implementation. While modularity represents an important aspect of the combination of methods into hybrids, the main disadvantage of loose coupling is that methods work almost independently of each other and, therefore, it is difficult to make them work as a synergetic team.

As opposed to the conventional hybrid methods described in the previous subsection, the idea of the Multimethod approach is to dynamically combine and apply different methods with no predefined order to the same problem. The methods are applied not only in sequence, but also at different crosscut points of the internal knowledge representation. One solution can contain many knowledge representations that are hopefully the most appropriate for presenting a solution for a specified problem.

The key idea of the Multimethod approach is to incrementally evolve and improve knowledge by using different methods. The approach finds a way to enable dynamic combination of methods to the somehow quasi-unified knowledge representation to enable knowledge sharing between methods and not to set a limit to specific knowledge representation.

As a result, we can get multiple solutions, like with evolutionary algorithms, where each solution is obtained using the application of different methods with a different set of parameters. Hence, the concept of population is introduced, which is composed of individuals/solutions that have the common goal to improve their classification abilities on a given environment/problem. The existence of an abstract view makes it possible to enable the coexistence of different knowledge representations in the same population. With this approach, the solution space and expressive power is dramatically increased. This is also the main drawback of the approach, since the increased search space requires more computing power to find an optimal solution.

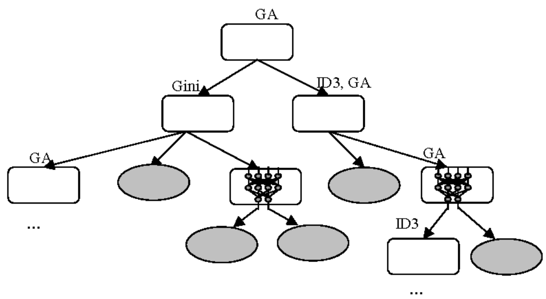
To be able to reuse and improve knowledge using another method that produced it, it is usually necessary to transform knowledge into a different form compatible with the application method. The conversion between knowledge representations (for example, between neural networks and decision trees) can be done using two different approaches: (1) convert one knowledge representation into another using different already known methods or (2) combine both knowledge representations into a single one using a meta approach. In both cases, knowledge transmutation is executed (Wolpert and Macready, 1997).

In the first case, the conversion between different knowledge representations must be implemented, which is usually not perfect and some parts of the knowledge may be lost. But it can provide a different view representing a good starting point in the solution search space.

The second approach, which is based on combining knowledge, requires some crosscut-points where knowledge representations can be merged. For example, in decision trees such crosscut-points are internal nodes, where the condition in an internal node can be replaced by another intelligent system (for example, support vector machine – SVM). The same idea can be also applied to decision leafs (Figure 6).

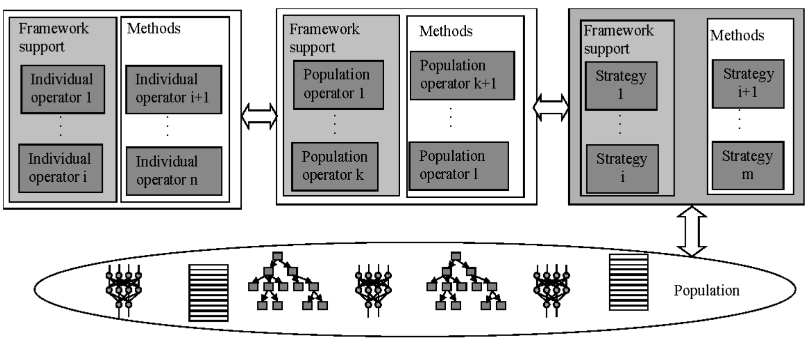
The transformation is usually time consuming and does not necessarily transform all aspects of the source knowledge representation, because of different hypothesis spaces and expressional capabilities of the target knowledge representation. To reduce the transformation cost and information lost during the knowledge transformation some common knowledge representation models have been standardised to support the applicability of different methods to individuals. Of course, each method of implementation can use its own internal knowledge representation that is not compatible with other methods. As an example, we present the WEKA (Weka, 2004) system that uses different knowledge representations but has no operations for transforming knowledge to make it available to other method and the available methods are not designed to incrementally improve extracted knowledge. In this case, the loose coupling approach is used and additional knowledge transformations are needed to make it available to other methods.

Figure 6 An example of a decision tree induced using the Multimethod approach. Each node is induced with an appropriate method (genetic algorithm, ID3, Gini, Chi-square, J-measure, SVM, neural network, etc.)



Using the idea of the Multimethod approach, we designed a framework that operates on a population of extracted knowledge representations – individuals. Since methods are usually composed of operations that can be reused in other methods we view methods on the basis of operators. Therefore, we introduced the operation on an individual as a function that transforms one or more individuals to a single individual. Operations can be part of one or more methods, such as the pruning operator, the boosting operator, etc. This approach provides the ability to simply add new operations to the framework (Figure 7).

Figure 7 Multimethod framework

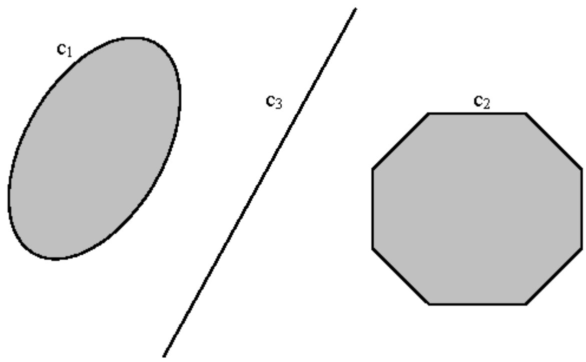


The representation with individual operations facilitates an effective and modular way to represent the results as a single individual, but in general the results of the operations can also be a population of individuals (for example, the mutation operation in evolutionary algorithm is defined both on the individual and the population level). Population operators can be generalised with higher order functions and thereby reused in different methods.

The Multimethod approach uses methods with different learning techniques. For example, heuristic approaches produce a good starting point in the solution space and can also represent local optima. This approach can be combined with search space strategies that are not limited to local optima and can improve the results from heuristic approaches. Additionally, heuristics can be used on solution search space approaches to converge to a different local optima that might represent a better solution than previous ones. Another important benefit is that using multiple methods we can produce multiple solutions with different perspectives to the initial problem.

Another benefit of combining two methods using a third classifier is the separation of concepts. For example, in Figure 8 there are two concepts, c_1 and c_2, that can be separated using an existing set of methods. Let us suppose that there is no available method that is able to acquire both concepts in a single knowledge representation. In such a case, we need a separation of the problem using another concept, c_3. Please note that in the scope of the domain problem, this additional concept has no special meaning to induce a successful composite concept.

Figure 8 Concept separation using GA



The Multimethod approach uses populations and evolutionary based techniques to search for solutions. The initial population of extracted knowledge is generated using different methods with a different set of attributes. The starting points in our solution search space are represented by individual methods. This implies that the results from a single method will definitely be included, possibly improved and considered as potential solutions.

The Multimethod approach was tested and applied on different domains, mainly medical (Lenic et al., 2003), where symbolic knowledge representation is required, which can give some explanation about decision making.

7 Experiments

In this section, we present the results of applying several data mining algorithms to carry out process path mining. We describe the data set that has been used, the algorithms applied to the data set, and finally we discuss the results obtained.

7.1 Data set description

The loan problem is the dataset that we used to illustrate how data mining methods can be applied to process path prediction. To generate our dataset, we started with the process presented in our introductory scenario (see Section 3), and using this as a process model graph, logged a set of 1000 process instance executions.

As explained in section three, our process log system has been extended to store information indicating the values of the input parameters passed to activities and the output parameters received from activities. The information also includes the path that has been followed during the execution of process instances. Each entry corresponds to an instance execution. The log system also stores a list of event records consisting of process names, instance identification, activity names, timestamps, variable names, etc (see Table 1 for an overview of the information stored in the process log).

From each process instance in the log system, a process profile is created. Each process profile provides the input for data mining algorithms. In our example, each profile is characterised by a set of six attributes: *income*, *loan_type*, *loan_amount*, *loan_years*, *name* and *SSN*. The profiles for the loan process contain two types of attributes: numeric and nominal. The attributes *income*, *loan_amount*, *loan_years* and *SSN* are numeric, whereas the attributes *loan_type* and *name* are nominal. As an example of a nominal attribute, *loan_type* can take the finite set of values *home-loan*, *education-loan*, and *car-loan*. Each profile is associated with a one of the six possible classes indicating the path followed during the execution of a process when the attributes of the profile have been assigned specific values (see Figure 1 to identify the six possible paths that can be followed during the execution of a loan process instance).

For the loan process application, the process instance profile and process path class schema is illustrated in Table 2. The data is formatted in a tabular format, with the process path as decision class, in a way that can processed by supervised learning algorithms.

7.2 Data mining algorithms

We have carried out process path mining to our data set using five distinct data mining algorithms: the Multimethod, and WEKA implementation of J48, Naïve Bayes, SMO, and MultiBoost. Since we have already explained the concepts behind the Multimethod approach in Section 4, we will only briefly describe the J48, Naïve Bayes, and SMO algorithms since they are fairly well-known to the data mining community.

J48. J48 algorithm is Weka's (2004) implementation of the C4.5 (Quinlan, 1993) decision three learner. The algorithm is commonly used to classify instances. Unfortunately the problem of finding an 'optimal' solution tree is a multi-objective problem and it is known to be NP complete. Therefore, C4.5 uses a heuristic approach to generate suboptimal decision trees. The hypothesis is represented in a symbolic form and can explain the reasons why some attributes and their values are involved in the decision making procedure.

Naïve Bayes. Naïve Bayes (NB) classifier technique is based on the so-called Bayesian theorem. It is a classification technique that is both predictive and descriptive. It analyses the relationship between each independent variable and the dependent variable to derive a conditional probability for each relationship. When a new case is analysed, a prediction is made by combining the effects of the independent variables on the dependent variables (i.e., the outcome that is predicted). Despite its simplicity, studies comparing classification algorithms have found the Naïve Bayesian classifier to be comparable in performance with classification trees and with neural network classifiers. The hypothesis is represented based on all the learning instances, but does not give a symbolic explanation of the decision making process.

SMO. SMO (Sequential Minimal Optimisation) (Platt, 1999) is a fast method to train SVM (Support Vector Machines) (Cortes and Vapnik, 1995). Training an SVM requires the solution of a very large quadratic programming (QP) optimisation problem. SMO breaks this large QP problem into a series of smaller possible QP problems.

Support Vector Machines (SVM) are learning systems that use a hypothesis space of linear functions in a high (possibly infinite) dimensional feature space, trained with a learning algorithm from optimisation theory that implement a learning bias derived from statistical learning theory. As SVM is supervised learning, a learning machine is required to scan through a given training set of examples with associated labels. The hypothesis is represented with a subset of important (support) learning instances with an importance weight.

MultiBoost. MultiBoost (MB) (Todorovski and Dzeroski, 2000) is an improved meta learning algorithm of AdaBoost (Freund and Schapire, 1999). It was constructed using the black box principle by observing the performance of previously constructed classifiers and adapting the weights of learning instances. It is widely used to improve the classification accuracy of weak learners.

J48 was selected as a good representative of the symbolic method, Naïve Bayes as a representative of a probabilistic method and the SMO algorithm as representative of a method that has been successfully applied in the domain of text-mining. Multiboost is expected to improve performance of single classifiers and can be used as direct comparison to Multimethod approach since it introduces meta-level classification.

7.3 Experiments

Each experiment has involved data from 1000 process instance executions. To effectively compare different learning methods that use heuristic and search space approaches, the dataset was separated in a learning (2/3 of the instances) and a testing (1/3 of the instances) set. For running instances not all the attributes are available once the instances are started. For that reason, we have conducted experiments with a variable number of attributes ranging from 0 attributes to 6 available attributes. We have conducted 64 experiments, i.e., all possible combination of attributes, analysing a total of 64,000 records containing data from process instance executions.

The first set of experiments was conducted using multiboost (MB) in conjunction with the selected classifiers. The Multimethod approach was not used in combination with multiboost since it is a meta-level approach and it is not integrated with WEKA. We should also mention that we have constrained the Multimethod approach not to use the Naïve Bayes and SMO methods; only symbolic knowledge representation was used.

The comparison of methods was implemented using the accuracy of the testing set. We obtained a very large number of results. For each one of the 64 experiments, we obtained data for seven different types of experiments. For reasons of simplicity and as a summary, we have compared each method to the best of all other methods and computed the relative error of accuracy on the testing set by dividing the accuracy produced by the target classifier with the accuracy of the best of all other classifiers. The comparisons of the most promising methods are presented in Figure 9. The points above the line represent the experiments for which the method under study was better than the best of all other methods. To summarise all result we have grouped (summed) differences based on the number of attributes used in the experiments. An overview of our results is given in Table 3.

Table 3 Comparison of approaches using multiboost algorithm for single methods

No. of attributes	*No. of exp.*	*Multi-method*	*MB J48*	*MB NB*	*MB SMO*
1	6	16.12%	–21.67	–23.00%	–47.49%
2	15	35.21%	–119.88	–49.27%	–106.88%
3	20	69.46%	–168.04	–90.57%	–116.08%
4	15	61.34%	–157.51	–75.21%	–74.59%
5	6	19.78%	–67.73	–28.67%	–19.66%
6	1	3.65%	–12.49	–3.52%	–4.81%
Average		*205.57%*	*–547.32*	*–270.24%*	*–369.50%*
Best or equal		*50*	*2*	*13*	*5*
Best		*45*	*0*	*11*	*2*

We can observe that the Multimethod approach performed on average much better than multiboosted WEKA classifiers. It produces a model with the best or equal accuracy in 50 out of 64 experiments and outperformed others in 45 experiments. The worst results were produced by the multiboosted J48 algorithm which uses symbolic decision tree knowledge representation. An interesting observation is that the Multimethod approach uses decision trees as knowledge representation since this representation is preferred by the approach because it has a symbolic nature.

It is safe to assume that the multiboosted approach overfitted and was not able to find a generalised concept. That is probably because of the nature of the dataset that contains attributes that introduce noise.

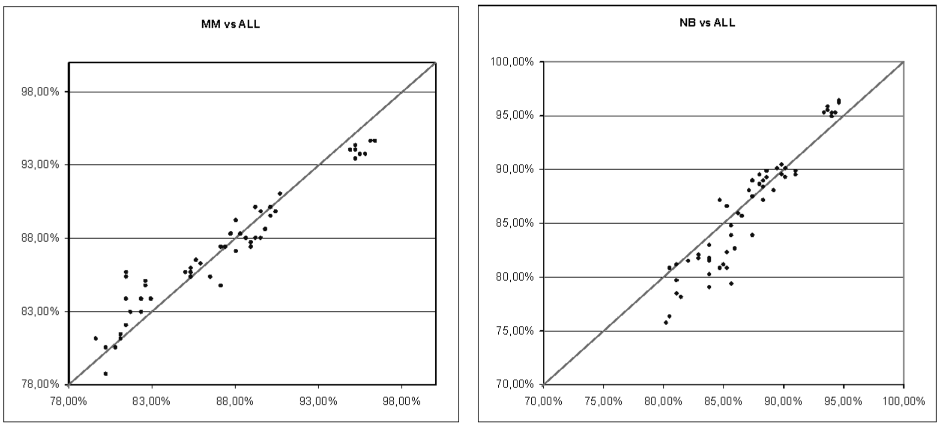
Next we removed the meta-level of the multiboost algorithm and repeated the experiments only with single methods. As expected, a single model made more accurate prognoses than ensembles of methods. The results are summarised in Table 4.

Table 4 Comparison of approaches without using the multiboost algorithm

No. of attributes	*No. of exp.*	*Multimethod*	*J48*	*NB*	*SMO*
1	6	15.38%	–19.88%	–23.00%	–47.49%
2	15	15.38%	–52.05%	–35.85%	–104.62%
3	20	6,07%	–55.58%	–38.93%	–108.69%
4	15	1.10%	–62.97%	–10.93%	–60.32%
5	6	–6.50%	–32.12%	6.65%	–20.10%
6	1	–1.89%	–5.66%	1.93%	–4.40%
Average		*29.52%*	*–228.26%*	*–100.13%*	*–345.63%*
Best or equal		*41*	*7*	*26*	*3*
Best		*31*	*0*	*21*	*0*

On average the Multimethod approach performs better then all other single methods when compared to each other. It can be observed that the Multimethod approach produces better results with smaller numbers of attributes. When the number of attributes is increased the accuracy of Naïve Bayes improves. Accuracy for concrete experiments is shown in Figure 8. Experiments that produced an accuracy less than 70% are not presented.

Figure 9 Graphical comparison of the Multimethod and Naïve Bayes. The dots above the line mean that the classifier performed better than all other classifiers



We can observe that Naïve Bayes produced more accurate results when a more appropriate set of attributes was proposed. The Multimethod approach did not include the Naïve Bayes and SMO classifiers in those experiments. This situation was changed when we have the SMO and the NB to the Multimethod framework. A proposed solution given by the Multimethod approach was composed of a decision tree, Naïve Bayes, and a SMO classifier. All the classifiers produced the highest accuracy in experiment n°15, since this experiment includes the 4 most informative attributes. The results are presented in Table 5.

Table 5 Compression of best results

Method	*J48*	*NB*	*SMO*	*MM*	*MM with NB and SMO*
Max acc. (%)	93,41	96,41	93,11	94,61	96,71
Experiment	15	15	15	15	15
No. of attributes	4	4	4	4	4

Why did we not include the Naïve Bayes and SMO methods in the Multimethod from the beginning? Following the idea of the Multimethod, we have only used symbolic approaches, so the final result had a symbolic knowledge representation. If we had included the Naïve Bayes and the SMO methods, the resulting solution (if there would be no better one) would be either for Naïve Bayes or SMO, which also implies that comparison with the method would not have made any sense. We were able to show that we can get very good results by using other (symbolic) knowledge representations using different methods. Of course, when we include an additional method search space, the expressive power of the Multimethod approach is increased.

We should also stress that the Multimethod approach combines different search space approaches, using quite a large amount of processing power, and it is not as fast in producing hypotheses as the other methods.

8 Related work

Since our work combines two important areas, path mining analysis and efficient data mining algorithms, we divide this section into three subsections to discuss the related work in each area and in the area that interrelates process mining, data mining, and path mining.

8.1 Process mining and process path prediction

Most of the research in this area has targeted process mining. The techniques and algorithms developed use information stored in process logs to discover process models. Process mining is addressed in several papers and a detailed survey of this research area is provided in Aalst et al. (2003).

In Herbst and Karagiannis (1998) a machine learning component able to acquire and adapt a workflow model from observations of manually enacted workflow instances is described. The component is able to generalise a number of execution traces from different workflow instances to a single more general workflow model covering all traces.

In Weijters and van der Aalst (2001) a process mining technique is presented. The technique allows the discovery of workflow models from a workflow log containing information about the workflow process as it is actually being executed. The procedure includes three steps, the construction of the dependency/frequency table, induction of dependency/frequency graphs (D/F-graphs), and the generation of WF-nets from D/F-graphs. Their experiments on six noise free workflow logs showed that their method created six perfect D/F-graphs.

In Agrawal et al. (1998) an algorithm that allows the user to use existing workflow execution logs to automatically model a given business process as a graph is presented. The algorithm has been applied to synthetic datasets as well as logs obtained from a workflow management system (Flowmark) installation. The results showed that the graphs the algorithm derived in experiments were very good approximations of the original graph.

In Grigoria et al. (2004), a set of integrated tools that use data mining algorithms to support process execution by providing several features, such as analysis, prediction, monitoring, control, and optimisation is presented. The set of tools is referred to as the Business Process Intelligence (BPI) tool suite.

A few contributions exist in the field of path prediction (Cardoso, 2002; Cardoso et al., 2004), however, they are limited to estimating of process' QoS based on simple statistical techniques. We will show that advanced data mining algorithm are more accurate in predicting execution paths.

8.2 Data mining and the Multimethod approach

The idea of combining data mining techniques with a meta-level approach has been successfully applied on different domains, but mostly using a black box approach. Examples of approaches include AdaBoost (Freund and Schapire, 1996), Bagging and MultiBoost (Webb, 2000). This last method combines boosting and wagging.

Most of the research has been concentrated on improving and developing new methods with single or specific combinations of knowledge representations. For example, the hybrid system KBANN (Towell and Shavlik, 1995) maps symbolic rules into neural networks. The rules are used to determine the topology of the initial network, which serves as a good starting point for training.

INLEN (Michalski, 1997) applies a multistrategy approach based on Inferential Theory of Learning (Michalski, 1994) by simulating human learning and applying similar transformations to the knowledge base. Users must specify an appropriate transformation, using a knowledge generation language, to transform knowledge and produce different ontologies. The Multimethod approach follows a similar idea, but it is applied to the field of automatic knowledge extraction from learning instances.

HHL (Lee and Shin, 1999) applies the inductive logic programming algorithm by sequentially using multiple strategies for learning rules in a restricted first-order logic from very large databases. The process is executed with user interaction that selects appropriate hierarchies to direct the learning process.

8.3 Process mining, data mining, and path mining

Compared to the approaches discussed in the previous sections, our work focuses on path mining, and not process mining. In path mining, the process model is known. On the other hand, process mining has the goal of discovering process models from BPMS logs, addressing the design phase of processes. We aim at the monitoring phase, where we predict the QoS of process instances.

The work which is most comparable to ours is described in Grigoria et al. (2004). Their work targets process analysis, prediction, monitoring, control, and optimisation. Nevertheless, they have not contemplated path mining analysis. Moreover, they carry out process analysis experiments with traditional data mining techniques. In our research, we realise experiments with new and improved data mining algorithms, such as the Multimethod which dynamically combines different methods into hybrids using tight coupling between the methods by sharing knowledge.

9 Conclusions

Business Process Management Systems, workflow systems, workflows, web processes, and web services represent fundamental technological infrastructures that efficiently define, manage, and support business processes. The data generated from the execution and management of processes, which is usually stored in a log system, is valuable and can be used to discover and extract knowledge about processes' executions, behaviour, and structure.

In this paper we have shown that one important area of research related to processes is the prediction of the paths that will be followed during the execution of a process instance. We call this type of prediction 'path mining'.

We have shown and illustrated how path mining can be achieved using data mining techniques to automatically extract path knowledge from process logs. Path mining knowledge is indispensable for Quality of Service (QoS) analysis and prediction.

Organisations operating in modern markets, such as e-commerce activities and distributed web services interactions require QoS management. Appropriate control of quality leads to the creation of quality products and services; these, in turn, fulfill customer expectations and achieve customer satisfaction.

Due to the increased importance of QoS management for organisations, we have compared several data mining algorithms to carry out path mining. J48 was selected as a good representative of a symbolic method, Naïve Bayes as a representative of a probabilistic method, SMO a as representative of a method that has been successfully applied in the domain of text-mining, and the Multimethod as a representative of a new method that dynamically combines and applies different methods (heuristic and search space learning methods) in no predefined order to the path mining problem.

The experimental results were encouraging. The results presented in this paper show that the Multimethod outperforms in practice the J48, Naïve Bayes, and SMO when applied to the path mining problem. The Multimethod approach has been previously tested and applied successfully on medical domains. From our experiments, we can conclude that the Multimethod approach is also a good solution to perform path mining on administrative and production processes.

References

Aalst, W.M.P.v.d., Dongen, B.F.v. *et al*. (2003) 'Workflow mining: a survey of issues and approaches', *Data and Knowledge Engineering (Elsevier)*, Vol. 47, No. 2, pp.237–267.

Agrawal, R., Gunopulos, D. *et al*. (1998) *Mining Process Models from Workflow Logs*, Sixth International Conference on Extending Database Technology, Valencia, Spain, Springer.

Anyanwu, K., Sheth, A. *et al*. (2003) 'Healthcare enterprise process development and integration', *Journal of Research and Practice in Information Technology, Special Issue in Health Knowledge Management*, Vol. 35, No. 2, pp.83–98.

CAPA (1997) *Course Approval Process Automation (CAPA)*, LSDIS Lab, Department of Computer Science, University of Georgia, Athens, GA.

Cardoso, J. (2002) *Stochastic Workflow Reduction Algorithm*, LSDIS Lab, Department of Computer Science, University of Georgia, Georgia, USA.

Cardoso, J. (2005) 'Path mining in web processes using profiles', *Encyclopedia of Data Warehousing and Mining*, Wang, J., Idea Group Inc., Pennsylvania, USA, pp.896–901.

Cardoso, J. and Sheth, A. (2003) 'Semantic e-workflow composition', *Journal of Intelligent Information Systems (JIIS)*, Vol. 21, No. 3, pp.191–225.

Cardoso, J., Bussler, C. *et al*. (2002) *Tutorial: Semantic Web Services and Processes: Semantic Composition and Quality of Service*, International Federated Conferences: DOA/ODBASE/CooPIS 2002, Irvine, CA.

Cardoso, J., Miller, J. *et al*. (2004) 'Modeling quality of service for workflows and web service processes', *Web Semantics: Science, Services and Agents on the World Wide Web Journal*, Vol. 1, No. 3, pp.281–308.

Chandrasekaran, S., Silver, G. *et al*. (2002) 'Service technologies and their synergy with simulation', *Proceedings of the 2002 Winter Simulation Conference (WSC'02)*, San Diego, California, pp.606–615.

Cortes, C. and Vapnik, V. (1995) 'Support vector networks', *Machine Learning*, Vol. 20, pp.273–297.

Cowley, S. (2002) *Study: BPM Market Primed for Growth*, September 23, http://www.infoworld.com/article/02/09/23/020923hnbpm_1.html.

Dietterich, T.G. (2000) 'Ensemble methods in machine learning', *First International Workshop on Multiple Classifier Systems*, Cagliari, Italy, Springer Verlag, New York.

Freund, Y. and Schapire, R. (1999) 'A short introduction to boosting', *Journal Japan Soc. for Artifitial Intelligence*, Vol. 14, No. 5, pp.771–780.

Freund, Y. and Schapire, R.E. (1996) 'Experiments with a new boosting algorithm', *13th International Conference on Machine Learning*, Morgan Kaufmann, 3–6 July, Bari, Italy, ISBN 1-55860-419-7.

Gillmann, M., Weikum, G. *et al*. (2002) 'Workflow management with service quality guarantees', *ACM SIGMOD'2002 International Conference on Management of Data*, Madison, Wisconsin.

Goldberg, D.E. (1989) *Genetic Algorithms in Search*, *Optimisation, and Machine Learning*, Addison Wesley, Reading, MA, USA.

Grigoria, D., Casatib, F. *et al*. (2004) 'Business process intelligence', *Computers in Industry*, Vol. 53, pp.321–343.

Hall, R.D., Miller, J.A. *et al*. (2003) 'Using workflow to build an information management system for a geographically distributed genome sequence initiative', in Prade, R.A. and Bohnert, H.J. (Eds.): *Genomics of Plants and Fungi*, Marcel Dekker Inc., New York, pp.359–371.

Herbst, J. and Karagiannis, D. (1998) 'Integrating machine learning and workflow management to support acquisition and adaption of workflow models', *Ninth International Workshop on Database and Expert Systems Applications*,Vienna, Austria, pp.745–752.

Kang, M.H., Froscher, J.N. *et al*. (1999) 'A multilevel secure workflow management system', *Proceedings of the 11th Conference on Advanced Information Systems Engineering*, Springer-Verlag, Heidelberg, Germany, pp.271–285.

Lee, C.H. and Shin, D.G. (1999) 'A multistrategy approach to classification learning in databases', *Data and Knowledge Engineering*, Vol. 31, No. 1, pp.67–93.

Lenic, M. and Kokol, P. (2002) 'Combining classifiers with Multimethod approach', *Second International Conference on Hybrid Intelligent Systems, Soft Computing Systems: Design, Management and Applications, Frontiers in Artificial Intelligence and Applications*, IOS Press, Amsterdam.

Lenic, M., Povalej, P. *et al*. (2003) 'Improved medical decision support with Multimethod approach', *IEEE Symposium on Computer-Based Medical Systems (CBMS 2003)*, IEEE Computer Society, New York, USA, pp.242–247.

Luo, Z., Sheth, A. *et al*. (2003) 'Exception handling for conflict resolution in cross-organisational workflows', *Distributed and Parallel Databases*, Vol. 12, No. 3, pp.271–306.

McCready, S. (1992) 'There is more than one kind of workflow software', *Computerworld*, Vol. 2, November, pp.86–90.

McGarry, K., Wermter, S. *et al*. (2001) 'The extraction and comparison of knowledge from local function networks', *International Journal of Computational Intelligence and Applications*, Vol. 1, No. 4, pp.369–382.

Michalski, R.S. (1994) 'Inferential theory of learning: developing foundations for multistrategy learning', *Machine Learning*, Vol. 4, pp.3–61.

Michalski, R.S. (1997) *Data Mining and Knowledge Discovery: A Review of Issues and A Multistrategy Approach*, George Mason University, Fairfax, VA.

Miller, J.A., Cardoso, J.S. *et al*. (2002) 'Using simulation to facilitate effective workflow adaptation', *Proceedings of the 35th Annual Simulation Symposium (ANSS'02)*, San Diego, California.

Musa, J.D. (1993) 'Operational profiles in software-reliability engineering', *IEEE Software*, Vol. 10, No. 2, pp.14–32.

Musa, J.D. (1999) *Software Reliability Engineering: More Reliable Software, Faster Development and Testing*, McGraw-Hill, New York.

Platt, J. (1999) 'Fast training of support vector machines using sequential minimal optimisation', in Scholkopf, B., Burges, C.J.C. and Smola, A.J. (Eds.): *Advances in Kernel Methods – Support Vector Learning*, MIT Press, Cambridge, MA, pp.185–208.

Q-Link (2002) *BPM2002: Market Milestone Report*, Q-Link Technologies, http://www.qlinktech.com.

Quinlan, J.R. (1993) *C4.5: Programs for Machine Learning*, Morgan Kaufmann, Morgan Kaufmann, San Francisco, ISBN: 1-55860-238-0.

Sheth, A.P., Aalst, W.v.d. *et al*. (1999) 'Processes driving the networked economy', *IEEE Concurrency*, Vol. 7, No. 3, pp.18–31.

Smith, H. and Fingar, P. (2003) *Business Process Management (BPM): The Third Wave*, Meghan-Kiffer Press, Florida, USA.

Todorovski, L. and Dzeroski, S. (2000) 'Combining multiple models with meta decision trees', *Proceedings of the Fourth European Conference on Principles of Data Mining and Knowledge Discovery*, Springer, Berlin, Germany, pp.54–64.

Towell, G.G. and Shavlik, J.W. (1995) 'Knowledge-based artificial neural networks', *Artificial Intelligence*, Vol. 70, pp.119–165.

Webb, I.G. (2000) 'MultiBoosting: a technique for combining boosting and wagging', *Machine Learning*, Vol. 40, No. 2, pp.159–196.

Weijters, T. and van der Aalst, W.M.P. (2001) 'Process mining: discovering workflow models from event-based data', *13th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC 2001)*, Amsterdam, The Netherlands.

Weka (2004) Weka, http://www.cs.waikato.ac.nz/ml/weka/.

WfMC (2002) *Workflow Management Coalition*, http://www.wfmc.org/.

Wolpert, D. and Macready, W. (1997) 'No free lunch theorems for optimisation', *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, pp.67–82.