Karlsruher Institut für Technologie

Karlsruhe Reports in Informatics 2012,1
Edited by Karlsruhe Institute of Technology,
Faculty of Informatics
ISSN 2190-4782

A Graphical Audit Facility for Data Processing and its Evaluation with Users

Jens Müller, Murat Kavak, and Klemens Böhm

2012

KIT – University of the State of Baden-Wuerttemberg and National Research Center of the Helmholtz Association

Fakultät für Informatik

A Graphical Audit Facility for Data Processing and its Evaluation with Users

Jens Müller, Murat Kavak, and Klemens Böhm

Information Systems Group,
Institute for Program Structures and Data Organization,
Faculty of Informatics, Karlsruhe Institute of Technology (KIT), Germany
{jens.mueller,klemens.boehm}@kit.edu, murat.kavak@student.kit.edu

Abstract. Personally-identifiable information (PII) is increasingly processed in a distributed way. This makes it much harder for individuals to oversee how their PII is used. In the legal systems of many countries, processing of PII is subject to restrictions. In particular, companies have to inform an individual on how they use his PII, and which external parties they transfer it to. We hypothesize that naïve approaches like log messages or plain text are not sufficient to this end. We in turn have developed a user-friendly auditing facility based on business processes (BPs). It visualizes data processing in real time, using the graphical process models one would deploy on a BP engine for execution. We also propose an approach to let a BP-management system generate the necessary audit events at runtime. An evaluation of realistic scenarios with users shows that our tool helps them to understand how their PII is used.

1 Introduction

Today, companies outsource parts of their processes to other companies that can perform them more efficiently. As an example, think of a loan-approval process: Consumer loans are highly standardized products with low profit margins. Banks and other organizations granting loans share information about risk factors and credit defaults through specialized credit bureaus like SCHUFA in Germany. As the first step of the loan approval, the creditor queries risk information from such an agency. Then it calculates the interest rate. This calculation can be outsourced as well. If a credit default occurs, information is sent to the credit agency. This example shows that outsourcing leads to personally-identifiable information (PII) to be transferred to third parties. PII is protected by law in the European Union [5] and elsewhere. The law gives individuals (data subjects, i.e., the person that the data relates to) the right to request information on how their data is processed, and where it is transferred. It also requires the informed consent of individuals to any processing and transfer of PII that is not necessary for the service provided. In order to give this *informed* consent, the user must be able to assess how his or her data will be used.

Current mechanisms fulfill the law formally, but are not useful in reality. Companies usually require individuals to give consent by signing terms and conditions that are both very broad and detailed. Information on data processing that companies provide usually is in text form. With large amounts of text, it is hard for users to find the details they are interested in. When PII is processed in a distributed way, companies have to tell individuals which other companies they have transferred it to, or where they have acquired it. In principle, this allows users to ask the other companies for information on their data. In practice, this is too tedious and time-consuming for users, especially when they have to follow their data over multiple hops. We conclude that information must be structured in a way that is easily accessible to individuals.

Business-process management (BPM) supports the complete lifecycle of orchestrations, i. e., applications combining lower-level functionality, from models to executable processes. Non-functional requirements, including security requirements, can be expressed as annotations to graphical process models [9]. Using an aspect-oriented approach, they can be translated into a process running in a BPM system (BPMS) with a standard business-process engine extended with security components.

The goal of this article is to study how to let users track business processes that use and transfer their data with ease. Because the right to information is not limited to finished cases, users must be able to get information about running processes as well. This is known as *auditing*. A BPMS should automatically generate the necessary events at run-time. To design auditing tools, we also need to understand how real users work with them. All these tasks are challenging, for various reasons at different levels:

- We need a succinct representation of audit information that is easy to understand at first glance. A respective system should also allow for drill-down in order to get more details. In particular, it should be easy to switch to another process following the data flow. Such a representation is not obvious.
- It is not possible to determine which auditing features are useful for users without a realistic scenario they themselves are part of.
- The functionality envisioned should re-use artifacts that are created anyway when modeling and deploying a business process. This minimizes the additional effort for application developers. This point concerns two issues in particular: First, audit information must be presented to the user. This step can re-use graphical business-process diagrams. However, additional information about their structure is needed. Second, the audit tool must acquire the necessary information at runtime. This requires an interface between the audit tool and the security components of the BPMS. However, the implementation is not obvious, as we will explain.

To this end, we have designed and implemented an auditing tool dubbed WoSec (*Wo*rkflow *Sec*urity) and have evaluated it with real users. More specifically, our contributions are as follows:

- We have analyzed which information must be provided to users in order to audit data transfer in distributed applications, and how it can be presented visually.

- We have developed WoSec, a web-based tool for auditing the handling of PII. It works with graphical representations of BPMN models of applications. It allows users to "follow their data" when it is transferred to another organization that also provides data to WoSec. It can also be used to visualize how an organization *intends* to handle PII, enabling users to give more informed consent.
- We propose an extension of a BPMS with extensions for managing security configuration and enforcing security decisions (secure BPMS) to provide events to the audit tool. It is not clear how to best do this. We have compared possible alternatives and explain our choice. Eventually, we propose to instrument business-process (BP) definitions before they are deployed to a BP engine.
- We have designed several example use cases for distributed data processing that are sufficiently complex for a realistic evaluation: applying for an internship, trading items at an online marketplace, and buying a car.
- We have designed a user study for evaluating our tool and various features of it. Having carried out the study, it shows that users prefer graphical audit facilities, and that these lead to a better understanding of data transfers. We have discovered that usability is curbed severely when process diagrams do not fit the screen without scrolling. This finding as well as other ones give way to an improved version of our audit facility.

2 Fundamentals

Because our auditing approach is based on BP models, we introduce some fundamentals regarding (secure) BPM in this section. BPM is an ideal solution for distributed information processing using loosely coupled systems. It is used to orchestrate (i.e., coordinate) the behavior of different components and actors. The starting point is a BP diagram, which is typically graphical, where a domain expert explicitly models control and data flow. In service-oriented architectures, BPM implementations are based on web-service technology such as SOAP, WSDL and WS-BPEL. Via SOAP calls, BP instances can communicate with automated services (i. e., web services). No standard has yet evolved for facilitating the participation of humans in BPs, although respective proposals exist [3,4].

The Workflow Management Coalition has proposed a reference architecture of workflow-management systems (WfMS), the workflow reference model [7]. It identifies five interfaces of a WfMS. This model still fits well the present use of WfMS (now usually called BPMS) in service-oriented architectures [6]. The purpose of Interface 1 is the deployment of process definitions into the BPMS. Interface 2 handles the participation of humans, usually via a worklist handler. Interface 3 concerns the interaction with application, which are available as web services. Interface 4 handles communication with other BPs. Because BPs provide web-service interfaces, the functionality of Interfaces 3 and 4 is quite similar. Finally, Interface 5 concerns *Audit and Monitoring*. The WfMC has specified a format for audit messages [13]. However, this format is focused on states of process and activity instances and does not address the handling of data items and the relationship between different processes through message exchanges.

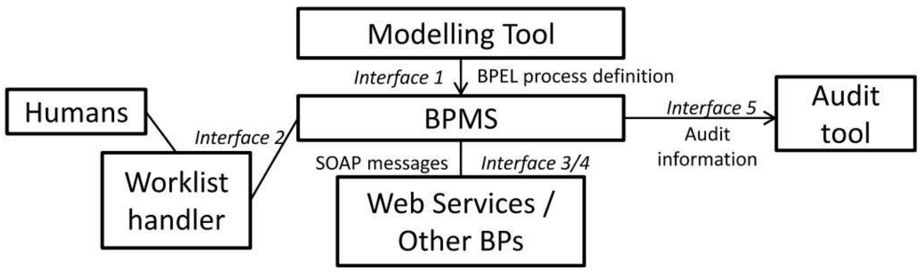


Fig. 1: Interfaces of the WfMC workflow reference model

When a running BP instance communicates with the environment using Interfaces 3–5, the BPMS has to make and enforce security decisions, e.g., which user is allowed to perform a certain activity. The state of BP instances and context information such as which entities are associated with them can influence security decisions. For example, choosing which third parties data is sent to can be up to the user performing a certain task in the process. A specification that is the basis for such decisions must be expressed in a form based on BP state and context [8]. There exists a proposal for an architecture of a business-process management system (BPMS) that takes context into account and allows respective configuration [10]. This architecture extends a conventional BPEL engine with security components. These security components capture and store the context of BP instances. This context, such as who has performed activities, is important for audit in the general case. The components also enforce security decisions regarding interactions of the BP with the outside world. BPs are instrumented using an AOP-style approach in order to run in that extended BPMS.

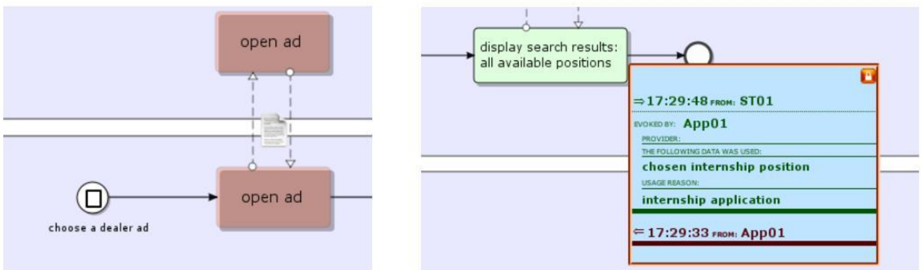
3 Functionality

In the following, we present requirements on the functionality of our audit tool from the user perspective. We have derived them by inspecting systematically which kinds of data that users are interested in arise in real-world processes. This description serves as a basis for our proposal of an architecture able to provide this functionality.

In more detail, we take a user-centric approach: We think of business processes handling data of one individual. This individual is allowed to track how his data is processed. To this end, he can access a list of BP instances handling his data and a detailed audit view for each of them. This view contains historical and live information, as we will explain below.

General structure: Because BPMN is a generally accepted standard, and BPMN diagrams already exist for applications modelled as business processes, we decide to use them as the basis for visualizing audit information. Note that such diagrams are static and do not contain information on the current state of process instances. They contain lanes (horizontal) representing the process (coordinating the overall application behavior), roles involved in it, and external parties, activities (rectangles), solid arrows (mostly horizontal) representing the control flow, and arrows (mostly vertical, dashed) representing the data flow between the process and persons/external parties.

Execution progress: WoSec visualizes execution progress by highlighting activities already executed and currently executing in this diagram, using a different color. When an activity starts execution, the tool automatically scrolls the viewport to that activity and notifies the user acoustically. Detailed information about activities is available, such as the time when it was performed and the user who has performed it.



(a) Animation of data flow

(b) Information box with details about an activity

Fig. 2: Screenshots of WoSec

Data transfer: A data object moving between activities indicates Data transfer (Fig. 2a). An information box provides details about data transferred (Fig. 2b). It is shown when pointing at activities and contains a colored mark indicating whether the activity sends or retrieves data, the external party involved, and the data itself, possibly with attachments displayed as links. The audit view allows access to historical information as follows: A timeslider allows to move to some point in the past and cause a playback of events from there. In case of loops, activities can be executed more than once. Information boxes for activities also contain information about earlier executions of an activity, not only the most recent one.

Multiple diagrams: When multiple organizations are involved in a process, they model their respective part independently, so there is no overall diagram for the whole process. Instead, several diagrams show the perspective of each participating organization, where the processes of other organizations are only represented by their interfaces called by the current process. This leads to smaller diagrams that are easier to oversee. It must be easy for users to discover what happens with their data in the other process. WoSec accomplishes this as follows: When an activity transfers data to another process, the user can jump to the other process instance from there. WoSec then opens it in another view and scrolls to the activity which receives the data.

Color scheme: As we add information to the basic BPMN diagrams by changing the color of elements, we need a color scheme in line with our intentions. We start out with the diagrams that use the color scheme of Intalio BPMN Designer [1], i. e., white tasks on a very light blue pool background. Our goal is that active tasks stand out, finished tasks retreat into the background, and information boxes are prominently visible even compared to active tasks. This leads to active tasks in red, finished tasks in light green, and information boxes in a saturated light blue.

4 Architecture and Design

We now describe the architecture of our application that yields the functionality presented in Section 3. It has to address three main issues: First, the BPMS has to provide information on the execution of BP instances to the auditing tool. Second, BP diagrams that have been created in order to be translated into executable processes must be provided to WoSec in a suitable form, i. e., a mapping from activities to graphical elements must be created. Third, we need to develop the internal architecture of WoSec.

Providing audit information: We see two alternatives for providing audit information: (1) We could rely on the audit events generated by the BPEL engine. However, [13] is insufficient for our purposes. But without a widely supported standard format, this would require separate implementations for every BPEL engine to be supported (either by adding event generation to it, or by handling the format used by that engine). In addition, formats used by existing BPEL engines do not include the necessary information either. (2) The BP definitions written can be instrumented. This means that processes are modified so that they generate audit events. A similar approach is pursued in [14]. It needs to be generalized in order to include all information necessary. – Because of the wider applicability and the likely easier implementation, we advocate (2).

Creation of graphical process models: Domain experts and process modelers initially create business processes as BPMN diagrams in a graphical modeling tool. These models are translated into executable BPEL processes. It is possible to instrument them automatically to provide audit information. Then the models are deployed in a BPEL engine. From the same BPMN diagrams, we need to create graphical representations for our auditing tool. It must be easy to highlight elements in them, and it must be clear which elements belong to which process activities. Intalio BPMN Designer produces a SVG version of process diagrams where elements are annotated with activity names. We create a description file for the mapping between activities and graphical elements. The SVG graphic and the description file are then deployed to WoSec. Fig. 3 shows how the auditing tool fits in the overall BPMS architecture.

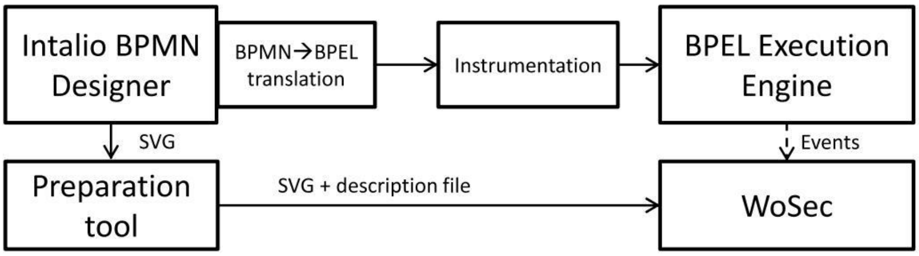


Fig. 3: Integration of the auditing tool into the overall BPMS architecture

Event type	Meaning
EventCommand	to initialise a process instance
StartingTask	to mark a task as active
FinishingTask	to mark a task as finished
SpecifyingParticipant	to specify the communication partner
TransferingData	for data transfers between two active tasks

Table 1: List of event types the server sends to the client

Internal Architecture of WoSec: WoSec itself displays audit information by overlaying it on the SVG diagrams. We have taken the following design decisions: (1) We have chosen a client/server design because we need a server part that stores BP definitions and is able to receive audit events at any time. In general, an application running at the computer of the user cannot achieve this. (2) The client can either be a stand-alone application or a web application running in a browser. This also brings up the question of where to handle the graphical representation. Because a web application does not require installing any software and is thus easier to use, and modern browsers natively support SVG, we pursue this option. (3) A question in any client/server application is the granularity of information transferred between client and server. In our solution, all event information is cached on the client side and thus sent only once in a login session to minimize response times. To be able to display the events, the following information is provided for each event: an event type (see Table 1), a timestamp, the id of the SVG element the element applies to, a list of all entities involved, attachments sent, a list of data used and the purpose of use.

5 Study Design

The goal of our study is to find out whether our visualization improves the understanding of users of how their PII is processed. In addition, we want to discover shortcomings of WoSec, possible misconceptions of how audit information should be presented to users, and consequently opportunities for improving WoSec in particular and audit facilities for end users in general.

To achieve these goals, we need an appropriate scenario and a realistic baseline for comparison. We also need to measure the understanding of the participants. This requires asking participants for their assessment according to criteria we deem important. In addition, we need to check these answers for plausibility. We do this by asking questions about what has happened in the scenarios and analyzing the behavior of the participants during the study. We use textual audit messages as the baseline, as organizations still use it to answer information requests and to fulfill their legal obligations.

The scenarios used in the study must be sufficiently complex. This means that several organizations should be involved in processing different kinds of data. The scenarios must match the interests of the study group. Although they are less detailed than in the real world, they must appear as natural as possible.

We provide a realistic web application and the corresponding audit view at the same time, using a split browser window. The upper half shows the mock-up of a web application, while the lower half contains an audit view, which either uses our visualization or is text-based.

We chose to evaluate an early prototype of WoSec in order to get preliminary feedback. Subsequently, we have improved it and then carried out a more extensive and sophisticated study. In both studies, we have compared our graphical auditing tool to a text-based audit view. Before going through the scenario, we asked the participants questions about their Internet usage and some demographic information. In each round, the participants worked with one scenario and one variant of the audit view. After each round, we asked the participants assessment questions. Additionally, we asked them control questions about the scenario. Finally, we asked them to compare the several rounds. The full questionnaires can be found in [2].

5.1 Pre-Study

The early prototype was tested with individuals who are IT users, but who do not have a heavy computer-science education. The participants were 16 high-school students. The scenario was a fictitious social network called “FaceVZ”. Such a scenario is well-known to the target group and, according to our expectations, triggers increased privacy awareness due to recent media reports about privacy threats in such networks. The scenario has included several steps: Images are uploaded to a social network, face recognition linking the pictures to accounts is performed, and finally free prints of the pictures can be ordered. We have used two versions of the scenario with a subtle difference: In one version the user address was submitted to the print service, in the other one the prints had to be picked up in a shop. All participants used both the graphical and the text-based audit. There have been two groups of participants with different order, to rule out learning effects.

This pre-study revealed some shortcomings in the implementation leading to visualization errors. A core result has been that the participants liked the graphical version better. When asking for a decision between versions on a 7-step Likert scale where 1 indicates a strong preference for the textual version and 7 for the graphical version, the participants preferred the graphical version regarding user-friendliness (mean: 5.69) and clarity (mean: 5.13). The group that tried out the graphical version first was able to give about 50 % more answers to the control questions. Unfortunately, the participants did not fully understand the purpose of such an audit system, as they were not able to give any examples of possible applications of such audit systems.

5.2 Main Study

For the main study, we used an improved version of the visualization with more features. We have tested the following hypotheses:

- H1: Our visualization helps users to understand which of their data is transmitted to which organizations.
- H2: The visual audit facility has good usability.
- H3: Users prefer a graphical audit facility over a text-based one.

Because the participants of our pre-study have not been able to really see the benefits of an audit facility, presumably because of their young age, we chose to perform the main study with more experienced participants. To evaluate the functionality of WoSec in full, we devised a more sophisticated study setup. We decided to use two different configurations of the visualization, in addition to the text-based audit view: While both configurations allow to access all audit information available, one contains additional features aiming at improved usability. This allows to test the influence of the non-essential features and to explore the opinion of the participants regarding the additional functions. To avoid learning and order effects when comparing the configurations, we used different scenarios for the different configurations. The participants went through the scenarios in the same order, but the order of configurations was randomized.

In both configurations, the visualization was immediately updated when something relevant had happened in the web application. Both contained the basic animations, i. e. activities starting execution blink, and data transfer is animated. In addition, the full-featured configuration automatically scrolls to activities becoming active. It also shows the actual data transferred, the user causing the activity, and the reason for transferring the data. For activities, a window shown on right-click contains a textual description of the activity and all users involved.

We recruited our study participants from our directory of individuals interested in user studies related to information systems, which includes mostly university students and adults with university education. We designed three scenarios specifically for this group:

(1) *Internship application:* In this scenario, eligible students get support for placement in an internship program, which relates to their course of study. First, participants enter their registration information. They then wait for several steps: A university coordinator has to approve their application, registration data is written to a database, the university coordinator chooses a placement service, which then sends a list of possible internships. Finally, the participant chooses an internship. (2) *Online trade:* In this scenario, participants have to sell an item. First, they have to enter their trader data and the item description. They then wait for another customer to open the ad, buy the item and send his or her contact data. The participant now gets the address of the buyer, has to prepare the parcel and hand it to a parcel service. (3) *Car purchase:* Here, the participants have to buy a car on credit. They have to wait for a list of available cars and choose one. They then have to enter their personal data and, in our example, state that they want to buy the car on credit. They choose a bank for the loan and accept the terms of the credit bureau. They then have to wait for several steps happening in the background: The bank receives the credit application and receives a score from the credit bureau. We assume that it grants the application. The car dealer receives a confirmation from the bank, reserves the car chosen and sends a purchase confirmation to the customer.

We paid the participants 10 EUR for their participation. To incentivize active participation, we promised an additional amount based on the level of participation. We computed this amount based on the number of questions answered. This means that participants who answered all 60 questions were paid another 5 EUR. In addition to the questions answered, we recorded for which scenarios, tasks, and participants information boxes for activities were shown. To get an overall impression how the user interface was used, we recorded a so-called heatmap, overlaying mouse clicks onto a screenshot of the user interface.

6 Results

In total, 17 individuals have participated in our study. The study group included participants of different age (20–74 years). All of them had some technical background, and all expressed some privacy concerns regarding Internet usage.

We performed the study in two separate meetings with participants, with 7 of them in the first and 10 in the second one. Due to technical problems, we were only able to test the full visualization at the first meeting. This means that 7 participants used the full visualization for all three scenarios. We weighted them with $1/3$ when computing the mean values to achieve the same weight per participant.

We compared the ratings given by the participants regarding the text-based and the graphical audit view, as well as the ones for the restricted and the full graphical version. We first tested all samples for normal distribution with the Shapiro/Wilk test [11]. Because this test did not confirm a normal distribution, we had to use the Wilcoxon/Mann/Whitney test to compare samples. The average answers and the result of the significance tests are shown in Table 2. For all tests, we required a level of significance $\alpha = 0.05$. We performed one-sided tests whether the underlying random variable of the sample with the larger mean actually is significantly larger. Table 3 contains the mean values of the questions asking for a direct comparison. The samples have not shown a significant difference from the neutral value 4.

	G	T	F	R	G/T^s	R/F^s
Q1 Have you been able to trace the flow of the data?^{S1}	5.71	4	5.31	6.1	>	=
Q2 Have you been able to trace why a data flow has happened?^{S1}	5.99	4.3	5.78	6.2	>	=
Q3 Could you predict following steps?^{S1}	5.43	4	5.26	5.6	>	=
Q4 How was the number of animations?^{S2}	4.8	3	4.6	5	>	=
Q5 Have you been able to orient yourself without problems?^{S1}	5.01	5.2	4.51	5.5	=	=
Q6 How clear was the auditing tool?^{S3}	4.61	4.1	3.92	5.3	=	>
Q7 How user-friendly was the auditing-tool?^{S4}	4.6	4.3	4.39	4.8	=	=
Q8 How much information content did the auditing tool contain?^{S5}	5.34	3.9	4.63	6.13	>	>
Q9 Do you feel adequately informed about all actions?^{S1}	4.93	3.9	4.75	5.11	>	=

Legend:
G : all graphical versions T : textual version F : full graphical version
R : restricted graphical version s : significance test
S1 : 1 = absolutely not, 7 = absolutely yes S2 : 1 = too few, 7 = too much
S3 : 1 = not clear at all, 7 = absolutely clear S4 : 1 = not user-friendly at all, 7 = absolutely user-friendly
S5 : 1 = very few information content, 7 = very much

Table 2: Assessment of different audit configurations

in Section 5, we have hypothesized that our visualization helps users to understand which of their data is transmitted to which organizations. (H1) The statistically significant difference for Q1, Q2, Q3, Q8, and Q9 shows that the visualization indeed led to a better understanding of data transfers. Moreover, the participants were slightly more satisfied with the amount of information available in the visualization (Q11), although the difference was not significant.

Regarding H2: *The visual audit has good usability*, we cannot show any statistical significance for Q5, Q6, Q7 and Q12. This indicates that the usability of all audit systems is equal. Nevertheless, the restricted visual audit scores significantly better than the full one. We conclude that automatic changes of the viewport decrease usability.

As one may expect, we could not show any statistically significant difference for H3, *Users prefer a graphical audit facility over a text-based one* through questions Q10, Q11, Q12 and Q13. Yet, except for a small outlier (Q10), the participants evaluated the visual audit slightly better than the textual audit.

In total, our tool improves the understanding of users, but there is potential for better usability. in particular, participants were annoyed by automatic scrolling. On the other hand, some participants have noted that the size available for the diagram was too small, and the heatmaps showed that the scrollbars were used a lot. We conclude that automated scrolling should be improved, or scrolling should be made unnecessary. We believe that splitting diagrams into parts and easy navigation between the parts can alleviate the respective problems. Participants mostly accessed the additional information provided by the information boxes when answering the questions. This indicates that they had used the audit facility mainly for that purpose. In a future study, participants should answer the control questions solely based on their usage of the system up to that point.

	Mean
Q10 How clear was the textual visualization compared to the graphical visualization?^{S1}	4.8
Q11 How much information content did the textual representation contain compared to the graphical visualization?^{S2}	3.5
Q12 How user-friendly was the textual representation compared to the graphical visualization?^{S3}	3.3
Q13 Which visualization do you prefer?^{S4}	4.5

Legend:
S1 : 1 = little clear, 7 = very clear S2 : 1 = very few, 7 = very much
S3 : 1 = not user-friendly at all, 7 = very user-friendly S4 : 1 = rather textual, 7 = rather graphical

Table 3: Direct comparison of text-based and graphical audit

7 Related Work

The WfMC has specified a audit data format for business processes [13] and started the development of an XML-based successor [15]. Both formats focus on the execution only, i. e., the state of BP instances and activities. They do not address data handling and the messages exchanged between BP instances. Weske [12] mentions a monitoring component that visualizes the status of business-process instances, but makes no statement how such a visualization should look like. The *BPMS Console* of [1] lists available BP definitions and instances and marks activities currently running. However, it does not address data transfer and is targetted at administrators, not end users. To the best of our knowledge, the effectiveness of generic audit facilities for data processing dedicated to end users has not been studied or empirically evaluated in the literature.

8 Conclusions and Future Work

We have created a graphical tool that allows end users to audit business processes involving data transfer and have described its integration into a BPMS. We have carried out a study comparing the tool with text-based audit facilities, which represent the current state of the art, and have assessed its impact on effectiveness and usability. Next to other points, the results show a usability problem related to limited viewport sizes and automatic scrolling. As future work, we plan to assess whether splitting diagrams into parts can alleviate these problems. We also plan to automate the BPMS integration.

Acknowledgment

This research has received funding from the Seventh Framework Programme of the European Union (FP7/2007-2013) under grant agreement n^o 216287 (TAS^3 - Trusted Architecture for Securely Shared Services)[1].

We thank Oleg Peters, Philipp Lingel, Justus Maier, David Rieger, and Thorsten Haberecht for their part in the design and implementation of WoSec. We also thank the participants of our study.

A shorter version of this report is scheduled for publication in the proceedings of the 14th Asia-Pacific Web Conference (APWeb), published by Springer-Verlag GmbH Berlin Heidelberg as Volume 7235 of the Lecture Notes in Computer Sciences (LNCS).

References

1. Intalio BPMS Designer, `http://www.intalio.com/bpms/designer`
2. WoSec website, `http://dbis.ipd.uni-karlsruhe.de/english/1746.php`
3. Active Endpoints, Adobe, BEA, IBM, Oracle, SAP AG: Web Services Human Task (WS-HumanTask), Version 1.0 (June 2007)
4. Active Endpoints, Adobe, BEA, IBM, Oracle, SAP AG: WS-BPEL Extension for People (June 2007)
5. European Community: Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data
6. Hollingsworth, D.: Workflow Handbook 2004, vol. 10, chap. The Workflow Reference Model 10 Years On
7. Hollingsworth, D.: The Workflow Reference Model. WfMC Specification TC00-1003, Workflow Management Coalition (1995)
8. J. Müller et al.: Secure Business Processes in Service-Oriented Architectures – A Requirements Analysis. In: ECOWS 2010
9. Mülle, J., von Stackelberg, S., Böhm, K.: Modelling and Transforming Security Constraints in Privacy-Aware Business Processes. In: Proc. SOCA 2011. To appear
10. Müller, J., Böhm, K.: The Architecture of a Secure Business-Process-Management System in Service-Oriented Environments. In: ECOWS 2011

[1] The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The above referenced consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law.

11. Shapiro, S.S., Wilk, M.B.: An analysis of variance test for normality (complete samples). Biometrika 3(52), 1–22 (1965)
12. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer (2007), http://dx.doi.org/10.1007/978-3-540-73522-9
13. Workflow Management Coalition: Audit Data Specification (1998)
14. Yao, J., Chen, S., Wang, C., Levy, D., Zic, J.: Accountability as a service for the cloud. In: Proceedings of the 2010 IEEE International Conference on Services Computing. pp. 81–88. SCC '10 (2010)
15. zur Muehlen, M. (ed.): Business Process Analytics Format (BPAF). WfMC Draft Standard WFMC-TC-1015 (February 2008)