FlowSpy: exploring Activity-Execution Patterns from Business Processes

Cristian Tristão[1], Duncan D. Ruiz[2], Karin Becker[3]

[1]Departamento de Informática - PUC-Rio
Rua Marquês de São Vicente 225, 22453-901 - Rio de Janeiro, RJ - Brazil

[2]Faculdade de Informática - PUCRS
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. *This paper describes FlowSpy, an environment that addresses the understanding of business process behavior by combining the exploratory analysis of process executions and process key performance indicators. FlowSpy employs a sequence mining technique to discover and analyze the actual execution paths of business processes. It supports the detailed analysis of business behavior and the quantification of different execution flows, and offers abstraction mechanisms to deal with process complexity and different process views. FlowSpy has also features for the synergic exploration of information originated from both sequential mining techniques and measurements of processes, activities and resources. FlowSpy is part of a broader scenario for business process analysis, which also encompasses the capturing and preparation of process execution data, together with a wide range of functionalities for analysis, monitoring and visualization of such data.*

1 Introduction

Business processes (BPs) are increasingly automated and controlled by information systems, such as Workflow Management Systems (WfMS), ERP and CRM. The systematic use of these systems creates and stores huge amounts of data, which reflect the actual state and behavior of BPs. Lately, organizations have shifted the focus from process automation and control, to the measurement, analysis, monitoring and prediction of BPs (Casati, 2005; Golfarelli, 2005; Golfarelli, Rizzi and Cella, 2004). Analyses based on (1) the summarization of BP historical data (Castellanos et al., 2005; Golfarelli, 2005; Golfarelli, Rizzi and Cella, 2004; Grigori et al., 2004), (2) BP execution monitoring using key performance indicators (KPIs) (Castellanos et al., 2005; Golfarelli, 2005; Golfarelli, Rizzi and Cella, 2004; Grigori et al., 2004), and (3) process mining (Aalst, 2008; Rozinat and Aalst, 2008; Song and Aalst, 2008; Aalst and Günther, 2007; Aalst, 2005; Aalst et al., 2003), are some of the approaches currently used to gain insights on the actual behavior of BPs 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, 2008; Aalst et al., 2007). PM has been leveraged to obtain three main types of knowledge: (1) process discovery, (2) process comparison, and (3) process prediction. These three aspects are referred to as discovery, conformance, and extension by Aalst et al (Aalst, 2007; Aalst et al., 2007; Aalst and Günther, 2007; and Song and Aalst, 2008). *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 (Rozinat and Aalst, 2006; Castellanos et al., 2005; Han and Kamber, 2006; Tan, Steinbach and Kumar, 2005: Witten and Frank, 2005).

Two PM techniques for process comparison are presented by Aalst (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). Aalst and Günther (Aalst and Günther, 2007) suggest concepts to simplify and present “spaghetti-like models” used in the field of cartography to analysis complex topologies, such as: (1) aggregation, (2) abstraction, (3) emphasis, and (4) customization. However, there is no support for the analysis of parts of the process representing execution flows of interest.

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 (Tristão et al., 2008). Differently from works that highlight the differences between two graphical process representations (e.g. Aalst, 2005; Aalst and Günther, 2007), WUM allows the analysis to be limited to flows of interest. In this way, it is possible to detect similar patterns in different graph sections and quantify the processes that follow each execution pattern. Nevertheless, this approach still has limitations regarding the analysis of complex processes, revealing the need for abstraction mechanisms that address both process complexity and the different process views according to organization roles and goals.

This paper describes FlowSpy (Tristão et al., 2008), an environment that employs the sequence mining technique proposed by Spiliopoulou (Spiliopoulou, 2000) to discover and analyze the actual execution paths of BPs. The striking contribution of FlowSpy with regard to (Spiliopoulou, 2000) is the addition of abstraction mechanisms that deals with process complexity and which allow exploratory analysis, according to different views of the same process. These 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. Inserted in a broader BP analysis scenario, FlowSpy enables information exploration from sequential mining results and BP execution metrics. The paper describes the striking features of FlowSpy environment, providing further details on the features for exploratory analysis, the abstraction mechanisms for process behavior investigation and the approach for information integration. Some of these features were originally presented in (Tristão et al., 2008).

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 striking features of FlowSpy; 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 (Spiliopoulou, 2000) is an environment 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 developed as follows. Given a web server log, page access data is initially organized as an aggregate tree, a trie tree structure that unifies 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 clickstream, 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 types of flow, shown in Fig. 1.I. The number in parenthesis represents the number of user sessions that match a given 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’, which is observed in 6 flows). WUM provides the mining language MINT, which allows users to specify navigation patterns through a query criteria (e.g. routes that start in page ‘b’ and finish in page ‘e’, as depicted in Fig. 1.III). The resulting pattern is presented to the user also as a tree (Fig 1.IV), which unifies the different flows that meet the specified criteria (shown in bold in Fig. 1.II).

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 the 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 develop without the support of abstraction mechanisms over site topology and resulting sequential patterns.

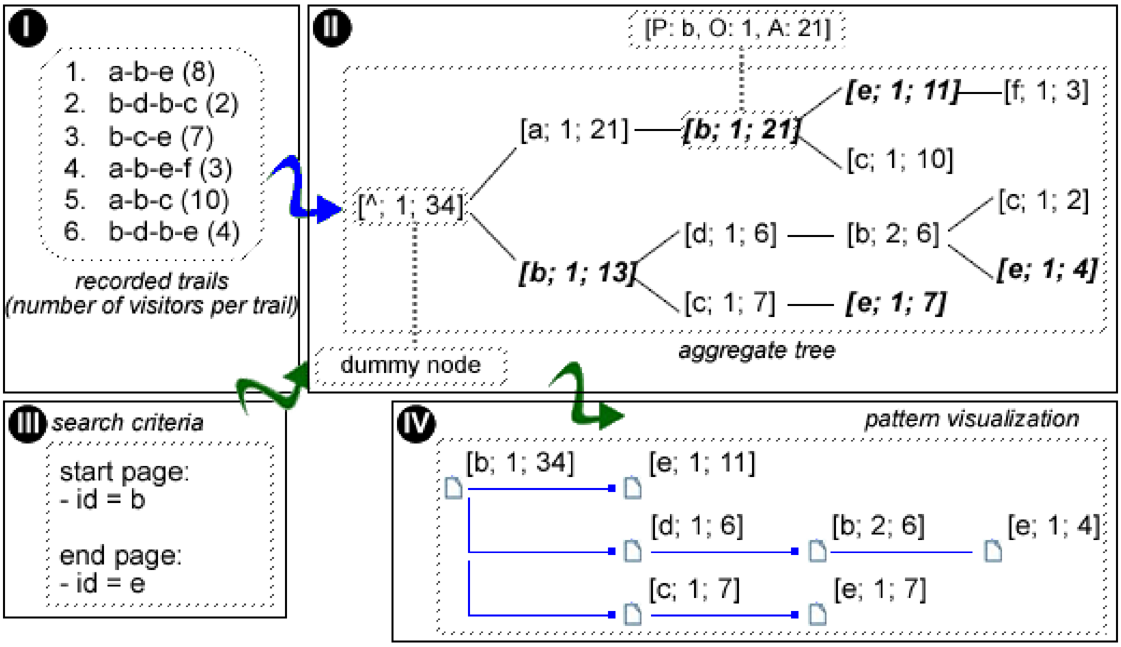

Figure 1: Mining and visualization of navigation patterns

3 WUM AND PROCESS MINING: A CASE STUDY

This section describes a case study that illustrates the advantages of applying to BPs the exploratory analysis based on WUM's sequence mining approach, together with the difficulties one may face in 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 types of flow.

Let us suppose that the goal is to find all sequence patterns in which a request for software development process was not finished, which corresponds to Activity A:19 in S1 sub process, leading to activity A:20 in the main process. Hence, using the mining language, one defines a query that seeks for all patterns that converge to activity A:20. The result is shown in Figure 3, where 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. Because the resulting pattern is very complex, Fig. 3 shows only an excerpt, where the initial activities, as well as some inner activities, were omitted for legibility's sake.

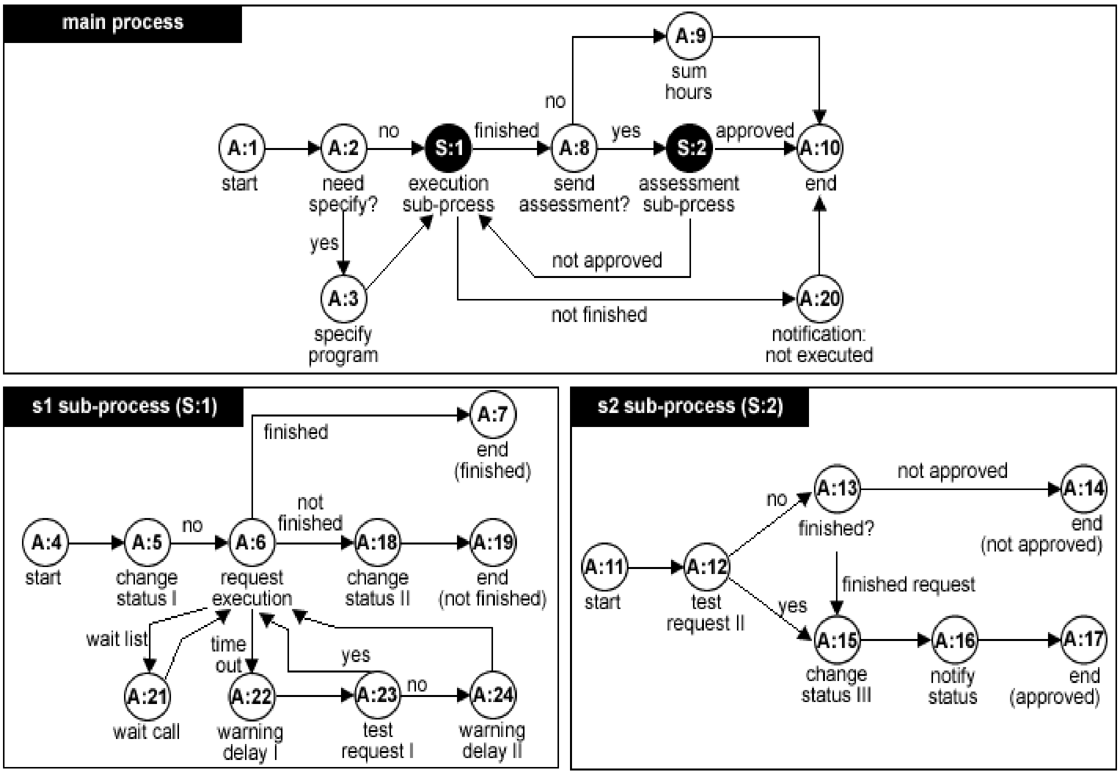

Figure 2: Case study process model

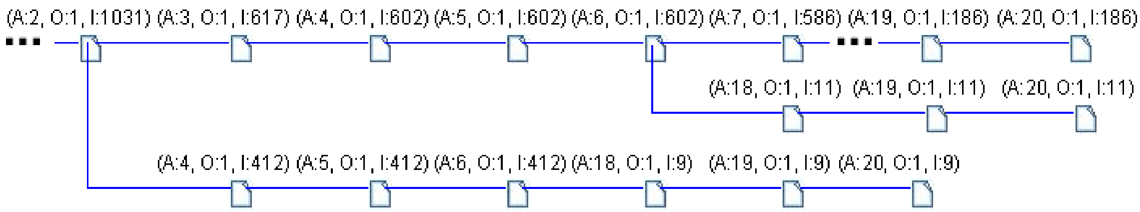

Figure 3: Excerpt of the execution pattern obtained by WUM

All patterns shown in Fig. 3 meet the restriction, as the leaf-nodes always refer to activity A:20. Node A2 in Fig. 3 represent all software development requests started, which amount to 1031 process instances. By summing the instances associated with leaf of the resulting tree (206 instances), one can reach the conclusion that approximately 20% of all software development requests 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 types of flows were followed.

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

The visualization and analysis of some patterns can be jeopardized 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 actually reveals. In this case study, the pre-processing flattens all the hierarchical process/sub-process relationships between the activities in order not to loose any possible important aspect of the process. It is not possible to eliminate from the pattern the irrelevant activities, unless one re-preprocesses the log to remove them.

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 on the analysis and interpretation of complex processes and execution patterns.

4 FlowSpy

FlowSpy is a support environment for business processes analysis that addresses the understanding of organization’s behavior by exploring the synergy between the exploratory analysis of process executions and process KPIs. FlowSpy is based on the sequence-mining algorithm proposed by Spiliopoulou (Spiliopoulou, 2000), for which it provides a more user friendly, form-based query interface, instead of a complex and textual language. The distinctive feature of FlowSpy with regard to approaches such as (Aalst and Günther, 2007; Aalst, 2005; and Spiliopoulou, 2000) is the provision of abstraction mechanisms to deal with process complexity and different process views. Another distinctive feature is that it also combines the information originated from the application of sequential mining techniques, with metrics from processes, activities and resources, measured according to KPIs.

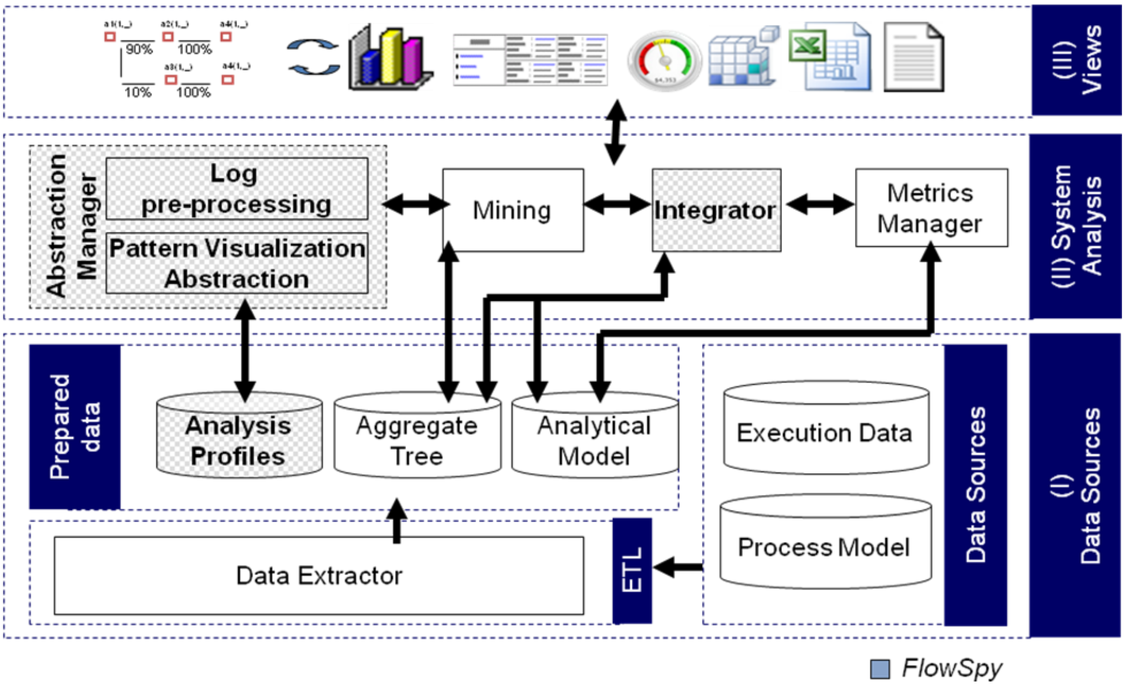

Figure 4: BP analysis architecture

FlowSpy is part of a broader scenario for BP analysis depicted in Fig. 4, which encompasses also: (I) process execution data capturing and preparation, together with a wide range of functionalities for (II) analysis, monitoring and (III) 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, according to some process analytical models (e.g. Grigori et al., 2004; Casati et al., 2007). 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 section addresses FlowSpy functionalities, providing the mining, analysis, and visualization of process behavior patterns and KPIs. The remaining of this section addresses the abstraction mechanisms, focused on the improvement of the data interpretation and understanding (pattern visualization), performance of the sequential mining algorithm (Spiliopoulou, 2000) (pre-processing), and information integration. FlowSpy also allows the definition of process analysis profiles to delimit the analysis target.

4.1 Process Analysis Profiles

Process analysis profiles are composed of the set of activities that define 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 is the starting node and E is a set of ending nodes, given $s \in N$ and $E \subset N$. SG is a connected graph, and all edges in D connect only nodes $n \in N$. This definition is quite similar to the concept of process region proposed by Grigori et al. (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 that 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 pattern analysis.

Fig. 5 shows three examples of analysis profiles, which represent the structure of the process displayed 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.

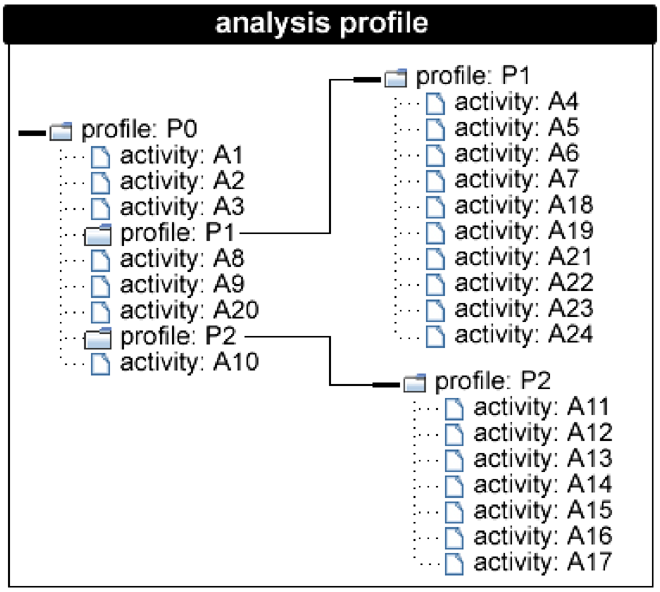

Figure 5: Analysis Profile

4.2 Abstraction Manager

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 in the exclusive form, pre-processing removes all activities contained in a given analysis profile from the log, prior to the construction of the aggregate tree. 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 depicted in the Fig. 6.I, and its corresponding tree (Fig. 6.II), were produced from the complete load of processes execution logs. Suppose that the goal is to verify the execution behavior of this process disregarding activities 6, 7, 8, 9, 10 and 11. The profile P3 that contains these activities is used to remove these activities from the log. Fig. 6.III shows the simplified resulting aggregate tree, after pre-processing the execution log according to exclusive P3. Activities of the P3 profile are replaced in the resulting tree by P3-named nodes. The consequence of using the abstraction for log pre-processing is that the mining task becomes lighter due to a smaller tree, and consequently the user can handle a smaller set of activities to define the mining query. Likewise, the resulting pattern will have fewer activities.

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 mechanisms 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 sub-trees 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.

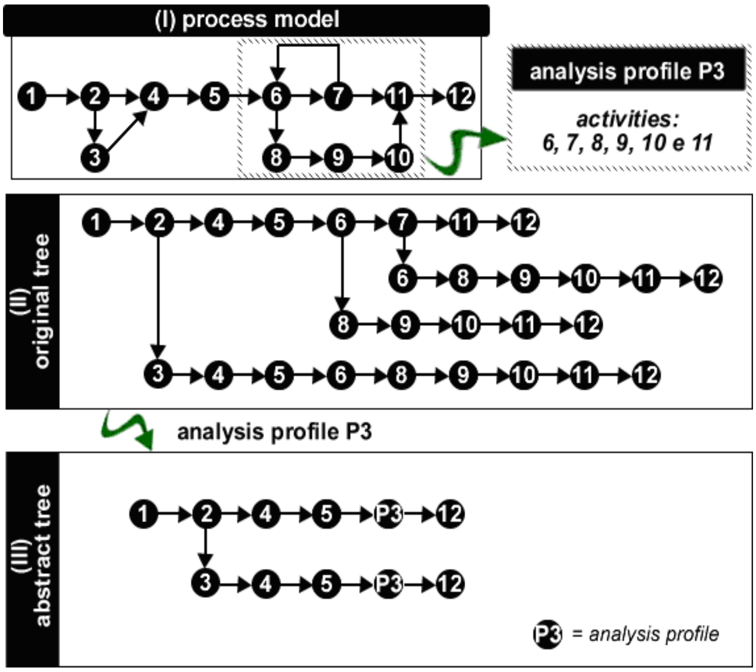

Figure 6: Log pre-processing

The visualization can be produced in two forms: aggregation and removal. In the aggregation, the resulting graph is a simplified execution pattern that replaces 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. 7.I) is simplified by the application of the analysis profile P1 (Fig. 5), making its interpretation easier. Thus, all the activities related to the sub-process S1 (A:4 … A:19) are grouped in one node ([P1, I]), as illustrated for the Fig. 7.II. The user can also remove nodes from the visualization tree interactively. These nodes can correspond to the atomic nodes or abstracted sub-flows (by aggregation). Fig. 7.III depicts an example of removal. In this example, the user simplifies the resulting pattern of Fig. 7.II removing the activity 3 (A: 3). After node removal, it must be verified in the new pattern if it is possible to reduce it by 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. 7.II were reduced to node (P1, O:1, I:1014) in Fig. 7.III.

4.3 Integrator

The integrator is the component responsible for providing the synergy between the information of web-usage sequence mining technique and the quantitative analysis of business process data. Thus, from a process execution flow, it is possible to verify related process instances performance through indicators that express company goals, and vice-versa (we can try to understand the behavior of their instances by flow analysis). The way FlowSpy integrates information across different forms of business processes analyses is summarized by Fig. 8. The basic idea is that, given a node of interest, it is possible to derive the KPI for the set of process instances represented by that node. Conversely, given a set of process instances that present a certain performance indicator, it is possible to analyze the specific flow of these instances.

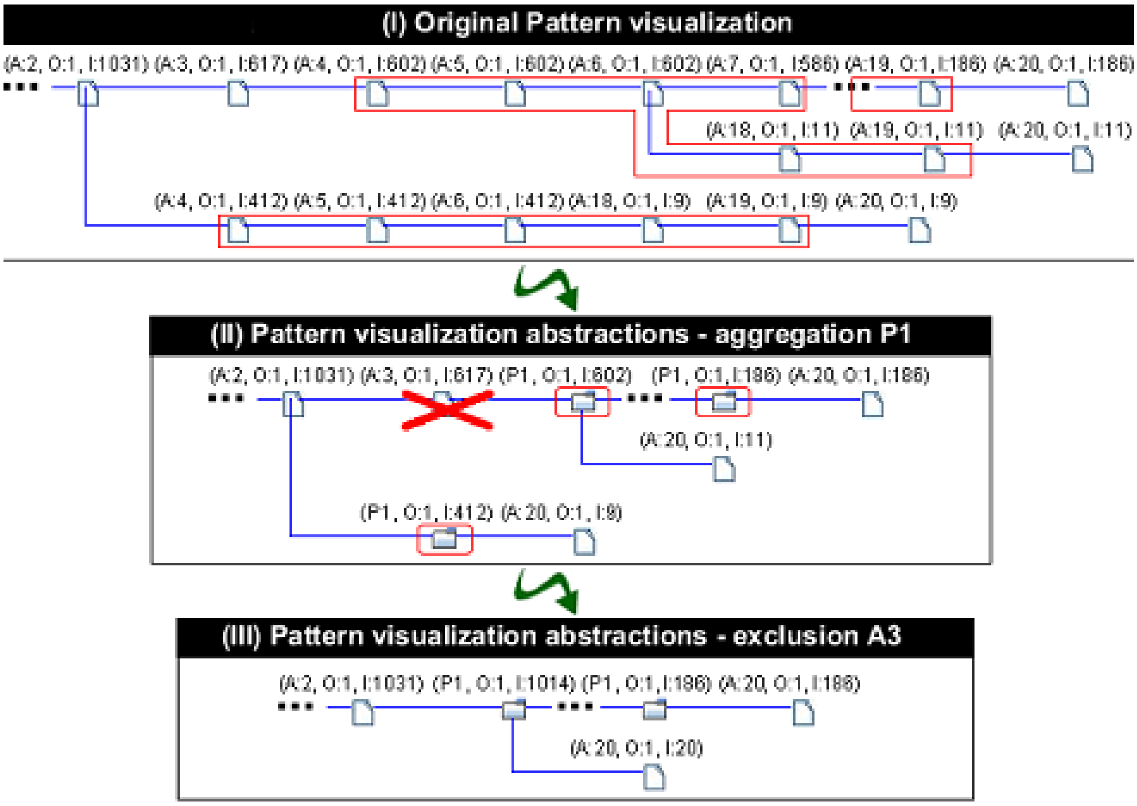

Figure 7: Pattern visualization abstractions

One of the main difficulties when applying the sequential mining technique is to identify, in each node, which process instances executed a specific flow activity. Our solution is to store the set of process instances that correspond to each pattern node (the activity in flow). Then, when a user selects an instance set, he/she can perform quantitative analysis according to the desired goal and corresponding indicator.

The structural analysis performed from the quantitative analysis adopts the same idea. When selecting a performance level, it is produced a sequential structure containing only the corresponding instances.

To allow the information integration, the data structure that stores the aggregate tree, the process instances execution pattern and KPI, must record at all times the set of related process instances. For example, In Fig. 8.I, it is depicted all process instances that are associated to each node in the pattern tree. Hence, Node A is associated to 6 instances (instances 1, 2, 3, 4, 5 and 6), whereas node E is representative of two instances (instances 1 and 3). Likewise, each KPI has the set of associated process instances for each performance level. In the example of Fig. 8.II, the green performance level KPI is related instances 02, 04 and 06. Such instances are subject to structural analysis to verify possible reasons of this behavior.

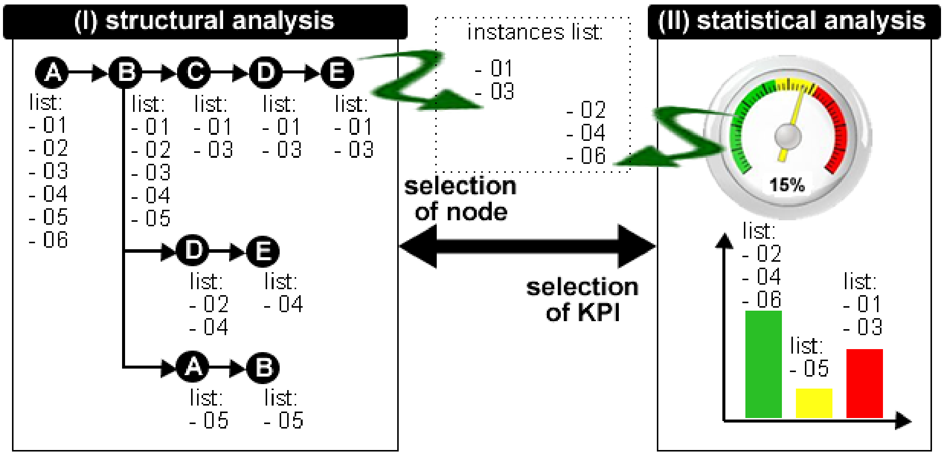

Figure 8: Integrator

The current implementation of the prototype allows the user to switch easily among the two types of analysis: starting from the structural analysis according to execution flows, possibly by applying abstraction mechanisms as discussed in Section 4.2, it is possible to swap to a quantitative analysis according to KPIs, and then back-and-forth, as sketched in Figure 8.

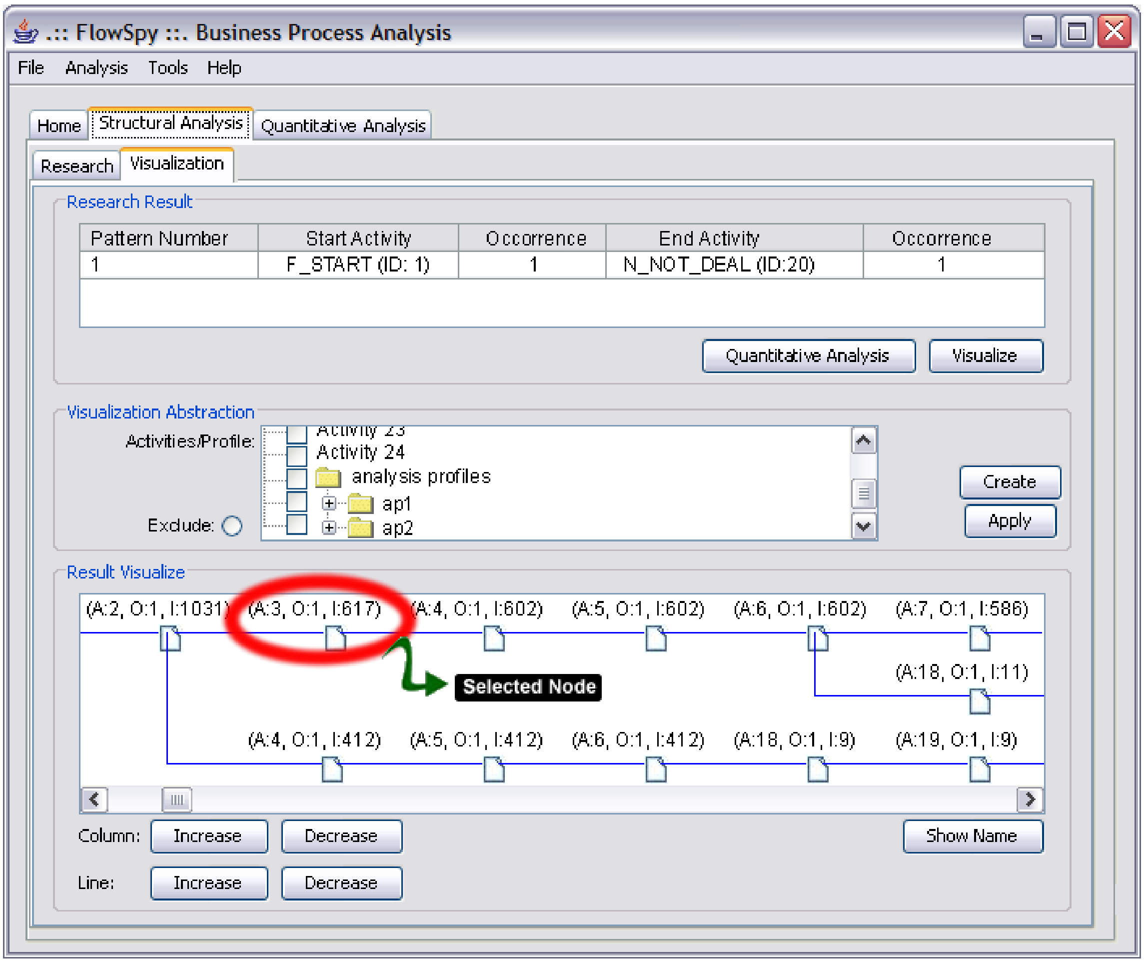

Figure 9: Prototype Dialog for the Structural Analysis.

Fig. 9 shows the pattern illustrated in Fig. 3, as presented in the prototype interface. After selecting a node (A:3; O:1; I:617), the user can click on the Quantitative Analysis guide, on the top of the dialog window. Then, the user is redirected to some dialogs where a metric can be selected from a predefined set, e.g. Count of Process Instances by Taxonomy. As a result, the prototype presents Fig. 10 window with the selected KPI, where the user can verify that 15.5% of instances are classified as "fast", 60.5% as "acceptable" and 24% as "slow". Tristão (Tristão, 2007) presents further details of the implemented prototype.

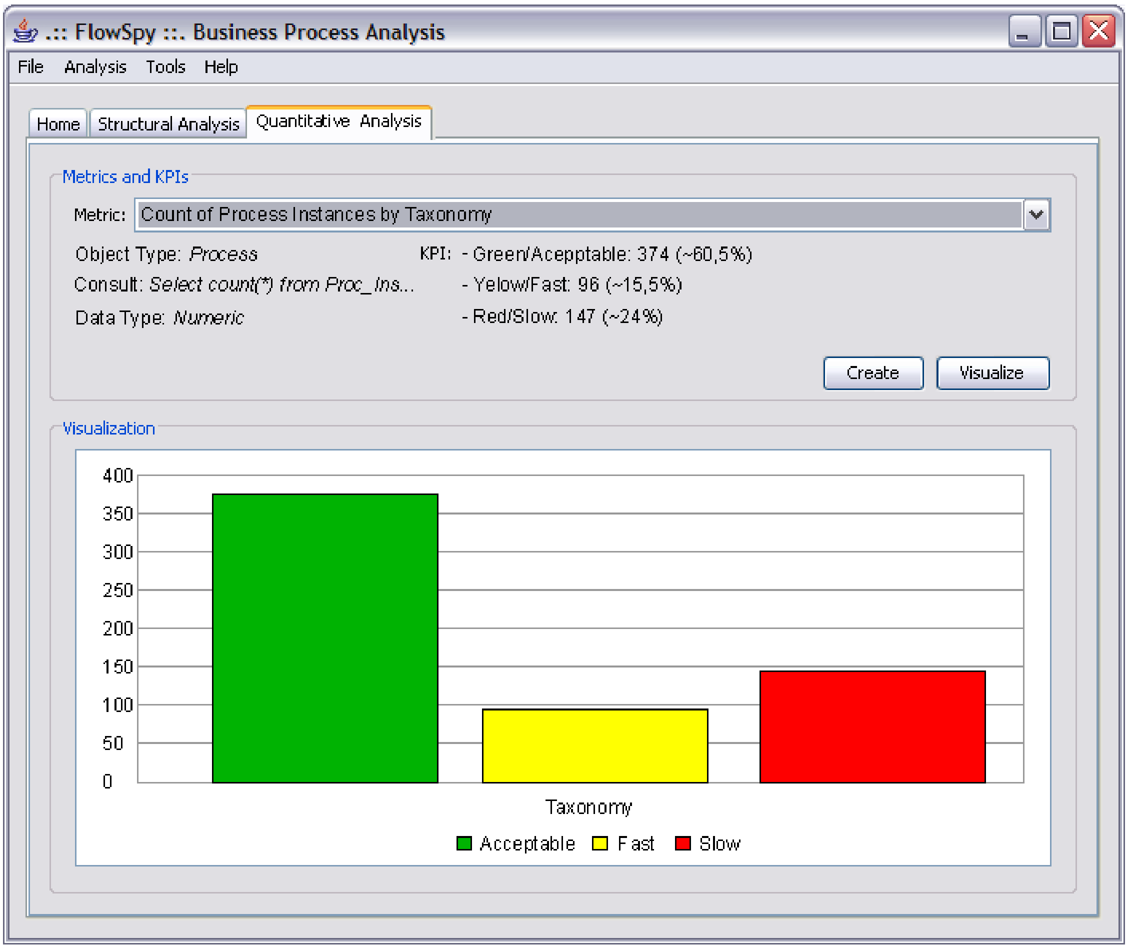

Figure 10: Prototype Dialog for the Quantitative Analysis.

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 is the most used formalism for this purpose (Rozinat and Aalst, 2008; Aalst et al., 2007; Aalst, 2007; Aalst and Günther, 2007). 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 surrogates that identify the process to which each activity instance belongs. 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 (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 a process, 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. Aalst and Günther (Aalst and Günther, 2007) suggest some concepts to simplify complex models or present different views. Rozinat and Aalst (Rozinat and Aalst, 2008) propose an incremental approach to check the conformance of a process model and an event log: (1) fitness and (2) appropriateness.

With few adaptations for the business context, WUM environment (Spiliopoulou, 2000) could be employed to both process discovery and comparison, through exploratory search of execution flows having specific properties. However, WUM lacks abstraction mechanisms to produce more useful patterns, as well as to make easier the interpretation of them, which is required for complex processes, or process models using sub-processes.

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. Depending on the information contained in the log and with the assistance of the abstraction mechanisms, FlowSpy enable the process analysis under different perspectives. Aalst et al. (Aalst et al., 2007) distinguish three different perspectives: (1) the process perspective (“How?”), (2) the organizational perspective (“Who?”) and (3) the case perspective (“What?”).

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. (Grigori et al., 2004), List and Machaczek (List and Machaczek, 2004) and Schiefer et al. (Schiefer et al., 2004).

6 CONCLUSION AND FUTURE WORK

FlowSpy is an environment for business process mining. Differently from Aalst (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 allows the accurate tracking of activity executions. 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 striking component of FlowSpy when compared 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 simpler aggregate tree. The latter facilitates pattern interpretation by producing on demand, more detailed or generic trees to represent the obtained patterns.

Currently FlowSpy implements the mining algorithm of Spiliopoulou (Spiliopoulou, 2000), provides a form-based interface for mining, allows the tree visualization abstraction described in Section 4, as well as the presentation of processes according to KPIs. We are implementing the interfaces for log pre-processing tools and to incorporate process metadata. The conclusion of this prototype will allow its validation in a real business process analysis case study.

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, FlowSpy provides the integrator component, whose function is to combine these issues in a synergic approach, and 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, according Fig 8. The status of our research is to study and develop a data storage structure (aggregate tree) by means of performance metrics. Such structure may then be used as an execution model to predict results and behaviors. 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. (2008) "Decision Support Based on Process Mining". Handbook on Decision Support Systems 1. International Handbooks on Information Systems. Springer-Verlag, Berlin, January 2008, chapter 29, pp. 637-657.

Aalst, W.M.P. van der; Reijers, H.A.; Weijters, A.J.M.M.; Dongen, B.F. van; Medeiros, A.K. Alves de; Song, M. and Verbeek, H.M.W. (2007) "Business Process Mining: An Industrial Application", Information Systems, July 2007, 32(5), pp.713-732.

Aalst, W.M.P. van der. (2007) "Trends in Business Process Analysis: From Verification to Process Mining". 9th International Conference on Enterprise Information Systems (ICEIS 2007), June 2007. **Proceedings…** Medeira, Portugal: Institute for Systems and Technologies of Information, Control and Communication, INSTICC, pp. 12-22.

Aalst, W.M.P. van der and Günther, C.W. (2007) "Finding Structure in Unstructured Processes: The Case for Process Mining". 7th International Conference on Applications of Concurrency to System Design (ACSD 2007), Bratislava, Slovak Republic, July 2007. **Proceedings…** Los Alamitos, California: IEEE Computer Society Press, pp. 3-12.

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

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, November 2003, 47(2), pp. 237-267.

Casati, F.; Castellanos, M.; Salazar, N.; Dayal, U. (2007) "Abstract Process Data Warehousing". 23rd International Conference on Data Engineering (ICDE 2007), Istanbul, Turkey, April 2007. **Proceedings…** IEEE Computer Society, pp. 1387-1389.

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

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

Golfarelli, M. (2005). "New Trends in Business Intelligence". 1st International Symposium on Business Intelligent Systems (BIS'05), 2005. **Proceedings…** Opatija, Croatia, pp. 15-26.

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, Washington, November 2004. **Proceedings...** New York: ACM Press, pp. 1-6.

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

Han, J.; Kamber, M. (2006) “Data mining: concepts and techniques” (Second Edition). San Francisco, CA: Morgan Kaufmann, March 2006, 770 p.

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

Rozinat, A. and Aalst, W.M.P. van der. (2008) “Conformance Checking of Processes Based on Monitoring Real Behavior”. Information Systems, March 2008. 33(1), pp. 64-95.

Rozinat, A. and Aalst, W.M.P. van der. (2006) “Decision Mining in ProM”. 4th International Conference on Business Process Management (BPM 2006), Vienna, Austria, September 2006. **Proceedings…** Berlin: Lecture Notes in Computer Science, 4102, pp. 420-425.

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), San Diego, July 2004. **Proceedings...** San Diego: IEEE Computer Society, pp. 162-169.

Song, M. and Aalst, W.M.P. van der. (2008) “Towards Comprehensive Support for Organizational Mining”. To appear in Decision Support Systems, 2008.

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

Tan, P.; Steinbach, M.; Kumar, V. (2005) “Introduction to data mining”. Boston : Addison-Wesley, May 2005, 769 p.

Tristão, C.; Ruiz, D. D. A.; Becker, K. (2008). “FlowSpy: exploring Activity-Execution Patterns from Business Processes”. IV Simpósio Brasileiro de Sistemas de Informação, Rio de Janeiro, abril 2008. **Proceedings...** Porto Alegre: SBC, 2008, 1, pp. 152-163.

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, June 2005, 525 p.