FlowSpy: exploring Activity-Execution Patterns from Business Processes

Cristian Tristão[1], Duncan D. Ruiz[1], Karin Becker[2]

[1]Faculdade de Informática - Pontifícia Universidade Católica do Rio Grande do Sul
Av. Ipiranga 6681, 90619-900 – Porto Alegre – RS – Brazil

[3]Quality Knowledge – Porto Alegre – RS – Brazil

{ctristaobr@gmail.com, duncan@pucrs.br, kbeckerbr@gmail.com}

Abstract. The paper describes FlowSpy, an environment that employs a sequence mining technique to discover and analyze actual process execution paths from business processes, for both process comparison and process discovery. FlowSpy focuses on exploratory analysis of the different execution flows, enabling a detailed analysis of business behavior, quantification of different execution flows, and abstraction mechanisms (log pre-processing and visualization abstraction) that deal with process complexity and different process views. Log pre-processing aims at improving the data mining phase, with a more restricted aggregate tree. Visualization abstraction facilitates pattern interpretation by producing trees that represent the obtained patterns.

1 Introduction

Business processes are increasingly automated and controlled by information systems that generate and store large datasets. Business Process (BP) management and automation have become a reality in organizations. There exists a broad range of products that control BP, such as Workflow Management Systems (WfMS), ERP and CRM. The systematic use of these products creates and stores huge amounts of data, which reflect the actual state and behavior of BPs. Lately, organizations have shifted the focus from process management and control, to the measurement, analysis and monitoring of BPs (Casati, 2005; Golfarelli, Rizzi and Cella, 2004). Analyses based on (1) the summarization of BP historical data (Castellanos et al., 2005; Golfarelli, Rizzi and Cella, 2004; Grigori et al., 2004), (2) BP execution monitoring using key performance indicators (KPIs) (Castellanos et al., 2005; Golfarelli, Rizzi and Cella, 2004; Grigori et al., 2004), and (3) process mining (Aalst, 2005; Aalst et al., 2003), are some of the approaches currently used to gain insights on the actual BP behavior with regard to organization goals.

Process mining (PM) provides techniques and tools for discovering process, control, data, organizational, and social patterns from event logs (Aalst, 2005). PM has been leveraged to obtain three main types of knowledge: process discovery, process comparison, and process prediction. Process discovery aims at the generation of a process model, when one has not previously and explicitly been defined. So, it is possible to discover how people and business procedures really interact, by identifying the activities and the sequence in which they are actually executed. When the organization has a predefined model, PM enables business alignment through the

comparison of pre-defined model with their actual executions. For example, one may identify that paths originally modeled as alternative paths for representing exceptions stand for frequent procedures in the business, or even detect unexpected execution flows. Prediction is used to detect, as early as possible, undesired behaviors that require correction measures, based on historical data on process executions, using predictive mining techniques, e.g. decision trees (Castellanos et al., 2005; Han and Kamber, 2001; Tan, Steinbach and Kumar, 2006: Witten and Frank, 2005). Two PM techniques for process comparison are presented by Aalst (2005): delta analysis and conformance testing. However, the analysis and visualization of the models resulting from these techniques may be difficult, given the lack of mechanisms for model abstraction. In addition, they do not provide support for the analysis of parts of the process representing execution flows of interest (e.g. to identify the converging paths to a specific activity or the possible flows starting at it).

Web Utilization Miner (WUM) (Spiliopoulou, 2000) is an environment designed in the context of web usage mining, with the goal of gaining insight and knowledge about navigation behavior of the site users. WUM uses sequence mining to represent the observed navigation paths (i.e. sequences of page views). WUM supports the exploratory analysis of navigation flows from data on web server logs, in order to provide a deeper understanding on user behavior, page structuring and contents, enabling the comparison between expected and actual navigation behavior. It is also appropriate to discover and compare BP execution flows. Differently from (Aalst, 2005), which underlines the differences between two graphical process representations, WUM allows to limit the analysis to flows of interest, detecting similar patterns in different graph sections and quantifying processes that follow each execution pattern. Nevertheless, this approach still has limitations regarding the analysis of complex processes, revealing the need of abstraction mechanisms to address both process complexity and the different process views according to organization roles and goals.

This paper describes FlowSpy, an environment that employs the sequence mining technique proposed by Spiliopoulou (2000) to discover and analyze actual process execution paths. The innovative contribution of FlowSpy with regard to (Spiliopoulou, 2000) is the addition of abstraction mechanisms to provide user support when he or she faces difficulties in the analysis of complex processes, as well as the exploratory analysis, according to different views of the same process. The mechanisms aid the analyst in the definition of the activities of interest, which can be considered both in a) seeking for activity execution patterns, thus restricting the search space, and b) in the visualization of results, providing generalized or specialized views of the execution patterns, in an analogy with OLAP mechanisms. The paper describes the striking features of FlowSpy environment, with the focus on exploratory analysis features and the abstraction mechanisms for process behavior investigation.

The remainder of this paper is organized as follows: Section 2 summarizes the sequence mining technique of WUM (Spiliopoulou, 2000); Section 3 shows a case study on the use of WUM for BP exploratory analysis, reporting the contributions and limitations; Section 4 describes the features of FlowSpy, with an emphasis on the abstraction mechanisms; Section 5 discusses related work; and Section 6 addresses conclusions and future work.

2 WUM

Web Usage Mining is a research field that aims at extracting web page navigation patterns. The main data source is a web server log that records every access to the pages of a website. WUM is an environment (Spiliopoulou, 2000) that supports clickstream investigation through sequential mining and visualization mechanisms of navigation patterns (i.e. most frequent, rarely followed, or unexpected paths).

Pattern mining and visualization in WUM is carried out as follows. Given a web server log, page access data is initially organized as an aggregate tree, a trie tree structure, unifying navigation paths that have pages in common. The root node represents the total number of flows. Each node in the tree is represented by a triplet [P,O,A], in which P is the web page accessed, O is the occurrence of the page in the flow (e.g. first access in the click stream, second access, n-th access), and A is the number of accesses for the page in that point of the navigation trail. The arcs connecting the nodes describe the different navigation paths observed in the log.

For example, let us consider a log with 6 different flow types, shown in Fig. 1.I. The number in parenthesis represents the number of user sessions that followed along a determined flow type. The aggregate tree (Fig. 1.II) unifies these flows, based on common prefixes (e.g. to begin with page 'a' or 'b'). Revisits are shown by O > 1 (e.g. [b; 2; 6] means the second access to page 'b' observed in 6 flows). The aggregate tree enables the user to explore the log in different ways, supported by a mining language called MINT. Such seeking aims to find patterns (sub-trees) that meet a given query criteria (e.g. routes that start in page 'b' and finish in page 'e', depicted in Fig. 1.III). The resulting pattern is presented to the user also as a tree (Fig 1.IV), in which the different flows that meet the required criteria (shown in bold in Fig. 1.II) are unified.
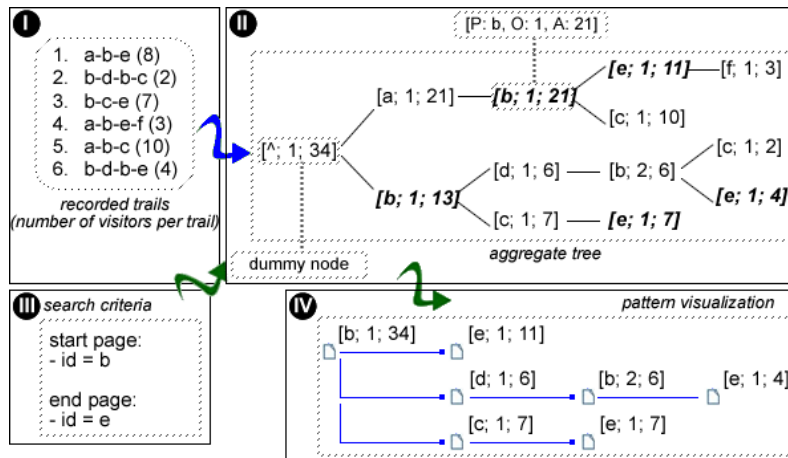

Figure 1: Mining and visualization of navigation patterns

However, the mining, analysis and interpretation of WUM results are not trivial tasks. First, in order to explore the different navigation paths, the user has to know procedural mining language MINT. Second, despite the usefulness of the exploratory analysis over different navigation paths, the analysis and interpretation of results for complex sites is difficult to carry out without the support of abstraction mechanisms over site topology and resulting sequential patterns.

3 WUM AND PROCESS MINING: A CASE STUDY

This section describes a case study to illustrate the advantages of applying to BPs the exploratory analysis based on WUM's sequence mining approach, as well as addresses the analysis difficulties practice. The process analyzed is a real workflow for Software Development Requests, which is supported by the Oracle Workflow tool. The process model is depicted in Fig. 2, and it involves 24 activities distributed in one main process and 2 sub-processes. The entry log for the generation of the aggregate tree was obtained by pre-processing the data extracted directly from workflow logs, which amounts to 1031 real instances of this process. The log arranges all activity instances of a same process instance as a sequence, ordered by activity execution start timestamp. The corresponding aggregate tree presented 34 different flows.
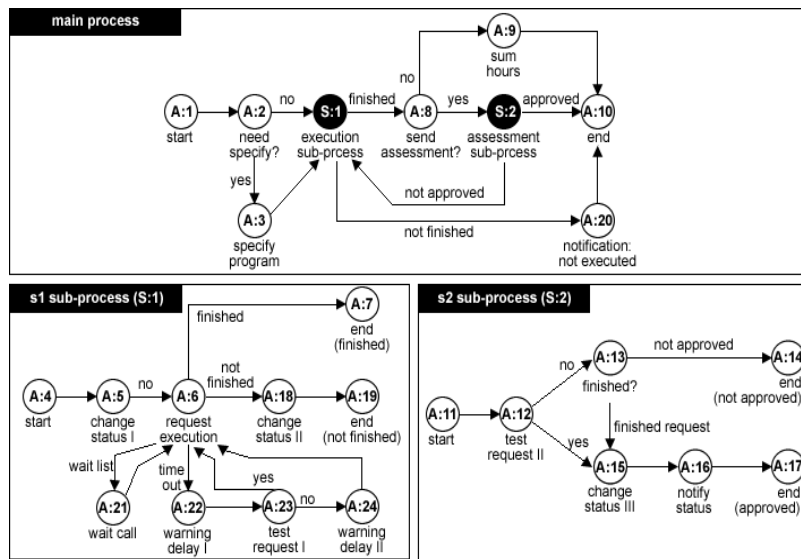

Figure 2: Case study process model

Let us suppose that the goal is to find all sequence patterns in which a request for software development process was not finished (Activity A:19 in S1 subprocess, of Fig. 2). Using the mining language, one has to define a query that seeks for all patterns that converge to activity A:20. Since the resulting pattern is rather large, Fig. 3 shows an excerpt, where the initial activities, as well as some inner activities, where omitted for legibility's sake. Activities are represented by their numerical identification. Each node is shown as a triplet [A, O, I], representing respectively activity identifier, activity occurrence in the flow, and process instances in that flow.

It is possible to observe that the pattern shown in Fig. 3 meets the restriction, as the leaf-nodes always refer to activity A:20. An expressive number of the requests started (represented in the pattern excerpt by (A:2)) were not implemented, i.e. 206 instances, as represented by the sum of the leaf instances. Hence, approximately 20% of all software development requests (I:1031) were not implemented, which may be quite a concern in a software development context. It can be also be seen that nearly 90% of these processes (I:186) followed the upper execution flow (A:2 - A:3 - A:4 - A:5 - A:6 - A:7 - … - A:19 - A:20). In the other 20 cases, 2 other flows were followed.

(A:2, O:1, I:1031) (A:3, O:1, I:617) (A:4, O:1, I:602) (A:5, O:1, I:602) (A:6, O:1, I:602) (A:7, O:1, I:586) (A:19, O:1, I:186) (A:20, O:1, I:186)

(A:18, O:1, I:11) (A:19, O:1, I:11) (A:20, O:1, I:11)

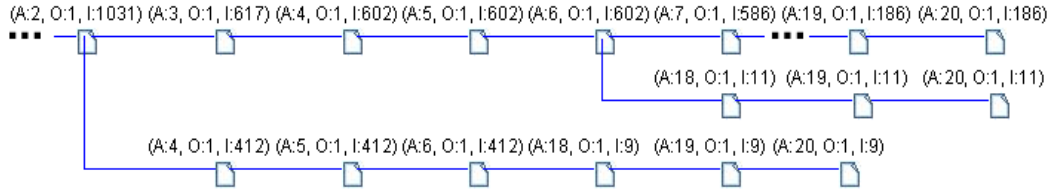(A:4, O:1, I:412) (A:5, O:1, I:412) (A:6, O:1, I:412) (A:18, O:1, I:9) (A:19, O:1, I:9) (A:20, O:1, I:9)

Figure 3: Execution pattern obtained by WUM

If the BP has a pre-defined model, it is possible to compare it to the execution pattern obtained and to verify the absence of activities expected in the model. Also, cycles ($O > 1$ in [A, O, I]) and the instances at each possible path are quantified. With this information, it is possible to check the most frequent paths, paths occurring more or less than expected, frequency of exceptions above than the expected, etc.

Moreover, the analysis of some can be harmed by the presence of flows involving large number of built-in flows, as represented by the possible flows within the sub-processes, and activities. Consequently, it is difficult to locate the activities of interest and to interpret what the pattern data reveals. These difficulties arise because the hierarchical relationships between the activities are not taken into account by the approach. Likewise, it is not possible to eliminate from the pattern the irrelevant activities, unless one preprocesses the log to remove them. Thus, the result always considers the whole flat execution tree representation, which displays sequences of activities based on timestamp order.

This case study has shown that the exploratory analysis of execution flows enables the understanding of BP behavior, allowing insights on actual process execution and enabling the comparison with expected behavior. However, the approach still imposes difficulties to analyze and interpret complex processes and execution patterns.

4 FlowSpy

FlowSpy is an environment that addresses the understanding of organization's behavior through the exploratory analysis of process executions, and sequence patterns mined from process event logs. The distinctive feature of FlowSpy with regard to approaches such as proposed by Aalst (2005) and Spiliopoulou (2000), is the provision of abstraction mechanisms to deal with process complexity and different process views. FlowSpy is based on the sequence-mining algorithm proposed by Spiliopoulou (2000), for which it provides a more user friendly, form-based query interface, instead of a complex and textual language.

FlowSpy is part of a broader scenario for BP analysis, which encompasses also process execution data capturing and preparation, together with a wide range of functionalities for analysis, monitoring and visualization of the process execution data. In this scenario, the data referring to BP logic and execution is captured, integrated and stored in an analytical repository, e.g. according to the model proposed by Grigori et al. (2004). Analysis, monitoring and mining techniques are applied upon data stored in this database. Process instances are visualized according to the business view and the type of information required. This paper addresses FlowSpy functionalities, providing the mining, analysis, and visualization of process behavior patterns. In this way, the

remaining of this section addresses the abstraction mechanisms, focused on the improvement of the data interpretation and understanding (pattern visualization), and performance of the sequential mining algorithm (Spiliopoulou, 2000) (pre-processing). Before that, it is defined the process analysis profile, used to delimit the analysis target.

4.1 Process Analysis Profiles

Process analysis profiles are composed of a set of activities that defines the particular interest of the analysis at hand. An analysis profile can be defined in terms of both (1) ad-hoc activities and (2) process sub-flows. An ad hoc activity is any activity of process P. A process sub-flow is a graph SG composed of a set of nodes N and edges D, where s the starting node and E is a set of ending nodes, given s ∈ N and E ⊂ N. SG is a connected graph, and all edges in D connect only nodes n ∈ N. This definition is quite similar to the concept of process region proposed by Grigori et al. (2004). Notice that a sub-process is a type of sub-flow. An analysis profile can be defined in an inclusive or exclusive manner, just before the use of the abstraction mechanisms (log pre-processing and pattern visualization abstraction). Thus, the user may define the analysis profile either in terms of the specific activities and sub-flows of interest, or the ones which should be disregarded. In addition, the user has operations to define sub-flows. When the process model is available, the interface presents the existing sub-processes to the user. On the other hand, to define an arbitrary sub-flow, the user selects the initial node, and interactively FlowSpy displays the adjacent nodes (i.e. the ones that can be immediately reached from it), which can then be selected by the user, recursively, until he or she defines the sub-flow final node(s). If the process model is available, process structure is used to indicate the adjacent nodes. If not, the possibilities are derived from the sequences of activities observed in the log. Once profiles have been defined, they can be explored for both pre-processing the logs and during pattern analysis.

Fig. 4 shows three examples of analysis profiles, which represent the structure of the process shaped in Fig. 2. The analysis profile P1 represents the activities of the S1 sub-process and the profile P2 the activities of the S2 sub-process (both contains only ad-hoc activities). The P0 profile represents the activities and sub-processes (profiles P1 and P2) of the main process.
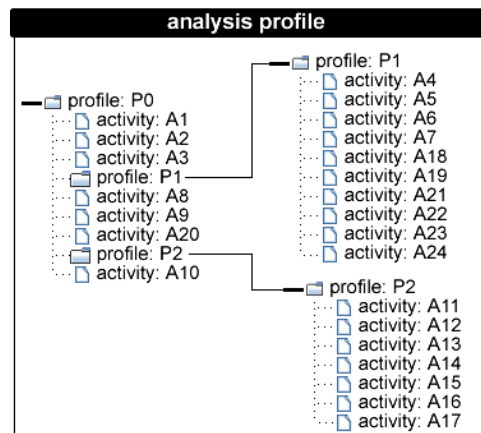

Figure 4: Analysis Profile

4.2 Abstraction Mechanisms

Log pre-processing. Log pre-processing aims at generating a shorter aggregate tree, containing only the activities/sub-flows of interest, as represented by a given analysis profile. If the analysis profile is applied of exclusive form, pre-processing involves removing from the log, prior to the construction of the aggregate tree, all activities contained in a given analysis profile. Also, all sets of activity instances representing the sub-flow(s) of the profile must be replaced by a single entry in the log, representing the sub-flow as a whole, of which the information is the one of the corresponding starting node. For example, both the process shaped in the Fig. 5.I, and its corresponding tree (Fig. 5.II), were produced from the complete load of processes execution logs. For hypothesis, the goal is to verify the execution behavior of this process when activities 6, 7, 8, 9, 10 and 11 are disregarded. It is used the analysis profile P3 that contains these activities. Fig. 5.III shows the simplification of the resulting aggregate tree, after pre-processing the execution log using P3. Activities of the resulting tree, belonging to P3 profile, are replaced by P3-named nodes. As a consequence of using the abstraction during log pre-processing, the mining task becomes lighter due to a shorter tree. In such a way, the user can handle a smaller set of activities to define the mining query and thus the resulting pattern will have fewer activities.
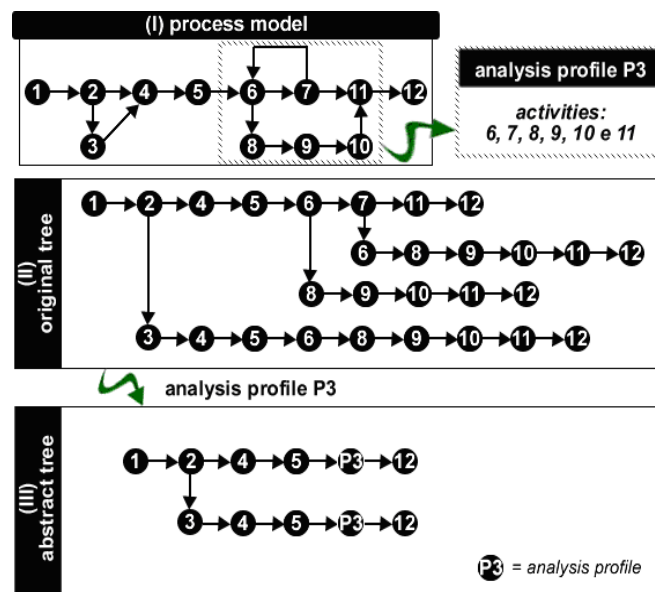

Figure 5: Log pre-processing

Pattern visualization abstraction. Pattern visualization abstraction involves simplifying an analysis pattern to improve its interpretation. The idea is analogous to the drill-up and drill-down operations commonly used by OLAP to increase or decrease the detail level of the flows represented by the pattern. The simplification can be based on a pre-defined analysis profile, or interactively. In the former case, the resulting pattern is simplified by eliminating the activities and by substituting subtrees of the pattern by an atomic node. In the latter, users interactively indicate tree nodes that should be removed (which can correspond to either an atomic process or previously abstracted sub-flows), as well as sub-processes that should be aggregated.

The visualization can be produced in two forms: aggregation and exclusion. In the aggregation, the resulting graph is a simplified execution pattern by replacing activities and sub-flows, belonging to an analysis profile, by an atomic node. Fig. 6 shows an example of aggregation. In this example, the pattern illustrated in Fig. 3 (Fig. 6.I) is simplified by the application of the analysis profile P1, making easier its interpretation. In such a way, all the activities related to the sub-process S1 (A:4 A:5 A:6 A:7,…A:18 A:19) are grouped in one node ([P1, I]), as illustrated for the Fig. 6.II. In the exclusion, the user indicates, interactively, the tree nodes that must be hidden by the visualization. These nodes can correspond to the atomic nodes or abstracted sub-flows (by aggregation). Fig. 6.III depicts an example of exclusion, taking as input the resulting pattern of Fig. 6.II. In this example, it is assumed that the user desires to hide activity 3 (A: 3). After hiding A:3, it must be verified, in the new pattern, if it is possible to reduce the pattern combining edges linking nodes with the same activity identifier. In the example, the nodes (P1, O:1, I:602) and (P1, O:1, I:412) in Fig. 6.II were reduced to node (P1, O:1, I:1014) in Fig. 6.III.
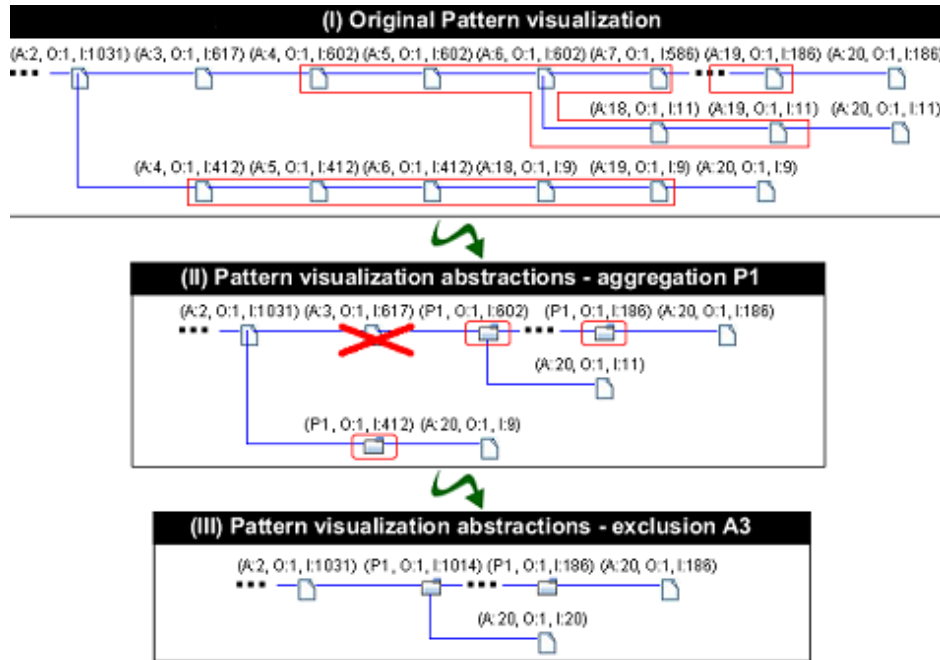

Figure 6: Pattern visualization abstractions

The current implementation of the prototype permits the user to change the way he or she is performing his/her analyses. Starting from the structural analysis as depicted in Fig. 6, it is possible to swap to a quantitative analysis. Fig. 7 shows the pattern illustrated in Fig. 3 by the prototype interface. After selecting the node (A:3; O:1; I:617), the user can click on Quantitative Analysis guide, on the top of the dialog. Then, the user is redirected to some dialogs where he or she selects a metric from a predefined set. In the example, Count of Process Instances by Taxonomy. As a result, the prototype presents Fig. 8 dialog, where the user can check that 15.5% of instances are fast, 60.5%

are acceptable and 24% are slow, according to the predefined KPI (Key Process Indicator). Tristão (2007) presents more details of the implemented prototype.
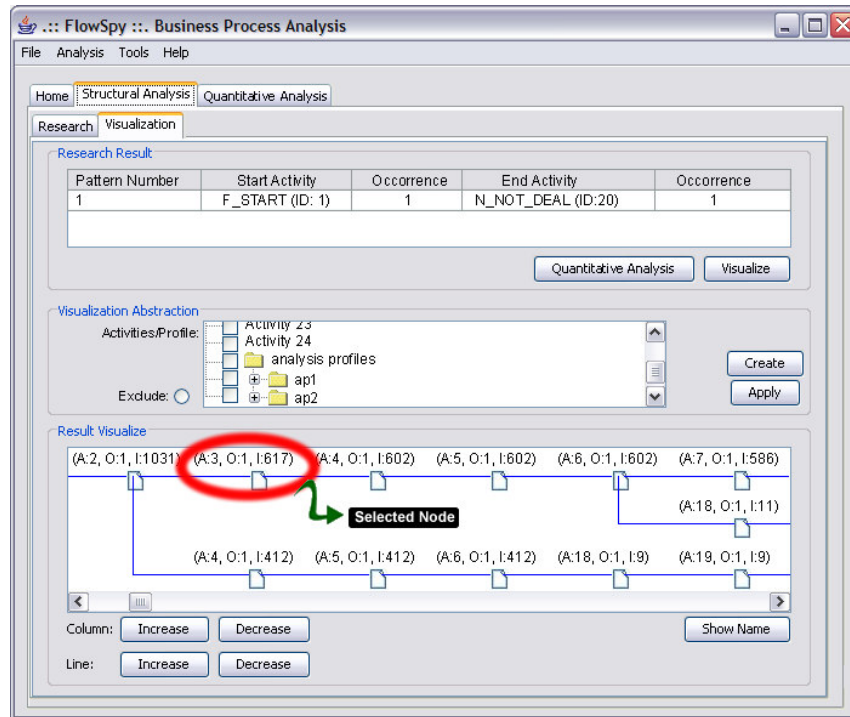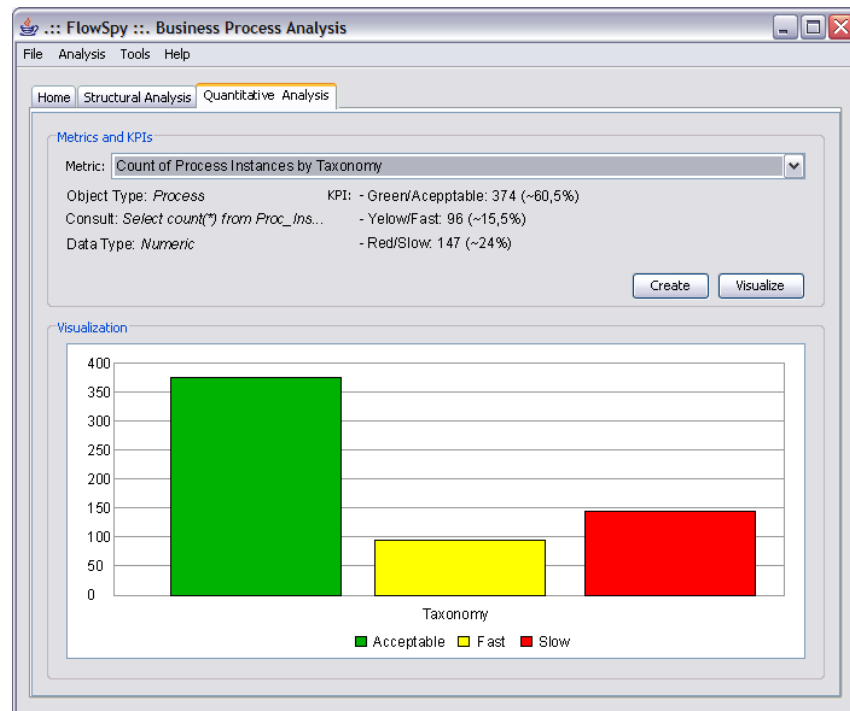

Figure 7: Prototype Dialog for the Structural Analysis.

Figure 8: Prototype Dialog for the Quantitative Analysis.

5 RELATED WORK

As mentioned, process mining has been addressed three main purposes: process discovery, process comparison and process prediction. Works such as (Aalst et al., 2003) address process mining focusing on the discovery of workflow models and its inherent issues, such as execution cycles. Petri Nets are the formalisms most used for this purpose. The main challenges are related to obtain workflow logs that store information of the nature of events and on the transition between activities. This issue is dealt in FlowSpy by the use of activities timestamps, and process instance identifiers to which each activity instance belongs to. In addition, our approach does not attempt to produce an abstract representation of the process in the form of a graph representation. Instead, we present the process structure in terms of a flat tree. The exploratory analysis helps analysts to focus on the parts of interest.

Process comparison is addressed by Aalst (2005), in a research focused on measuring business alignment, i.e. comparing the actual behavior with the expected one of the information system. For this purpose, two techniques are proposed: delta analysis and conformance testing. These techniques compare two graph representations of processes, and do not provide support for the analysis of segments of the process representing execution flows of interest. They lack abstraction mechanisms to deal with complex process or different views of a same process, and do not focus on expressing the representative of each possible flow.

Section 3 discussed the contribution of WUM (Spiliopoulou, 2000) to both process discovery and comparison, through exploratory search of execution flows having specific properties. However, such approach lacks abstraction mechanisms to produce more useful patterns, as well as to make easier the interpretation of them.

Process mining plays a crucial role in the Business Intelligence context (Golfarelli, Rizzi and Cella, 2004). Business Process Intelligence (BPI) (Grigori et al., 2004) is an environment to support the analysis, monitoring, and prediction of processes restricted to workflows produced using a specific WfMS. BPI has a Process Mining Engine, among other components, and its goal is to establish predictive models of process behavior, using classification algorithms. To deal with process abstractions, the concept of Process Region is proposed, and it is used to select desired segments from process instances. This concept is employed in FlowSpy to define analysis profiles. However, we assume that processes do not necessarily have a pre-defined model, and therefore, users may not be able to define sub-flows from both the process model and process instances, as represented by the log. iBOM (Castellanos et al., 2005) is an evolution of BPI. One of the main differences lies in the capture of process events according to different abstraction levels, considering, in addition, a heterogeneous process management environment. FlowSpy does not address application events capturing, assuming that they are captured and stored in the log with a specific format. However, FlowSpy provides different abstraction levels using the pre-processing and visualization abstraction mechanisms. Both BPI and iBOM are designed to produce summaries of processes using OLAP and indicators. FlowSpy is part of a broader Business Intelligence environment, and the idea is to establish a synergic coupling between execution flows and performance summaries. Issues related to process

repository design and process event capturing, not explicitly addressed in this paper, are discussed by Grigori et al. (2004), List and Machaczek (2004) and Schiefer et al. (2004).

6 CONCLUSION AND FUTURE WORK

FlowSpy is an environment for business process mining. Differently from Aalst (2005), FlowSpy focuses on exploratory analysis of the different execution flows, enabling a detailed analysis of business behavior, quantification of different execution flows, and abstraction mechanisms that deal with process complexity and different process views. This approach is suitable for both process comparison and process discovery, since it does not assume a pre-defined model. The use of Web Usage Mining sequence analysis has allowed the tracking of activity executions accurately. It is thus possible to identify the activities, or resources, that lead to undesired execution flows, to find the different execution flows that converge towards a given activity, and to validate the process model by the identification of convergence between activities (probability of execution). Therefore, business behavior can be better understood.

The Abstraction Manager is the innovative component of FlowSpy, in comparison to WUM (Spiliopoulou, 2000). Two main abstraction mechanisms are available: log pre-processing and visualization abstraction. The former aims at improving the data mining phase, with a more restricted aggregate tree. The latter facilitates pattern interpretation by producing on demand, more detailed or generic trees, representing the obtained patterns.

Currently FlowSpy implements the mining algorithm of Spiliopoulou (2000), provides a form-based interface for mining, and allows the tree visualization abstraction described in Section 4. The patterns displayed in Figs. 3, 4, 6 and 7 were produced using the current implementation. We are now implementing the interfaces for log pre-processing tools and to incorporate process metadata.

The tools and environments available in the software market are focused on data integration, statistical process summarization and KPI managers. However, the analysis and quantification of a detailed execution flow of activities are not addressed. Hence, the status of our research is to study how to combine these issues in a synergic approach, and how to incorporate the resources to analyze and monitor BPs into our prototype. Given an execution flow, the idea is to verify its performance using pre-defined metrics targeted to meet the organization's goals. Also, once a performance metric is defined, its behavior may be interpreted by analyzing instance flows. Another possible subject of study is the development of a data storage structure (aggregate tree) by means of performance metrics. Such structure may then be used as an execution model to predict results. Thus, it is expected that FlowSpy meet the three ways through which mining knowledge is obtained nowadays.

Considering FlowSpy applies a sequence-mining algorithm originally proposed for web use, web-based studies using FlowSpy may be done. Thus, another topic is the use of the proposed abstraction mechanisms to improve the mining, visualization, and analysis of site topologies and user navigation behaviors.

References

Aalst, W.M.P. van der. (2005) "Business Alignment: Using Process Mining as a Tool for Delta Analysis and Conformance Testing". Requirements Engineering Journal, 10(3), pp. 198-211. Nov.

Aalst, W.M.P. van der; Dongen, B.F. van; Herbst, J.; Maruster, L.; Schimm, G.; Weijters, A.J.M.M. (2003) "Workflow Mining: A Survey of Issues and Approaches". Data and Knowledge Engineering, 47(2). pp. 237-267.

Casati, F. (2005) "Industry Trends in Business Process Management: Getting Ready for Prime Time". International Workshop on Database and Expert Systems and Applications (DEXA'05), 16., 22-26 Aug. 2005, Copenhagen. **Proceedings…** Copenhagen: IEEE Computer Society, Aug. pp. 903-907.

Castellanos, M.; Casati, F.; Ming-Chien Shan; Dayal, U. (2005) "iBOM: A Platform for Intelligent Business Operation Management". International Conference on Data Engineering (ICDE 2005), 21., 5-8 Apr. 2005, Tokyo. **Proceedings…** Tokyo: IEEE Computer Society, Apr. pp. 1084-1095.

Golfarelli, M.; Rizzi, S.; Cella, L. (2004) "Beyond data warehousing: what's next in business intelligence?". 7th ACM International Workshop on Data Warehousing and OLAP, 7., Nov. 2004, Washington. **Proceedings...** New York: ACM Press, Nov. Nov. pp. 1-6.

Grigori, D., Casati, F., Castellanos, M., Dayal, U., Sayal, M. and Shan, M. C. (2004) "Business Process Intelligence". Computers in Industry, 53(3), pp. 321-343. Apr.

Han, J.; Kamber, M. (2001) "Data mining : concepts and techniques". San Francisco, CA : Morgan Kaufmann. 550 p.

List, B. and Machaczek, K. (2004) "Towards a Corporate Performance Measurement System". ACM Symposium of Applied Computing, 19., 14-17 Mar. 2004, Nicosia. **Proceedings…** New York: ACM Press, Mar. pp. 1344-1350.

Schiefer, J.; Jeng, J.; kapoor, S.; Chowdhary, P. (2004) "Process information factory: a data management approach for enhancing business process intelligence". IEEE International Conference on e-Commerce Technology (CEC'04). 6-9 July 2004, San Diego. **Proceedings...** San Diego: IEEE Computer Society,. pp. 162-169.

Spiliopoulou, M. (2000) "Web Usage Mining for Site Evaluation. Making a site better fit its users". Communications of the ACM, 43(8), pp. 127-134. Aug.

Tan, P.; Steinbach, M.; Kumar, V. (2006) "Introduction to data mining". Boston : Addison-Wesley, 769 p.

Tristão, C. (2007) "An Integrated Environment for Business Process Analyses". Porto Alegre: PPGCC-PUCRS, 68 p. (in Portuguese)

Witten, I. H.; Frank, E. (2005) "Data Mining: Practical Machine Learning Tools and Techniques" (Second Edition). San Francisco, CA: Morgan Kaufmann. 525 p.