Path Mining in Web Processes Using Profiles

Jorge Cardoso
University of Maderia, Portugal

INTRODUCTION

Business process management systems (BPMSs) (Smith & Fingar, 2003) provide a fundamental infrastructure to define and manage business processes, Web processes, and workflows. When Web processes and workflows are installed and executed, the management system generates data describing the activities being carried out and is stored in a log. This log of data can be used to discover and extract knowledge about the execution of 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 process. I call this type of discovery path mining. Path mining is vital to algorithms that estimate the quality of service of a process, because they require the prediction of paths. In this work, I present and describe how process path mining can be achieved by using data-mining techniques.

BACKGROUND

BPMSs, such as workflow management systems (WfMS) (Cardoso, Bostrom, & Sheth, 2004) are systems capable of both generating and collecting considerable amounts of data describing the execution of business processes, such as Web processes. This data is stored in a process log systems, which are vast data archives that are seldom visited. Yet, the data generated from the execution of processes are rich with concealed information that can be used for making intelligent business decisions.

One important and useful piece of knowledge to discover and extract from process logs is the implicit rules that govern path mining.

In Web processes for e-commerce, suppliers and customers define a contract between the two parties, specifying quality of service (QoS) items, such as products or services to be delivered, deadlines, quality of products, and cost of services. The management of QoS metrics directly impacts the success of organizations participating in e-commerce. A Web process, which typically can have a graphlike representation, includes a number of linearly independent control paths. Depending on the path followed during the execution of a Web process, the QoS may be substantially different. If you can predict with a certain degree of confidence the path that will be followed at run time, you can significantly increase the precision of QoS estimation algorithms for Web processes.

Because the large amounts of data stored in process logs exceeds understanding, I describe the use of data-mining techniques to carry out path mining from the data stored in log systems. This approach uses classification algorithms to conveniently extract patterns representing knowledge related to paths. My work is novel because no previous work has targeted the path mining of Web processes and workflow. The literature includes only work on process and workflow mining (Agrawal, Gunopulos, & Leymann, 1998; Herbst & Karagiannis, 1998; Weijters & van der Aalst, 2001).

Process mining allows the discovery of workflow models from a workflow log containing information about workflow processes executed. Luo, Sheth, Kochut, and Arpinar (2003) present an architecture and the implementation of a sophisticated exception-handling mechanism supported by a case-based reasoning (CBR) engine.

MAIN THRUST

The material presented in this section emphasizes the use of data-mining techniques for uncovering interesting process patterns hidden in large process logs. The method contained in the next section is more suitable for administrative and production processes compared to Ad-hoc and collaborative processes, because they are more repetitive and predictable.

Web Process Scenario

A major bank has realized 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. 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.

A Web process is composed of Web services and transitions. Web services are represented by circles, and transitions are represented by arrows. Transitions express dependencies between Web services. A Web service 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* Web services enable only one outgoing transition after completing their execution. And-split Web services are represented with '•', and xor-split Web services are represented with '⊕'. A Web service with more than one incoming transition can be classified as an and-join or xor-join. *And-join* Web services start their execution when all their incoming transitions are enabled. *Xor-join* Web services are executed as soon as one of the incoming transitions is enabled. As with and-split and xor-split Web services, and-join and xor-join Web services are represented with the symbols '•' and '⊕', respectively.

The Web process of this scenario is composed of 14 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 out 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 Loan, or Check Car Loan.

Educational loans are not handled and managed automatically. After an educational loan application is submitted and checked, a notification is immediately sent informing the client that he or she has to contact the bank personally.

A loan request can be either accepted (Approve Home Loan and Approve Car Loan) or rejected (Reject Home Loan and Reject Car Loan). In the case of a home loan, however, the loan can also be approved conditionally. The Web service Approve Home Loan Conditionally, as the name suggests, approves a home loan under a set of conditions.

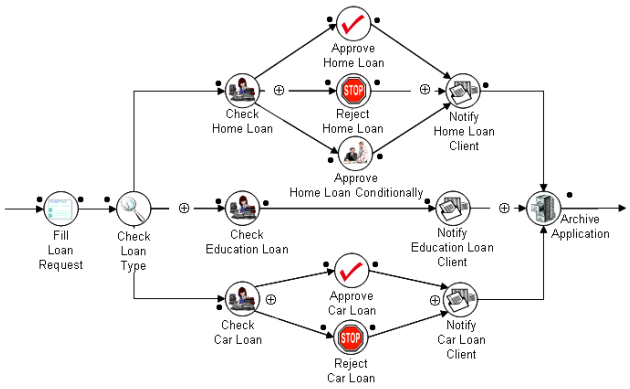
The following formula is used to determine if a loan is approved or rejected.

$$MP = (L*R*(1+R/12)12*NY)/(-12+12*(1+R\ 12)12*NY) \quad (1)$$

MP=Monthly payment,L=Loan amount,R=Interest rate,NY=Number of years

When the result of a loan application is known, it is e-mailed to the client. Three Web services are responsible for notifying the client: Notify Home Loan Client, Notify Education Loan Client, and Notify Car Loan Client. Finally, the Archive Application Web service creates a report and stores the loan application data in a database record.

Figure 1. The loan process



Web Process Log

During the execution of Web processes (such as the one presented in Figure 1), events and messages generated by the enactment system are stored in a Web process log. These data stores provide an adequate format on which path mining can be performed. The data includes real-time information describing the execution and behavior of Web processes, Web services, instances, transitions, and other elements such as runtime QoS metrics. Table 1 illustrates an example of a modern Web process log.

To perform path mining, current Web process logs need to be extended to store information indicating the values and the type of the input parameters passed to Web services and the output parameters received from Web services. Table 2 shows an extended Web process log that accommodates input/output values of Web services parameters generated at run time. Each Parameter/value entry has a type, parameter name, and value (e.g., string loan-type="car-loan").

Additionally, the Web process log needs to include path information describing the Web services that have been executed during the enactment of a Web process. This information can be easily 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 to the entry corresponding to the last service of a process to be executed. For example, in the Web process log illustrated in Table 2, the service NotifyUser is the last service of a Web process. The log has been extended in such a way that the NotifyUser record contains information about the path that was followed during the Web process execution.

Table 1. Web process log

Date	Web process	Process instance	Web service	Service instance	Cost	Duration	…
6:45 03-03-04	LoanApplication	LA04	RejectCarLoan	RCL03	$1.2	13 min	…
6:51 03-03-04	TravelRequest	TR08	FillRequestTravel	FRT03	$1.1	14 min	…
6:59 03-03-04	TravelRequest	TR09	NotifyUser	NU07	$1.4	24 hrs	…
7:01 03-03-04	InsuranceClaim	IC02	SubmitClaim	SC06	$1.2	05 min	…
…	…	…	…	…	…	…	…

Table 2. Extended Web process log

…	Process instance	Web service	Service instance	Parameter/value	Path	…
…	LA04	RejectCarLoan	RCL03	int LoanNum=14357; string loan-type="car-loan"	…	…
…	LA04	NotifyCLoanClient	NLC07	string e-mail="jf@uma.pt"	…	…
…	LA05	CheckLoanRequest	CLR05	double income=12000; string Name="Eibe Frank";	…	…
…	TR09	NotifyUser	NU07	String e-mail=jf@uma.pt; String tel="35129170023"	FillForm->CheckForm-> Approve->Sign->Report	…
…	…	…	…		…	…

Web Process Profile

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

A profile contains two types of attributes, numeric and nominal. Numeric attributes measure numbers, either real or integer-valued. For example, Web services inputs or outputs parameters that are of type byte, decimal, int, short, or double will be placed in the profile and classified as numeric. In Table 2, the parameters LoanNum, income, BudgetCode, income, and tel will be classified as numeric in the profile.

Nominal attributes take on values within a finite set of possibilities. Nominal quantities have values that are distinct symbols. For example, the parameter loan-type from the loan application and present in Table 2 is nominal because it can take the finite set of values: home-loan, education-loan, and car-loan. In my approach, string and Boolean data type manipulated by Web services are considered to be nominal attributes.

Profile Classification

The attributes present in a profile trigger the execution of a specific set of Web services. Therefore, for each profile previously constructed, I associate an additional attribute, the path attribute, indicating the path followed when the attributes of the profile have been assigned to specific values. The path attribute is a target class. Classification algorithms classify samples or instances into target classes.

After the profiles and a path attribute value for each profile have been determined, I can use data-mining methods to establish a relationship between the profiles and the paths followed at run time. One method appropriate to deal with my problem is the use of classification.

In classification, a learning schema takes a set of classified profiles, from which it is expected to learn a way of classifying unseen profiles. Because the path of each training profile is provided, my methodology uses supervised learning.

EXPERIMENTS

In this section, I present the results of applying my algorithm to a synthetic loan dataset. To generate a synthetic dataset, I start with the process presented in the introductory scenario and, using this as a process model graph, log a set of process instance executions.

The data are lists of event records stored in a Web process log consisting of process names, instance identification, Web services names, variable names, and so forth. Table 3 shows the additional data that have been stored in the Web process log. The information includes the Web service variable values that are logged by the system and the path that has been followed during the execution of instances. Each entry corresponds to an instance execution.

Web process profiles provide the input to machine learning and are characterized 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

Table 3. Additional data stored in the Web process log

Income	Loan_Type	Loan_amount	Loan_years	Name	SSN	Path
1361.0	Home-Loan	129982.0	33	Bernard-Boar	10015415	FR>CLT>CHL>AHL>NHC>CA
Unknown	Education-Loan	Unknown	Unknown	John-Miller	15572979	FR>CLT>CEL>CA
1475.0	Car-Loan	15002.0	9	Eibe-Frank	10169316	FR>CLT>CCL>ACL>NCC>CA
...	...	...	...	...	...	...

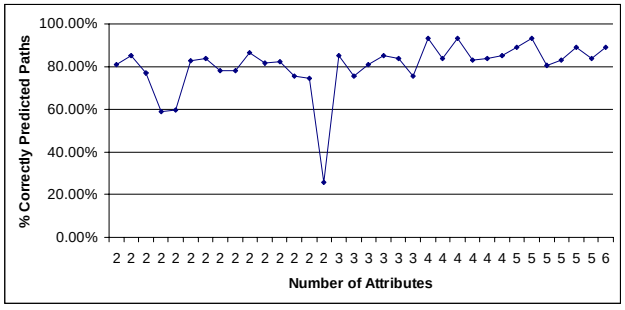
of attributes: numeric and nominal. The attributes income, loan_amount, loan_years, and SSN are numeric, whereas the attributes loan_type and name are nominal. As an example of a nominal attribute, loan_type can take the finite set of values home-loan, education-loan, and car-loan. These attributes correspond to the Web service input/output parameters that have been stored previously in the Web process log presented in Table 3.

Each profile is associated with a class indicating the path that has been followed during the execution of a process when the attributes of the profile have been assigned specific values. The last column of Table 3 shows the class named path. The profiles and path attributes will be used to establish a relationship between the profiles and the paths followed at runtime. The profiles and the class path have been extracted from the Web process log.

After profiles are constructed and associated with paths, these data are combined and formatted to be analyzed using Weka (2004), a set of software for machine learning and data mining. The data is automatically formatted using the ARFF format. I have used the J.48 algorithm, which is Weka's implementation of the C4.5 (Hand, Mannila, & Smyth, 2001) decision tree learner to classify profiles. C4.5 decision tree learner is one of the most well-known decision tree algorithms in the data-mining community. Weka system and its data format (ARFF) is also one of the most well-known data-mining systems in academia.

Each experiment has involved data from 1,000 Web process executions and a variable number of attributes (ranging from two to six). I have conducted 34 experiments, analyzing a total of 34,000 records containing data from Web process instance executions. Figure 2 shows the results that I have obtained.

Figure 2. Experimental results



The path-mining technique developed has achieved encouraging results. When three or more attributes are involved in the prediction, the system is able to predict correctly the path followed for more than 75% of the process instances. This accuracy improves when four attributes are involved in the prediction; in this case, more than 82% of the paths are correctly predicted. When five attributes are involved, I obtain a level of prediction that reaches a high of 93.4%. Involving all six attributes in the prediction gives excellent results: 88.9% of the paths are correctly predicted. When a small number of attributes are involved in the prediction, the results are not as good. For example, when only two attributes are selected, I obtain predictions that range from 25.9% to 86.7%.

FUTURE TRENDS

Currently, organizations use BPMSs, such as WfMS, to define, enact, and manage a wide range of distinct applications (Q-Link Technologies, 2002), such as insurance claims, bank loans, bioinformatic experiments (Hall, Miller, Arnold, Kochut, Sheth, & Weise, 2003), health-care procedures (Anyanwu, Sheth, Cardoso, Miller, & Kochut, 2003), and telecommunication services (Luo, Sheth, Kochut, & Arpinar, 2003).

In the future, I expect to see a wider spectrum of applications managing processes in organizations. 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).

The concept of path mining can be used effectively in many business applications — for example, to estimate the QoS of Web processes and workflows (Cardoso, Miller, Sheth, Arnold, & Kochut, 2004) — because the estimation requires the prediction of paths. Organizations operating in modern markets, such as e-commerce activities and distributed Web services interactions, require QoS management. Appropriate quality control leads to the creation of quality products and services; these, in turn, fulfill customer expectations and achieve customer satisfaction (Cardoso, Sheth, & Miller, 2002).

CONCLUSION

BPMSs, Web processes, workflows, and workflow systems represent fundamental technological infrastructures that efficiently define, manage, and support business processes. The data generated from the execution and management of Web processes can be used to discover and extract knowledge about the process executions and structure.

I have shown that one important area of Web processes to analyze is path mining. I have demonstrated how path mining can be achieved by using data-mining techniques, namely classification, to extract path knowledge from Web process logs. From my experiments, I can conclude that classification methods are a good solution to perform path mining on administrative and production Web processes.

REFERENCES

Agrawal, R., Gunopulos, D., & Leymann, F. (1998). Mining process models from workflow logs. *Proceedings of the Sixth International Conference on Extending Database Technology, Spain.*

Anyanwu, K., Sheth, A., Cardoso, J., Miller, J. A., & Kochut, K. J. (2003). Healthcare enterprise process development and integration. *Journal of Research and Practice in Information Technology, 35*(2), 83–98.

Cardoso, J., Bostrom, R. P., & Sheth, A. (2004). Workflow management systems and ERP systems: Differences, commonalities, and applications. *Information Technology and Management Journal, 5*(3–4), 319–338.

Cardoso, J., Miller, J., Sheth, A., Arnold, J., & Kochut, K. (2004). Quality of service for workflows and Web service processes. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 1*(3), 281–308.

Cardoso, J., Sheth, A., & Miller, J. (2002). Workflow quality of service. *Proceedings of the International Conference on Enterprise Integration and Modeling Technology and International Enterprise Modeling Conference, Spain.*

Cowley, S. (2002, September 23). *Study: BPM market primed for growth.* Available from the InfoWorld Web site, *http://www.infoworld.com*

Hall, R. D., Miller, J. A., Arnold, J., Kochut, K. J., Sheth, A. P., & Weise, M. J. (2003). Using workflow to build an information management system for a geographically distributed genome sequence initiative. In R. A. Prade & H. J. Bohnert (Eds.), *Genomics of plants and fungi* (pp. 359–371). New York: Marcel Dekker.

Hand, D. J., Mannila, H., & Smyth, P. (2001). *Principles of data mining.* Bradford Book.

Herbst, J., & Karagiannis, D. (1998). Integrating machine learning and workflow management to support acquisition and adaption of workflow models. *Proceedings of the Ninth International Workshop on Database and Expert Systems Applications*.

Luo, Z., Sheth, A., Kochut, K., & Arpinar, B. (2003). Exception handling for conflict resolution in cross-organizational workflows. *Distributed and Parallel Databases, 12*(3), 271–306.

Q-Link Technologies. (2002). *BPM2002: Market milestone report*. Retrieved from *http://www.qlinktech.com*.

Smith, H., & Fingar, P. (2003). *Business process management (BPM): The third wave*. Meghan-Kiffer Press.

Weijters, T., & van der Aalst, W. M. P. (2001). Process mining: Discovering workflow models from event-based data. *Proceedings of the 13th Belgium-Netherlands Conference on Artificial Intelligence*.

(2004). Weka [Computer software.] Retrieved from *http://www.cs.waikato.ac.nz/ml/weka/*

KEY TERMS

Business Process: A set of one or more linked activities that collectively realize a business objective or goal, normally within the context of an organizational structure.

Business Process Management System (BPMS): Provides an organization with the ability to collectively define and model its business processes, deploy these processes as applications that are integrated with its existing software systems, and then provide managers with the visibility to monitor, analyze, control, and improve the execution of those processes.

Process Definition: The representation of a business process in a form that supports automated manipulation or enactment by a workflow management system.

Web Process: A set of Web services that carry out a specific goal.

Web Process Data Log: Records and stores events and messages generated by the enactment system during the execution of Web processes.

Web Service: Describes a standardized way of integrating Web-based applications by using open standards over an Internet protocol.

Workflow: The automation of a business process, in whole or part, during which documents, information, or tasks are passed from one participant to another for action, according to a set of procedural rules.

Workflow Management System: A system that defines, creates, and manages the execution of workflows through the use of software, which is able to interpret the process definition, interact with participants, and, where required, invoke the use of tools and applications.