Applying Design Principles for Enhancing Enterprise System Usability

Tamara Babaian, Jennifer Xu and Wendy Lucas
Department of Computer Information Systems, Bentley University, Waltham, MA 02452, U.S.A.

Keywords: ERP Usability, Human-Computer Interaction, Design Principles, Process Visualization, Tutorial Systems.

Abstract: Experienced users learn the "tricks of the trade" that allow them to make effective and efficient use of enterprise systems. New users and even experienced ones accessing unfamiliar parts of the system, however, are stymied by the lack of navigational cues, the ubiquity of tabbed pages, and the sheer number of fields that may be required. In earlier work, we documented the usability issues users are facing in the field, derived four design principles for guiding the design of systems that collaborate with their users, and implemented a database framework for supporting those principles. In this paper, we present two interventions targeted at the design principle that calls for providing navigational and progress guidance, while supporting the principle requiring that the system make use of contextual information in presenting data and choices to the user. Using visualizations in conjunction with a playback mechanism, users can learn about system-supported processes, find detailed information on the particular process instances on which they are working, and view automated playbacks of users completing the same and other types of tasks. This paper is the next step in designing enterprise systems that help, rather than hinder, their human partners by sharing their knowledge.

1 INTRODUCTION

How to get started, what to do next, what came before, and where does my work fit in to the wider business context are some of the questions users of Enterprise Resource Planning (ERP) systems ask themselves time and again. Novice and experienced users alike are often stymied by the lack of navigational cues, missing indicators of required versus optional steps, and the opacity surrounding how the system's imposed implementation of tasks relates to the underlying business processes.

These hurdles are much more than a distraction, with serious repercussions for the bottom line. Industry reports attribute the inability of ERP systems to achieve their full potential on their poor usability characteristics, which negatively impact adoption rates and end-user productivity (Matthews, 2008; Iansiti, 2007). While advances stemming from the increasing use of service-oriented architectures (SOAs) offer the potential for more improvements in usability, as it is easier for ERP vendors to apply changes, the coupling of diverse components may also lead to additional usability issues related to navigation and redundant functionality (Matthews, 2008). Similarly, more flexibility in ERP systems has been achieved via cloud-based applications that use a software as a service (SaaS) delivery model. Implementations using this approach, too, have yet to solve the inherent usability issues of ERP systems (Ganly and Montgomery, 2013), which remain a significant source of angst to users and their employers.

Field studies we undertook as part of a comprehensive project for improving the usability of ERP systems (Topi et al, 2006; Babaian et al., 2010) revealed several sources of confusion and frustration experienced by users across the board. Issues frequently cited in our own and the work of others include difficulties with navigation, overly complex interfaces, and hard-to-understand error messages (Singh and Wesson, 2009; Calisir and Calisir, 2004; Lambeck et al., 2014).

The issues we identified were first analyzed in light of the human-computer collaboration paradigm, which holds that the system must support its users in the increasingly complex environments of modern applications (Grosz, 2005). We then systematically categorized those issues on the basis of non-collaborative behaviors between the system

Babaian T., Xu J. and Lucas W..
Applying Design Principles for Enhancing Enterprise System Usability.
DOI: 10.5220/0005099001620169
In *Proceedings of the 9th International Conference on Software Engineering and Applications* (ICSOFT-EA-2014), pages 162-169
ISBN: 978-989-758-036-9

and its users. From these categorizations, we derived four principles for guiding the design of enterprise systems that act as users' collaborative partners (Babaian et al., 2010).

In this paper, we present two novel interactive user interface components specially designed to address one of our design principles, which concerns the need for providing navigational and progress guidance in light of the business and user contexts of each interaction. The first component presents the user with interactive visualizations of the current business process being performed. The second component allows users to view on-demand, automated demonstrations of how to execute the current task or any other task that has been previously performed with the system. The two user interface features are integrated within a single ERP prototype in a way that conforms to another design principle, which requires that the system uses its knowledge of the context of the current interaction in presenting choices and data to the user.

These two design interventions address many of the concerns we heard most frequently from users in our field studies. Providing access to process-based visualizations and automated playback of tasks improves the users' understanding of the flow of information between tasks, task composition into processes, the users' performing those tasks, and other details.

In order for a system to provide user access to such information, it needs to be able to store and retrieve data on the tasks it supports, task composition into processes, task interfaces, and the histories of system-user interactions. A database framework consisting of a Task-Interface-Log (TIL) data model and algorithms for deriving process-related data provides the needed functionality and underlies the implementations described here (Lucas and Babaian, 2012; Babaian and Lucas, 2013a).

We have previously reported on earlier versions of the individual components of the work presented here (Babaian and Lucas, 2013b, Lucas et al., 2013). The contribution of this paper is in

- presenting a prototype that incorporates enhanced versions of the Interactive Process Visualization and Automated Playback features for on-going, continuous support of ERP users,
- explaining how these components and their integration within the prototype implement the design principles for achieving better ERP usability.

In the next section of the paper, we introduce the four design principles for improving the usability of ERP systems. Section 3 presents our design interventions implemented in an ERP prototype, and provides an overview of the TIL model, which makes those implementations possible. Related work is presented in Section 4, followed by conclusions and directions for future research in Section 5.

2 DESIGN PRINCIPLES

In the process of identifying usability issues with ERP systems, we interviewed and observed 33 enterprise system users in three organizations. The human-computer collaboration paradigm (Terveen, 1995) provided a unifying perspective for analyzing the usability issues and grouping them based on their underlying causes. The outcome of this evaluation was the four design principles shown in Figure 1. The "transactions" that are mentioned in these principles are typically referred to as "tasks" throughout this paper.

Design Principle 1 focuses on aligning the way information is presented to users with the internal conception users have developed based on organizational practices. Even experienced users often maintain lists of system terms side by side with the terms used in practice. The extensive and prolonged training typically required before employees can put ERP systems to use in their work is necessitated in large part by generic interfaces overflowing with components bearing unrecognizable labels, many of which are irrelevant to the user. Customizations addressing these issues are lost with system upgrades: hence the requirement for a means for incorporating earlier customizations.

Design Principle 2's call for navigational aids also addresses system learnability and the need for users to be able to recognize what to do. The deep menu structure of many ERP systems makes it virtually impossible for users to find the necessary transaction without either referring to detailed documentation they and their colleagues maintain or memorizing transaction codes. In addition, ERP interfaces often hide the underlying workflow from the user, including where the user is in the current process, what has been accomplished, and what steps remain. This can lead users to think they have completed a process when in fact that is not the case, leading to significant logistical problems down the road. Progress guidance is therefore another essential element for improving usability.

1. The user interface should provide a mechanism for customizing the vocabulary of terms used by the system in its communication to the user, the composition of business transactions, and the content of the system's informational output to match the practices of the organization. There should be a mechanism for incorporating the customizations from an earlier version of the system to a later one.
2. The system should provide navigational and progress guidance to a user performing a transaction, indicating the broader context of each interaction in terms of the related business process components and specifying the completed and remaining parts. A sufficiently competent user should be able to turn off this guidance if it becomes a distraction.
3. When the system detects a problem, it should identify the possible causes and ways of resolving it. If the fix is obvious, the system should inform the user and perform it. If it isn't obvious, the possible causes and resolution scenarios should be presented to the user and be readily executable. If the system is unable to identify resolution strategies, it should present the user with the relevant data and transactions.
4. In presenting selection choices, the system should utilize what it knows about the user, the organization, the task, and the context, and provide faster access to the more likely choices than the less likely ones. Where the choice of data or action is obvious, the system should have an option of not waiting for the user to enact it. The user should have an option to replace/cancel the system's provided choice of data/action.

Figure 1: Design principles for collaborative ERP systems.

Design Principle 3 deals with error situations, which can stymy even the most sophisticated of users. Oftentimes the error message fails to identify the cause of the error, much less presents strategies for resolving it, even when the system has access to the pertinent information. For significant errors, users often need to turn for help to a colleague with the appropriate expertise, adversely affecting the productivity of both parties.

Lastly, Design Principle 4 calls on the system to utilize all the information it has access to in presenting options and data to the user. In conducting a search, for example, a large, unfiltered list is often displayed rather than a selection that has been filtered based on the information entered in other fields. Adhering to this design principle would save time for the user and lessen the frustration that arises from having to specify obvious information.

The design interventions presented in this paper were driven by the requirements of Design Principle 2 while also supporting Design Principle 4, as described in the next section.

3 PROVIDING NAVIGATIONAL AND PROGRESS GUIDANCE

ERP systems provide modules and functions to enable and facilitate a large number of business processes used in modern organizations. Many of these business processes are complex, involving multiple paths, tasks, data objects, users, business units, etc. Each task may also require multiple pages or forms, which contain many fields, tables, menu options, and tabs.

One consequence of the complexity of these systems is that they cannot be put to successful use without extensive training, as reported in ERP studies (e.g., Scott, 2005) and noted frequently in our previously mentioned field studies. Even after training, users typically consult their colleagues when learning to operate an ERP interface for an unfamiliar task. In addition, they often create notes to help themselves and their coworkers learn details of how to perform tasks within the system (Topi, Lucas, and Babaian, 2006). These notes come in various forms, including flowcharts, step sequences, and annotated sequences of screenshots. They are remarkably well maintained (some are even laminated) in order to withstand heavy usage.

A time- and resource-saving alternative is for the system to provide intuitive, interactive, and context-sensitive navigational support and guidance that help users perform their tasks effectively and efficiently. Although system help menus and high-level process definitions (e.g., SAP's Business Blueprint) may exist, they are not usually interactive and do not alleviate the cognitive burden associated with learning to perform tasks and navigate through processes (Hipp, Mutschler, and Reichert, 2012). The goal of the design interventions described next is to provide this crucially needed support and guidance.

3.1 Interactive Process and Instance Visualizations

The visualization component in our ERP prototype was implemented based on Design Principle 2, which is intended to help ERP users navigate through tasks involved in a business process. The behavior of this component also adheres to Design

Principle 4 by presenting visualizations that take into account the context of the interaction without requiring the user to specify the obvious process and process instances of interest.

A business process is rendered as an interactive, clickable graph, which consists of two types of objects: nodes and links. Each node represents a task or a task instance (an execution of a specific task); and each link represents a data flow from one node to another. We distinguish between two types of graphs: a process graph and a process instance graph.

A *process graph* specifies what tasks are included in a business process and how data flow through those tasks. Processes in commercial ERP systems are usually predefined by system vendors or configured and customized by adopting organizations. A process definition reflects how an organization believes a process should typically be carried out. Processes supported by our prototype are also pre-specified in the TIL model as a composition of tasks that can be configured to match the practices of the organization, in accordance with Design Principle 1. Process graphs are derived from these static representations of processes within the TIL model, which maintains data on the tasks (e.g., IDs, names, descriptions, and output document types) and process structures (e.g., component tasks and data flow links).

A *process instance graph* represents a specific execution of a process corresponding to a set of task instances. Process instance graphs are dynamically constructed, based on the system's usage logs in the TIL model, by tracing the input/output chains backward from the current task instance to the starting instance.

A user can interact with a process or process instance graph by clicking on an object (node or link) to view its details. Through interactions with the visualized graph, a user can learn important contextual information about:

- how tasks are formed into a process (e.g., what has been done and what comes next),
- how documents flow between tasks,
- how a specific process instance is executed,
- users involved in a process instance, and
- task-specific information (e.g., pages to be filled out and the completion status of a task).

Henceforth, we will use material purchasing as the exemplar process to illustrate how the visualization component in our prototype implements Design Principle 2 while following Design Principle 4 in providing navigational support and guidance without users having to request it.

Figure 2 presents our prototype's typical interface, which is divided into two sides. The left-hand side is for selecting a task and entering parameters for performing it. The right-hand pane is for visualizing the process and instance graphs underlying the task. This pane can be detached and minimized for users who do not wish to use it.

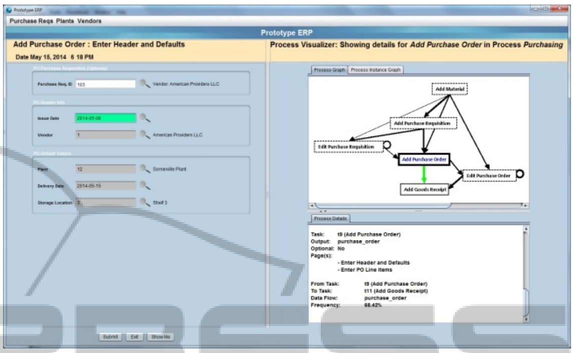

Figure 2: The system interface.

In this figure, the Enter Header and Defaults page is shown after the user has logged in and selected the Add Purchase Order task. The required fields (e.g., the Issue Date) are highlighted in green, while fields automatically populated by the system are shown in grey.

The right-hand pane is split into two panels. The upper panel contains two tabs, with one for the process graph and the other for the process instance graph. The lower panel is for displaying detailed information about graph objects selected by the user.

3.1.1 Visualizing Processes

Figure 3 presents only the right-hand side of the interface from Figure 2. It draws the graph for the purchasing process, in which the Add Purchase Order task is a component. Each box stands for a task node object and each arrow represents a data flow link object. Boxes with solid borders are mandatory tasks in this process, and boxes with dashed borders are optional. The box corresponding to the current task (Add Purchase Order) is highlighted with a heavy border and blue label. Two tasks (Edit Purchase Requisition and Edit Purchase Order) have circles attached to them, meaning that they are recursive and can be repeated many times. The links in the graph are weighted based on the frequency of the data flow.

When the user clicks on any object in the graph, the object's color will change to green and its details will be displayed in the lower panel. For example, the green link currently selected carries the output document (purchase_order) from Add Purchase Order and provides the input to the Add Goods Receipt task; and for 68.4% of the time a goods receipt is posted directly after a purchase order is added without the editing task being performed. This panel also displays details about the previously selected Add Purchase Order task in terms of its output document name, whether it is optional, and the two pages that need to be filled out.

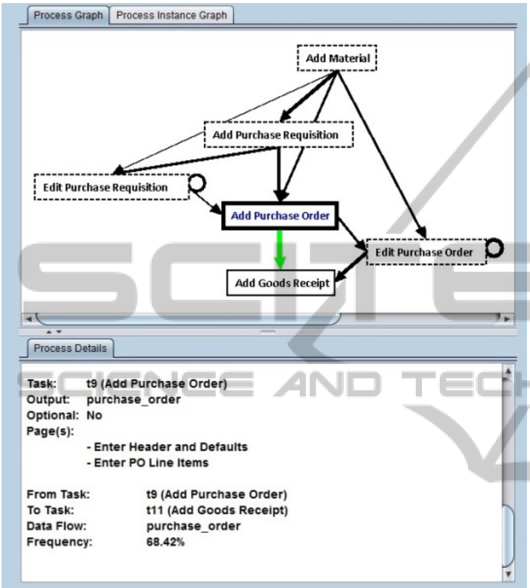

Figure 3: A process graph.

The visualization of a process graph can enhance the user's context awareness. In the above example, a user can easily see that the Add Purchase Order task is a required component within the business process for purchasing materials, that several tasks (e.g., Add Material and Add Purchase Requisition) may precede this task, and that more tasks (e.g., Edit Purchase Order and Add Goods Receipt) may follow it. The user can also get a clear idea about what data are needed as inputs for a task and what document will be generated as the output. For example, the inputs for Add Purchase Order include material data and a purchase requisition document. Moreover, link frequencies calculated based on system usage logs can, to some extent, serve as a recommendation for the most "popular" path of tasks among alterative paths in a process. The task details provide additional contextual information about whether a task is optional and which pages are required for it.

3.1.2 Visualizing Process Instances

The upper panel in Figure 4 displays the process instance graph tab for an Add Goods Receipt task instance. The graph for the purchasing process, of which the Add Goods Receipt is a component, can be viewed in the process graph tab (Figure 3).

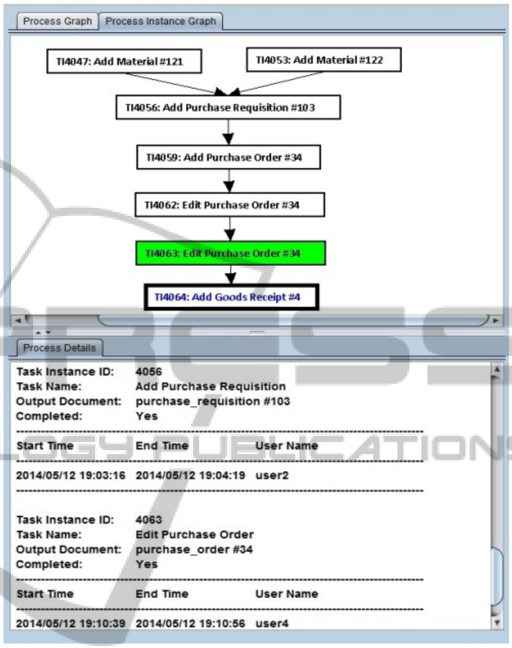

Figure 4: A process instance graph.

In the instance graph, each box represents a specific task instance labeled by its instance ID, task name, and the output document number. As in the process graph, the current task instance (TI4064: Add Goods Receipt #4) is highlighted with a heavy border and blue label. The links are document flows between task instances. It is clear that this process instance started with two materials (#121 and #122) being added in two task instances, TI4047 and TI4053, respectively. A purchase requisition (PR#103) was then created in task instance TI4056, based on which a purchase order (PO#34) was created in TI4059. Subsequently, PO#34 underwent two rounds of editing in TI4062 and TI4063. In the last step, a goods receipt (GR#4) was posted in TI4064.

Unlike the process graph in Figure 3 the instance graph includes only those task instances that were actually involved in the specific execution of the process. Tasks such as Edit Purchase Requisition are not part of this process instance's history.

The lower panel in Figure 4 shows detailed information about two selected task instances:

TI4056 and TI4063. By reviewing the task instance details, the user can quickly learn the instance ID, task name, output document number, status (complete or incomplete), the start and end times, and the user names of those performing the task instance. The system's ability to identify each of the task instances and who performed them is especially useful in error situations, as the current user can trace back through an instance graph to identify the sources of an error and the users who may be able to help resolve it.

3.2 Automated Playback

In developing the automated playback component and following the directive of Design Principle 2 to supply process guidance, we have sought to build into the system the capability to provide users with a feature for reviewing how a task or a business process instance was performed in the past. As previously noted, users often turn to their colleagues for demonstrations of how to complete new or infrequently performed tasks. With the automated playback feature, the user first selects a task or process instance. That instance is then replayed step-by-step by the system as originally executed, thus serving as a form of a tutorial. This tutorial is created dynamically by the system without any human intervention, based only on the automatically collected records of past system-user interactions stored in the TIL model, as described in detail in (Lucas and Babaian, 2012; Babaian and Lucas, 2013a). Access to this automated playback is provided from every task page via a click on a *Show Me* button.

A user can specify parameters to narrow down the list of task instances for replay by specifying a particular user, time window, or specific document produced by the task instance. However, even with filtering by search parameters, finding the appropriate task instance may yield a large set of instances. In the prototype, we extended the playback interface by providing the user with the option to review the associated process instance prior to selecting it for replay. As shown in Figure 5, when a user clicks on a task instance from the list on the left-hand side, the system automatically reconstructs the process instance graph that includes the chosen task instance and presents it in the process visualization pane on the right. This mechanism follows the directive of Design Principle 4 to use the information that is available to the system to simplify the selection task for the user. Upon the user submitting the selected task instance, the system runs an animation of that instance in a separate window. A video demonstration of the playback feature can be viewed by accessing the website at *http://cis.bentley.edu/ERP/demos.html.*

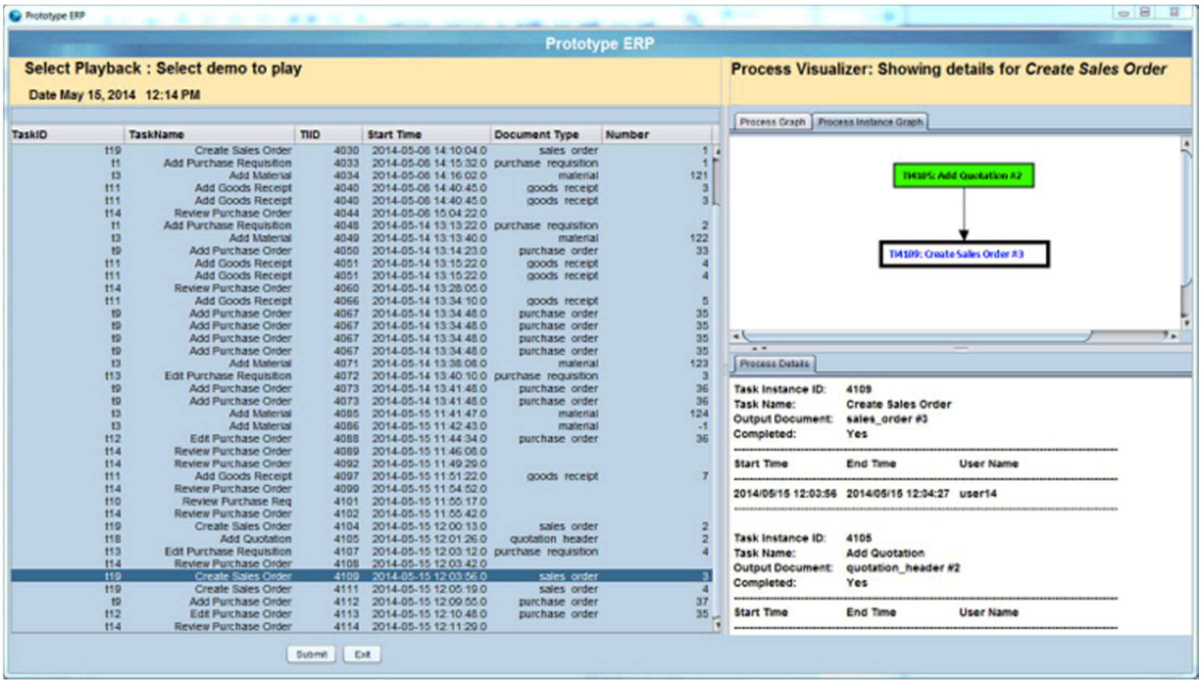

Figure 5: Integrated view of task instances for playback and the corresponding process instance visualization.

3.3 Supporting Infrastructure

The user support features presented here require system access to the history of system-user interactions. This history, which in our prototype is automatically collected during the regular course of use, is recorded within the aforementioned Task-Interface-Log (TIL) model. While commercial ERP systems typically include several logging facilities, the information available from their data logs is not sufficient to enable the kinds of user support presented here. The TIL model was specially designed (Lucas and Babaian, 2012; Babaian and Lucas, 2013a) to enable the system to be aware of its own functionality and interface components, as well as the users' interactions with those components.

The specification of the task and interface models within the TIL model allows user interaction data to be captured and later reconstructed within a broader task and process context, as is the case in the two interactive user interface components we have presented here. The TIL data provides the system with a rich infrastructure for runtime reasoning about the relationships between tasks, processes, domain data, and users, and for extracting knowledge about system usage. In addition to the history of task instances and records of system-user interactions, which are primarily exploited by our prototype, the data within the TIL model can also serve as a basis for other system interventions that implement the design principles presented here. For example, the data can be used to assess the user's familiarity with a task, find the most commonly entered values, and determine a set of related tasks. Furthermore, the TIL model makes it possible to specify the composition of processes from tasks and to adjust the labels on input fields to match a specific organization's practices and vocabulary.

4 RELATED WORK

Industry reports acknowledge that ERP systems are rife with usability problems (Hestermann, 2009; Matthews, 2008; Otter, 2008 Iansiti, 2007), but there is a dearth of research on means for improving the user experience. A study by Parks (2012) shows that interface complexity has a significant impact on task time. A recent survey-based study of ERP users (Lambeck et al., 2014) confirms that locating the desired functionality within a system remains a significant usability obstacle. The study also finds that visualization and error support reduce the perception of complexity of the interface.

Process visualization is a natural fit for aiding navigation and improving understanding of business processes. The advent of process-aware information systems (PAISs) has led to large collections of process models. Hipp, Mutschler, and Reichert (2012) note that process models are typically presented as static drawings. They present a framework for navigating large process spaces at varying levels of detail. The *proView* framework (Kolb and Reichert, 2013) provides personalized views from large collections of business processes and process instances by showing only those activities in which the user is involved. However, none of the frameworks that we found in literature has been integrated in an interactive way within an enterprise system for aiding end users during active system use, as ours has been.

In addition to their application for process discovery (van der Aalst, 2011), event logs have been used for providing context-aware assistance to users. Dorn et al. (2010) describe an approach to providing context-sensitive recommendations on the most suitable next step in flexible, people-driven processes. Schnonenberg et al. (2008) propose a recommendation service for use with flexible PAISs for providing users with recommendations on possible next steps based on past process executions.

Plaisant and Shneiderman (2005) note that recorded demonstrations of interfaces are very effective for helping users learn procedural tasks. These tutorials, however, are typically prerecorded demonstrations, not integrated with the user interface. CoScripter (Leshed et al., 2008) enables users to record their interactions with websites as editable, executable scripts, which are stored in a wiki for sharing among users. ActionShot (Li et al., 2010) provides visualization interfaces for exploring and searching the user's detailed history of interactions within a web browser for sequences of actions. Actions can be shared via email or the web, or converted into a CoScripter script and stored on the CoScripter wiki. Studies of CoScripter usage indicate that enterprise users chose to incorporate these process scripts into their work practices.

5 CONCLUSIONS AND FUTURE WORK

We have presented a prototype ERP system that integrates two kinds of user support features: process visualization and automated playback of previous task interactions. These features are included alongside the usual ERP functionality for providing additional, and much needed, support to the users. Inclusion of this type of interactive process and task guidance distinguishes our work from other business process mining and visualization research.

The presented prototype serves as an illustration of possible ways of implementing some of our design principles for achieving better ERP usability by focusing on improving the collaborative capability of the system. Furthermore, the presented interventions, which were designed using the TIL model and algorithms, confirm the suitability of TIL as a framework for implementing a variety of features supporting system-user collaboration.

The design principles and interventions were motivated by field studies of ERP usage. Thus, the next step in this research involves evaluating the proposed solutions with users and fine-tuning the solutions based on their feedback. We are also working on optimizing the algorithms for logging and information extraction from the TIL model.

REFERENCES

Babaian, T., Lucas, W., 2013a. Modeling data for enterprise systems with memories. *Journal of Database Management*, 24(2), 1-12.

Babaian, T., Lucas, W., 2013b. Composing interface demonstrations automatically from usage logs. *LNBIP*, vol. 141, Springer, 376-392.

Babaian, T., Lucas, W., Xu, J., Topi, H., 2010. Usability through system-user collaboration: Deriving design principles for greater ERP usability. *LNCS*, vol. 6105, Springer, 394-409.

Calisir, F., Calisir, F., 2004. The relation of interface usability characteristics, perceived usefulness, and perceived ease of use to end user satisfaction with enterprise resource planning (ERP) systems. *Computers in Human Behavior*, 20(4), 505-515.

Dorn, C., Burkhart, T., Werth, D., Dustdar, S., 2010. Self-adjusting recommendations for people-driven ad- hoc processes. In *Proc. of BPM'10*, *LNCS* vol. 6336, Springer-Verlag, 327-342.

Ganly, D., Montgomery, N., 2013. Hype Cycle for ERP, 2013. *Gartner, Stamford, CT*.

Grosz, B.G., 2005. Beyond mice and menus. In *Proceedings of the American Philosophical Society*, 149 (4), 529-543.

Hestermann, C., 2009. Key issues for enterprise resource planning. *Gartner, Stamford, CT*.

Hipp, M., Mutschler, B., Reichert, M., 2012. Navigating in complex business processes. In *Proc. of DEXA'12*, Part II, Springer, 466-480.

Iansiti, M., 2007. ERP End-user business productivity: A field study of SAP & Microsoft. http://download.microsoft.com/download/4/2/7/427edce8-351e-4e60-83d6-28bbf2f80d0b/KeystoneERPAssessment Whitepaper.pdf

Kolb, J., Reichert, M., 2013. Data flow abstractions and adaptations through updatable process views. In *Proc. of SAC'13, EE'13*. ACM, 1447-1453.

Lambeck, C., Fohrholtz, C., Leyh, C., Šūpulniece, I., Müller, R., 2014. Commonalities and contrasts: An investigation of ERP usability in a comparative user study, In *Proc. of ECIS-2014*.

Leshed, G., Haber, E.M., Matthews, T., Lau, T., 2008. CoScripter: Automating & sharing how-to knowledge in the enterprise. In *Proc. of CHI '08*, ACM, 1719-1728.

Li, I., Nichols, J., Lau, T., Drews, C., Cypher, A., 2010. Here's what I did: Sharing and reusing web activity with ActionShot. In *Proc. of CHI'10*, ACM, New York, NY, USA, 723-732.

Lucas, W., Babaian, T., 2012. Implementing design principles for collaborative ERP systems. *LNCS*, vol. 7286, Springer, 88-107.

Lucas, W., Xu, J., Babaian, T., 2013. Visualizing ERP usage logs in real time. In *Proc. of the 14th International Conference on Enterprise Information Systems* (ICEIS '13), 83-90.

Matthews, D., 2008. Usability as an ERP Selection Criteria, *IFS*. http://ifs.datahost.com/shop/images/wp-usability.pdf

Otter, T., 2008. Case study: Ness combines consumer application ease of use with ERP robustness. *Gartner, Stamford, CT*.

Parks, N. E., 2012. Testing & quantifying ERP usability. In *Proc. Of RIIT '12*. New York, USA, ACM, 31-36.

Plaisant, C., Shneiderman, B., 2005. Show me! Guidelines for producing recorded demonstrations. In *Proc. of VL/HCC 2005,* IEEE Computer Society, 171-178.

Schonenberg, H., Weber, B., van Dongen, B., van der Aalst, W., 2008. Supporting flexible processes through recommendations based on history. In *Proc. of BPM 2008*. *LNCS*, vol. 5240. Springer, Heidelberg, 51-66.

Scott, J. E. 2005. Post-implementation usability of ERP training materials: The user's perspective. *Information Systems Management*, 22(2), 67-77.

Singh, A., Wesson, J., 2009. Evaluation criteria for assessing the usability of ERP systems. In *Proc. of SAICSIT* 2009, ACM, 87-95.

Terveen, L.G., 1995. An overview of human-computer collaboration. *Knowledge-Based Systems Journal, Special Issue on Human-Computer Collaboration,* 8(2-3), 67-81.

Topi, H., Lucas, W., Babaian, T. 2006. Using informal notes for sharing corporate technology know-how. *European Journal of Information Systems,* 15(5), 489-499.

van der Aalst, W. M. P., 2011. *Process mining: Discovery, conformance, and enhancement of business processes,* Heidelberg, Springer-Verlag.