Improving Product Usage Monitoring and Analysis with Semantic Concepts

Mathias Funk[1], Anne Rozinat[2], Ana Karla Alves de Medeiros[2], Piet van der Putten[1], Henk Corporaal[1], and Wil van der Aalst[2]

[1] Dept. of Electrical Engineering, Eindhoven University of Technology, The Netherlands
{m.funk,p.h.a.v.d.putten,h.corporaal}@tue.nl
[2] Information Systems Group, Eindhoven University of Technology, The Netherlands
{a.rozinat,a.k.medeiros,w.m.p.v.d.aalst}@tue.nl

Abstract. Nowadays, complex electronic products, such as DVD players or mobile phones, offer a huge number of functions. As a consequence of the complexity of the devices, customers often have problems to use such products effectively. For example, it has been observed that an increasing number of technically sound products is returned due to, e.g., interaction problems. One possible root cause of this problem is that most product development processes are still too technology-driven, i.e., potential users are brought into contact with the product only at a very late stage. If early consumer tests are carried out, then these typically aim at abstract market evaluations rather than formulating concrete requirements towards the functionality of the product. As a result, products often have little meaning or relevance to the customers. Therefore, we need better ways to involve users in the development of such products. This can be achieved by observing product usage in the field and incorporating the gained knowledge in the product creation process. This paper proposes *an approach to build automatic observation modules into products, collect usage data, and analyze these data by means of process mining techniques exploiting a novel semantic link between observation and analysis*. This link yields two main benefits: (i) it adds focus to the potential mass of captured data items; and (ii) it reduces the need for extensive post-processing of the collected data. Together with the framework's flexibility to change observation modules remotely on-the-fly, these benefits speed up the information feedback cycle towards development.

Key words: product monitoring, log analysis, process mining, ontologies, semantic process mining

1 Introduction

Complex electronic products, both for private consumers and professional users, are hard to specify and design as no real information is available about the potential customers' expectations and needs. Meeting these expectations is, however, crucial as nowadays customers can choose among a wide variety of products, and will more easily reject products that do not suit their needs. A symptom of this problem is, for example, that an increasing number of technically sound products is being returned [1]. At the same time, it is not possible to perform lengthy user studies as there is a strong pressure on ‘time to market’. Moreover, it is difficult to gradually improve products by incorporating user feedback from the field as often only very few generations of the same product are made (to be replaced by new, more innovative products). In short, customers are becoming more demanding, whereas product development must be done with fewer iterations.

One way to ensure that products will suit the needs of potential customers is to involve these people as early as possible in the development process. This can be achieved by letting potential users test early prototypes, and to incrementally incorporate the gained knowledge into the product under development. However, to make this approach applicable in practice, two conditions need to be fulfilled.

1. It needs to be *feasible* to perform the tests in the first place, i.e., it should fit into today’s challenging development cycles.
2. The collected test data needs to be *useful*, i.e., valid (“Does this reflect our potential customers?”) and relevant (“Is this what we want to know?”).

To address the first condition, the test data needs to be collected and fed back to the development team as fast and automatically as possible. As we will demonstrate later, *our approach is supported by a tool chain that allows for seamless data collection, processing and analysis with a high degree of automation*. Addressing the second condition is more difficult as data quality depends on a variety of parameters. For example, to obtain valid data one needs to choose test users that reflect the actual target group. However, one common problem is that early user tests are often performed in non-representative environments, and that people do not behave normally as they feel observed. *Our approach allows for data collection from testers using the product in their habitual environment*. For example, test products are given to users who unpack, install and use the devices at home. The products themselves record usage information and automatically deliver it to the respective development unit in the company. This way, tests can easily run several weeks, and thus cover different phases of use [2]. Research has shown that the long-term usage behavior is often quite different from the behavior during the first few hours after unpacking the product. Finally, to ensure that the right data is collected, *we allow the observation logic to be changed dynamically by the development team, i.e., while the test is running*. This way, truly iterative data collection and analysis becomes possible. Furthermore, a visual approach to specifying the observation logic is taken to make it accessible to the (mostly non-technical) people that have an interest in the data collection process. These are, for example, product managers, quality engineers, interaction designers, or user interface developers.

With the aim to further increase both the feasibility and the usefulness of product usage observation, we extend the above-described approach by an important aspect: in this paper, *we establish a semantic link between the observation and analysis phase*. More precisely, we allow to semantically annotate the logged data during the specification of the observation logic, and these semantic annotations are preserved and actively leveraged in the analysis phase (by *semantic process mining techniques*). So-called *ontologies* [3], which are representations of a set of concepts within a domain and the relationships between those concepts, are used to define these semantic aspects. To allow different views on the data, multiple ontologies can be used to “tag” the observed data with orthogonal concepts at the same time. As a result, the logged data is pre-processed and structured using high-level concepts; consequently, there is no need for extensive and time-consuming post-processing of raw data. Instead, the data can be analyzed directly and in a more efficient way.

In the remainder of this paper, we first point at related work (Section 2). Then, we introduce an example scenario based on a case study that is currently performed (Section 3). Afterwards, we describe our semantic monitoring and analysis approach in more detail (Section 4), and present an implementation (Section 5). Finally, the paper is concluded.

2 Related work

Uses of remote product monitoring have been reported before [4, 5, 6, 7]. However, these approaches assume information stakeholders capable of programming and willing to use programming paradigms to achieve the sought-after data. In contrast, our approach aims at means to specify observation in a way that is doable by actual stakeholders of the collected information. Besides that, our approach towards product observation emphasizes the integration of observation functionality into the target system by using a software engineering process which is, in our opinion, necessary for widespread use. While previous work [8, 9] describes our product observation approach in more detail, this paper focuses on the novel semantic link between observation and analysis.

The idea of using semantics to perform analysis of processes is not new [10, 11, 12]. Our analysis approach is based on previous work on semantic process mining techniques [13, 14]. Process mining techniques can provide valuable insights into a real-life process based on data registered in event logs and have been successfully applied in practice [15]. *Semantic* process mining enhances the analysis by leveraging semantic information [13]. However, previous works do not present any real-life application of the semantic process mining tools. In this paper, we first applied our semantic process mining techniques to analyze processes based on product usage. More related work can be found in our technical report [16].

3 Example Scenario

In the following we want to use a simple example scenario to explain our approach. This example is a simplified version of (but based on) an industrial case study that is currently being performed.

We consider a product that offers a video playback and recommendation function as depicted in Figure 1. In the upper part of the screen one can see the video that is currently played. The video playback can be paused and resumed, and the playback window can be maximized to be displayed in fullscreen mode and brought back to the normal mode. In the lower part of the screen a number of recommendations related to the current video are displayed (using the right or the left arrow more related recommendations can be explored). Any of these recommendations can be viewed in more

Fig. 1. Schematic view on the user interface of a video playback and recommendation function of an industrial product in prototype stage

detail by moving the mouse pointer over it (as can be seen for the right-most recommendation) and selected for playback, after which it is displayed in the upper part of the screen. New recommendations are then retrieved and displayed according to the selected item. Furthermore, the product has a search function that allows to search for video content by name and categories, which is not shown in Figure 1.

We assume that a prototype of this product should be tested by potential end users in a number of different countries. We want to know how people typically navigate this user interface, and whether this differs depending on the cultural context. For example, it would be interesting to know whether users tend to follow the provided recommendations or rather search for video content on a case-by-case basis. Based on this information, the user interface of the product could be improved to best support the most common interaction flows.

4 Approach

Direct product information (i.e. the recording of the actual usage of a system) is potentially of use to a large group of professionals involved in the product development process: knowledge engineers, product managers, requirements engineers, developers, interaction designers, and other information stakeholders can benefit from such information. Note that the members of this group, in the following referred to as *domain experts*, have traditionally only a rather modest influence during some phases of the product creation process. Especially for the development of innovative products, the expertise of such domain experts is needed. These experts are the target users for our approach: initially, they might have a vague understanding about what should be observed in the product to answer open questions, but iteratively it is possible to map issues to observable items within the product, and finally, to obtain comprehensible and reliable information.

In the remainder of this section, we first provide an overview about our product usage monitoring approach (Section 4.1) and then elaborate on the role of ontologies as a semantic link between the different phases of observation and analysis (Section 4.2).

4.1 Overview

Consider Figure 2, which depicts an overview of our approach. The system we propose is a *combination of a logging framework and a process mining tool*. On top of that, one or more ontologies are used to link collected data items, hence, to connect observation and analysis on the information level. The figure shows that ontologies are connected to all three steps of the flow. Therefore, the definition and maintenance of one or more ontologies should be a concurrent task that accompanies the depicted flow.

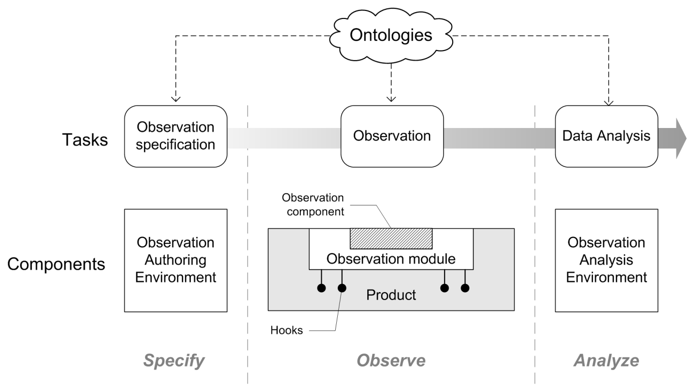


Fig. 2. Overview of our approach towards product usage monitoring and analysis

In Figure 2 one can see that the product to be observed is equipped with an *observation module* which has access to so-called *hooks*. These hooks and the observation module have to be built into the product beforehand. For example, in the scenario described in Section 3, before actually giving the prototypes to testers at home, the user interface would be instrumented with hooks that are triggered as soon as a video playback is started, a recommendation is selected etc.

During the actual test the following three steps are performed in an iterative manner: (1) The first step of the actual flow is the observation specification: domain experts visually define what information should be observed in the product and how this information relates to the concepts from the ontology. This task is done within an easy, but formal visual language. (2) The outcome are observation specifications which are used to automatically and remotely instruct the observation modules in the various products by simply replacing their *observation component*. The observation modules collect field data during product usage depending on their current configuration and send it to a central data storage. The semantic annotations of the observation specifications enable the observation module to categorize the captured data accordingly on-the-fly. This results in log data with an inherent semantic structure. (3) In the third step (data analysis) the data is processed using various (semantic) process mining techniques which provide different views on the aggregated data. This last step offers the possibility to extract the essence out of a potentially huge data set. Furthermore, it helps to present this information in a comprehensive and directly usable way.

Although the automatic processing chain from observation to analysis consists of several independent parts, it now becomes clear that a common connection is feasible by using ontologies for a semantic content structure. The whole process is of a strongly iterative nature. Cycles between the definition of ontology, observation specification, observation, and analysis are not only expected but encouraged to finally achieve the most reliable and accurate picture of product usage. For instance, during the observation phase, the domain expert might come across unexpected information that needs a special treatment and the extension of the connected ontology with new concepts. These changes can be carried out directly and lead to an immediate improvement of the quality of collected data.

4.2 Ontologies

Ontologies [3] define the set of shared concepts necessary for the analysis, and formalize their relationships and properties. Ontology elements are organized in a directed graph and there are several formalisms to build ontologies such as OWL [17] and WSML [18]. An example fragment of an ontology is depicted on the right in Figure 3.

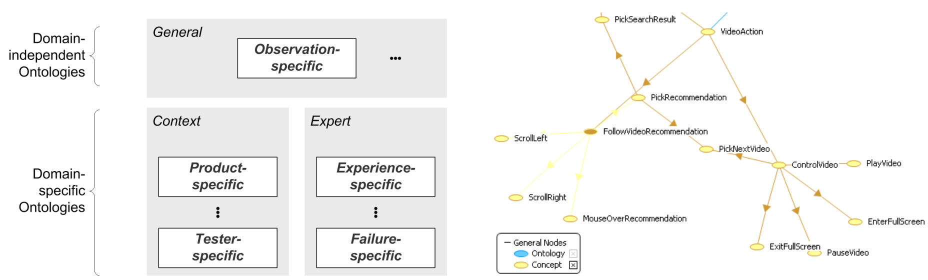


Fig. 3. Types of ontologies relevant for product usage monitoring (left) and an example fragment of a product-specific ontology representing user actions (right)

In the context of our product usage monitoring approach, the ontologies provide the link between conceptual level and information level, i.e., ontology concepts appear in the log data whenever a semantically annotated event is logged. We identify three types of ontologies: *general*, *context* and *expert* (cf. left side in Figure 3). These types can be characterized as follows.

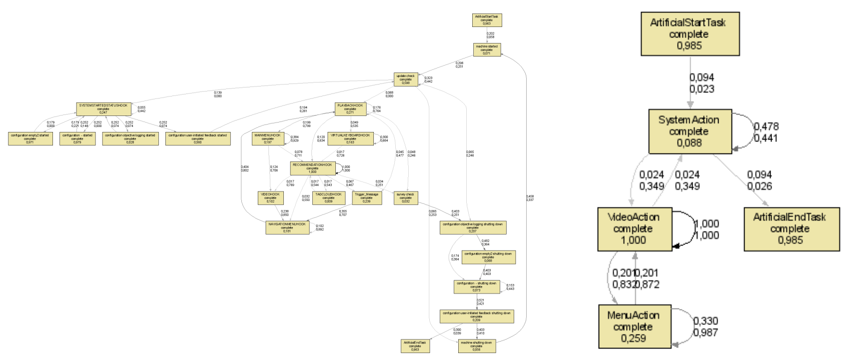
General General ontologies are domain-independent and they are used to capture concepts that are neither product nor experiment related. They are expected to be highly re-usable for a couple of experiments without changes.

Context Context ontologies provide information about the setting of an experiment. In other words, they might characterize certain aspects of the product to be observed (i.e., *product-specific*), the habitual context of actual product use, or the people that perform the tests (i.e., *tester-specific*). The applicability of these ontologies may be limited to a certain domain or product group, but they can be re-used across different experiments within that scope.

Expert Expert ontologies are related to specific analysis purposes. For example, we can think of certain domain-expert views, such as a user experience expert seeking emotional feedback from testers by popping up dialogues on the tested prototypes (i.e., *experience-specific*), or the quality engineer focusing on product failures (i.e., *failure-specific*). In principle, expert ontologies could be re-used across different product groups.

Note that multiple ontologies are used because the semantic observation and analysis is not done by one person alone. A team of domain experts should be able to work together, and to benefit from each other's insight into product usage. Therefore, many (potentially orthogonal) views on the topic have to be combined in an efficient way.

Nevertheless, in the remainder of this paper we focus on user actions only. On the right side in Figure 3, an excerpt of a product-specific ontology representing user actions for our example scenario in Section 3 is shown. One can see that concepts are organized in a hierarchical way, i.e., concepts may have one or more superconcepts. For example, the concept 'PlayVideo' is a subconcept of the 'ControlVideo' category, which in turn is a subconcept of 'VideoActions'. These subsumption relationships are a very useful tool as they enable the analysis of the data on different levels of abstraction.



(a) Model made using all events in the log and not using any semantic information

(b) Highly abstracted view on the same proces

Fig. 4. Two models that were mined from the same log data, but using different abstraction levels

This is illustrated by Figure 4, where process mining was used to automatically create a process model from the data collected in the example scenario. In the model depicted in Figure 4(a) the raw data and no semantic annotations were used to create the model. In fact, this model not only contains steps related to user actions but also includes unrelated information such as status checks of the observation system itself (since these are logged as well). In contrast, the model in Figure 4(b) only contains process steps relating to user actions. Furthermore, the depicted model provides a highly abstract view by making use of the semantic information in the log data. For example, since both 'PlayVideo' and 'PauseVideo' are a 'VideoAction' according to our ontology, they are not differentiated in this model. Note that although the model depicted in Figure 4(b) may seem too general, the level of abstraction can be varied at wish and without the need to modify the actual data itself. This way, varying models with even heterogeneous degrees of abstraction can be created easily. For example, we can create a model that provides a detailed view on 'VideoActions' but fully abstracts from 'MenuActions'.

5 Implementation

We have fully implemented the approach outlined above and are currently testing it in an industrial case study. In the following two sub sections, we describe the tools that we used for the realization (D'PUIS and ProM), focussing on newly added functionality and the semantic aspects.

5.1 Observation Specification and Data Collection via D'PUIS

We have developed the D'PUIS (Dynamic Product Usage Information System) [8, 9] as a platform-specific realization of the specification and observation approach depicted in Figure 2. This system consists of the following parts: (i) a visual editor to create observation specifications, (ii) a web application that distributes observation specifications as observation components and provides storage for collected product data, and (iii) an observation module which is integrated into product instances. An infrastructure connects these parts and enables an automatic flow from observation specification to actual product usage data.

In the context of semantically supported data collection, an interesting part of the observation system is the visual language as the the place where semantic links between data items are initially constructed. To do this, the visual language was extended to automatically incorporate each concept from a linked ontology as an available *Semantic Hook*. If such a semantic hook is triggered, a semantically annotated log entry is created. Often, the actual platform hooks can be connected to semantic hooks in a straightforward way, merely differentiating between a number of options. However, the processing nodes of our visual language also allow for more powerful observation specifications, which is demonstrated by the following example.

Consider Figure 5, which depicts a part of the observation specification for our example scenario in the visual editor. The lightbrown block in the middle represents the 'RecommendationHook' that is triggered whenever a recommended video is selected for playback by the user (cf. Section 3). However, in fact the same user interface component (and, thus, the same platform hook) is triggered when a user picks a search result after explicitly searching for video content. But in our scenario we want to differentiate between these two conceptual actions. Fortunately, we can create a context-aware observation specification that only triggers the semantic hook 'PickNextVideo' (i.e., the actual recommendation) when the user did not just enter the search mode via checking for the context node 'EnterSearch', which is also based on semantic information. If the search mode was entered before, the semantic hook 'PickSearchResult' is triggered instead. Note that this kind of domain-dependent reasoning would normally need to be made later in the analysis stage, or hard-coded into the product.

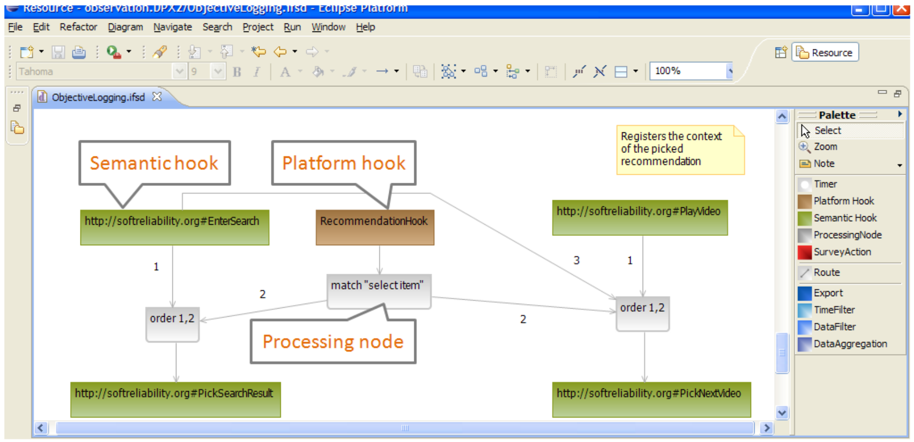


Fig. 5. Visual editor for observation specification with an example specification from the example scenario

Data that is acquired in the described way is not only more meaningful, but also it is *self-contained*. This is an important step forward as all the (usually implicit) information about the observation process, such as the characteristics of the observation environment, and the nature of data sources, is *explicitly* stated in a *machine-readable form*. In the analysis phase, specialized semantic process mining techniques can then exploit such information efficiently.

5.2 Semantic Process Mining using ProM

To be able to analyze the log data with our process mining tool kit ProM [19], we have developed a ProM*import* [20] plug-in that automatically extracts the recorded data from the D'PUIS database and converts them to the SA-MXML (*Semantically Annotated* Mining XML) format [14][1]. Note that this data conversion preserves the semantic annotations collected during the observation phase for analysis. Process mining techniques support various types of analysis based on the behavior registered during the execution of some process [15]. Semantic process mining uses semantic information to lift the analysis provided by current process mining techniques to the conceptual level [13, 14]. Seven semantic process mining plug-ins have been added to the ProM tool so far; we briefly introduce the following two: Performance Metrics in Ontologies and the Ontology Abstraction Filter.

The *Performance Metrics in Ontologies* plug-in provides feedback about (i) the processing times of tasks (or events) and (ii) throughput times of process executions. In our approach, the feedback in (i) is particularly important because it indicates how much time users typically spend in using certain functionalities of products. Moreover, this plug-in also shows how frequently instances of a given concept have been performed. Figure 6(a) contains a screenshot of this plug-in in action. Note that the coloring of the

[1] Both ProM and ProM*import* are open-source and freely available at *www.processmining.org*.

concepts in the ontology is based on the frequency of instances. From this graph, it is very intuitive to spot that the users in our example scenario were more often navigating between recommendations (concept 'FollowVideoRecommendation') than actually playing videos (concept 'ControlVideo').

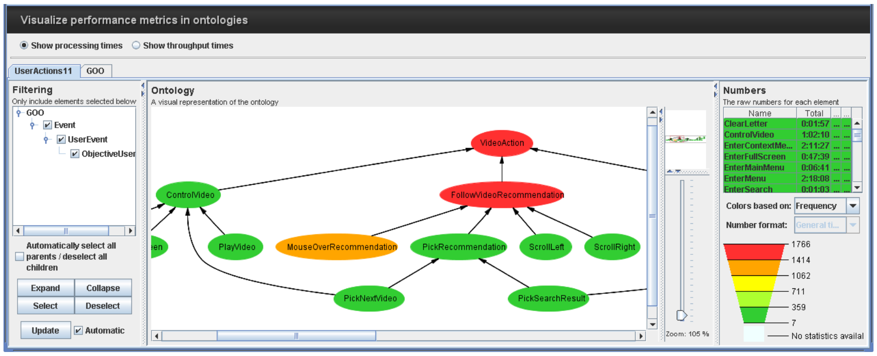
The *Ontology Abstraction Filter* plug-in supports ontology-based run time filtering of the data in a way that is accessible to all existing process mining algorithms in ProM (also if they are unaware of the semantic annotations in the log). In this filter, the desired level of abstraction is determined by selecting or deselecting concepts linked to events (the actual instances of these concepts) in logs. Afterwards, process mining algorithms can be used to create models on the current level of abstraction. For example, Figure 6(b) depicts a screenshot of the Fuzzy Miner [21] showing a detailed process model of the user actions. One can see that after searching for a video ('EnterSearch' followed by 'PickSearchResult' and 'PlayVideo') users tend to follow recommendations ('PickNextVideo') rather than going back to explicitly search for further videos.

6 Conclusion

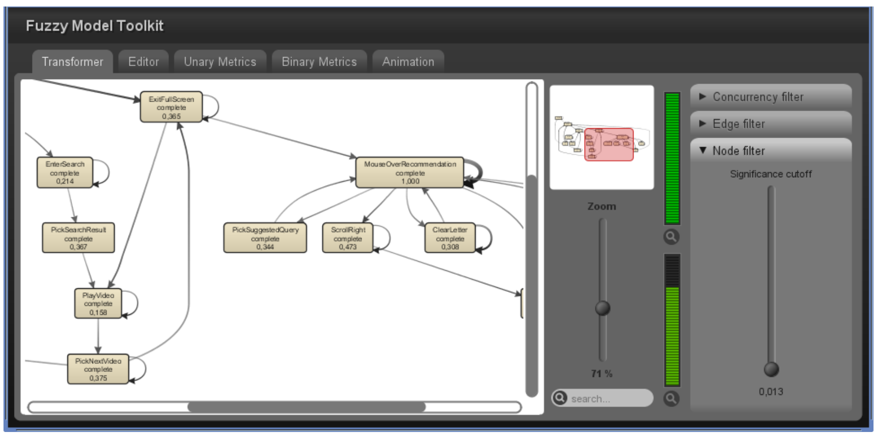
In this paper, we presented a novel approach to semantically link the observation and analysis of product usage data by conceptual information captured in ontologies. This link renders a potential mass of captured data items more manageable, and reduces the need for extensive post-processing. The presented approach ensures high information quality and speeds up the information feedback cycle towards development. Furthermore, we presented a tool chain that supports our approach throughout the phases of observation specification, data collection, and analysis. This chain of connected data processing components offers also the flexibility to change observation remotely on-the-fly, enabling fast data collection and analysis iterations.

Our vision is a fully automated data collection, processing, analysis and presentation chain which is specified by only a few (potentially re-usable) documents. Ontologies and visual languages seem to be good candidates for such specification documents as they are accessible to the actual stakeholders of the observed usage data (e.g., the various domain experts). By putting these people in the position of being able to *specify what they want to observe*, one of the main problems in log analysis, namely data quality, can be addressed. In many real-life scenarios, the data are often still of a poor quality; because of a low priority in implementing logging facilities, and a lack of anticipation of the kind of analysis that should be eventually performed, collected data are not good enough to answer all the questions of interest. However, due to the immense opportunities and increasing feasibility (resulting from novel automated approaches as presented in this paper) it can be expected, that the integration of observation functionality will have a more prominent role in future product developments. As a consequence, better analysis results can be expected.

Acknowledgements. This work is being sponsored by the Dutch Ministry of Economic Affairs under the IOP-IPCR program. Some of the authors are also supported by the European project SUPER. Furthermore, the authors would like to thank the industrial team for the possibility of applying our approach in a real product development context.



(a) Screenshot of the *Performance Metrics in Ontologies* semantic ProM plug-in. The current view shows the frequencies of tasks linking to concepts



(b) Screenshot of the *Fuzzy Miner* plug-in in ProM. Before, the semantic information in the log has been used to filter only events referring to user actions

Fig. 6. The converted log can be loaded and analyzed using the ProM tool

References

1. Brombacher, A., Sander, P., Sonnemans, P., Rouvroye, J.: Managing product reliability in business processes 'under pressure'. Reliability Engineering & System Safety **88** (2005) 137–146
2. den Bouwmeester, K., Bosma, E.: Phases of use: a means to identify factors that influence product utilization. In: CHI '06: CHI '06 extended abstracts on Human factors in computing systems, New York, NY, USA, ACM Press (2006) 117–122
3. Gruber, T.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition **5**(2) (1993) 199–220
4. Hartson, H., Castillo, J.: Remote evaluation for post-deployment usability improvement. Proceedings of the working conference on Advanced visual interfaces (1998) 22–29
5. Hilbert, D.M., Redmiles, D.F.: An approach to large-scale collection of application usage data over the internet. icse **00** (1998) 136

6. Kabitzsch, K., Vasyutynskyy, V.: Architecture and data model for monitoring of distributed automation systems. In: 1st IFAC Symposium on Telematics Applications In Automation and Robotics, Helsinki (2004)
7. Shifroni, E., Shanon, B.: Interactive user modeling: An integrative explicit-implicit approach. User Modeling and User-Adapted Interaction **2**(4) (December 1992) 331–365
8. Funk, M., van der Putten, P.H.A., Corporaal, H.: Specification for user modeling with self-observing systems. In: Proceedings of the First International Conference on Advances in Computer-Human Interaction, Saint Luce, Martinique, IARIA, IEEE Computer Society (February 2008) 243–248
9. Funk, M., van der Putten, P.H.A., Corporaal, H.: UML profile for modeling product observation. In: Proceedings of the Forum on Specification and Design Languages (FDL'08), Stuttgart, Germany, ECSI, IEEE Computer Society (September 2008) 185–190
10. Casati, F., Shan, M.: Semantic Analysis of Business Process Executions. In: 8th International Conference on Extending Database Technology (EDBT '02), London, UK, Springer-Verlag (2002) 287–296
11. Hepp, M., Leymann, F., Domingue, J., Wahler, A., Fensel, D.: Semantic Business Process Management: a Vision Towards Using Semantic Web services for Business Process Management. In: IEEE International Conference on e-Business Engineering (ICEBE 2005). (2005) 535 – 540
12. O'Riain, S., Spyns, P.: Enhancing the Business Analysis Function with Semantics. In Meersman, R., Tari, Z., eds.: OTM Conferences (1). Volume 4275 of Lecture Notes in Computer Science., Springer (2006) 818–835
13. Alves de Medeiros, A., Pedrinaci, C., van der Aalst, W., Domingue, J., Song, M., Rozinat, A., Norton, B., Cabral, L.: An Outlook on Semantic Business Process Mining and Monitoring. In Meersman, R., Tari, Z., Herrero, P., eds.: OTM Workshops (2). Volume 4806 of Lecture Notes in Computer Science., Springer (2007) 1244–1255
14. Alves de Medeiros, A.K., van der Aalst, W.M.P., Pedrinaci, C.: Semantic Process Mining Tools: Core Building Blocks. In: Proceedings of the 16th European Conference on Information Systems (ECIS). (2008)
15. van der Aalst, W., Reijers, H., Weijters, A., van Dongen, B., Alves de Medeiros, A., Song, M., Verbeek, H.: Business Process Mining: An Industrial Application. Information Systems **32**(5) (2007) 713–732
16. Funk, M., Rozinat, A., Alves de Medeiros, A., van der Putten, P., Corporaal, H., van der Aalst, W.: Semantic concepts in product usage monitoring and analysis. Technical Report ESR-2008-10, Eindhoven University of Technology (2008)
17. W3C: Web Ontology Language (OWL). http://www.w3.org/2004/OWL/
18. de Bruijn, J., Lausen, H., Polleres, A., Fensel, D.: The web service modeling language wsml: An overview. In Sure, Y., Domingue, J., eds.: ESWC. Volume 4011 of Lecture Notes in Computer Science., Springer (2006) 590–604
19. van der Aalst, W.M.P., van Dongen, B.F., Günther, C.W., Mans, R.S., Alves de Medeiros, A.K., Rozinat, A., Rubin, V., Song, M., Verbeek, H.M.W., Weijters, A.J.M.M.: ProM 4.0: Comprehensive Support for Real Process Analysis. In Kleijn, J., Yakovlev, A., eds.: Application and Theory of Petri Nets and Other Models of Concurrency (ICATPN 2007). Volume 4546 of LNCS., Springer-Verlag, Berlin (2007) 484–494
20. Günther, C.W., van der Aalst, W.M.P.: A Generic Import Framework for Process Event Logs. In Eder, J., Dustdar, S., eds.: Business Process Management Workshops. Volume 4103. (2006) 81–92
21. Günther, C., Aalst, W.: Fuzzy Mining: Adaptive Process Simplification Based on Multi-perspective Metrics. In Alonso, G., Dadam, P., Rosemann, M., eds.: International Conference on Business Process Management (BPM 2007). Volume 4714. (2007) 328–343