

Path Mining in Web Processes Using Profiles

Jorge Cardoso

University of Madeira, 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 workflows. 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 condition-

ally. 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.

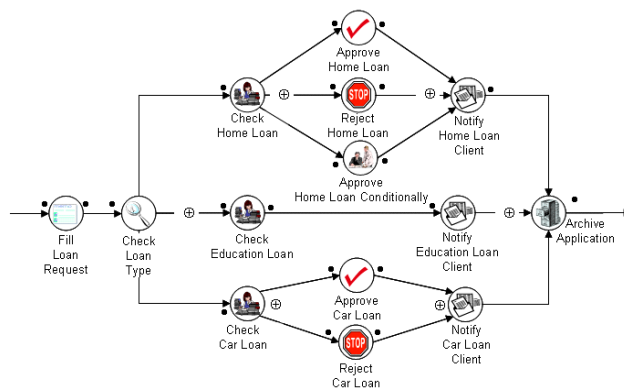
Web Process Log

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.

Figure 1. The loan process



4 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/path-mining-web-processes-using/10723

Related Content

Multimodal Analysis in Multimedia Using Symbolic Kernels

Hrishikesh B. Aradhye and Chitra Dorai (2005). *Encyclopedia of Data Warehousing and Mining* (pp. 842-847). www.irma-international.org/chapter/multimodal-analysis-multimedia-using-symbolic/10714

A Look back at the PAKDD Data Mining Competition 2006

Nathaniel B. Noriel and Chew Lim Tan (2008). *Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications* (pp. 84-92). www.irma-international.org/chapter/look-back-pakdd-data-mining/7634

Discovering Surprising Instances of Simpson's Paradox in Hierarchical Multidimensional Data

Carem C. Fabris and Alex A. Freitas (2008). *Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications* (pp. 3235-3251). www.irma-international.org/chapter/discovering-surprising-instances-simpson-paradox/7831

Feature Selection for the Promoter Recognition and Prediction Problem

George Potamias and Alexandros Kanterakis (2008). *Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications* (pp. 2248-2262). www.irma-international.org/chapter/feature-selection-promoter-recognition-prediction/7758

Materialized Hypertext View Maintenance

Giuseppe Sindoni (2005). *Encyclopedia of Data Warehousing and Mining* (pp. 710-713). www.irma-international.org/chapter/materialized-hypertext-view-maintenance/10689