COLLABORATING TO IMPROVE ERP USABILITY

Tamara Babaian, Wendy Lucas, Heikki Topi
Bentley College, Computer Information Systems Department, Waltham, MA, USA

Keywords: Usability, Enterprise systems, ERP, Collaboration theory, Collaborative user interfaces

Abstract: Anecdotal evidence strongly suggests that enterprise resource planning (ERP) systems have unintuitive user interfaces that hinder usability, frustrate users, and ultimately interfere with their successful adoption and utilization in organizations. Despite the huge costs associated with poorly implemented systems, ERP usability has received little attention from the research community. It is our contention that existing theories on usability must be extended to address the unique challenges resulting from the size, complexity and integrated functionality of these industrial behemoths. This paper discusses collaboration theory as a potentially beneficial way to conceptualize the relationship between the user and the system and to provide a foundation on which interfaces can be developed that enhance user performance and satisfaction with ERP systems.

1 INTRODUCTION

A recent study carried out by Forrester Research (Chew, Orlov, & Herbert, 2003) evaluated eleven different ERP products, including SAP, PeopleSoft, Oracle, JD Edwards, Microsoft and Lawson. Its findings confirmed the poor usability characteristics and unintuitive user interfaces of these systems, which contribute to decreased productivity and increased costs for businesses using them. In trying to perform a number of standard tasks that should have been "straightforward" without any training, the analysts from Forrester found that several of these tasks required "inordinate patience and expertise" to complete (Gilbert, 2003). The overall conclusion was that "users should demand better usability." Yet, there has been little movement to date toward improving the design of the user interface components of these systems by either ERP vendors or the usability community as a whole. The clear need for efforts directed at this task provides the motivation for the research initiative described here.

Given the time, effort, and money expended on implementation and training, it is surprising that so little attention has been focused on understanding the ways in which users interact with ERP software and the degree to which the interaction model supports the tasks being performed. In this paper, we suggest that collaborative user interfaces (Grosz, 1996; Shieber, 1996; Grosz & Kraus, 1996) provide a means for addressing the gap between the capabilities of the ERP system and a means for harnessing those capabilities for meeting each user's individual objectives. [Note: by "collaborative user interfaces," we are referring to collaboration between the user and the computer as opposed to between users, which is commonly referred to as computer-supported cooperative work (CSCW).] The novelty of our research lies in its emphasis on the relationship between collaborative support and task performance and satisfaction; that is, the more capable the technology is at recognizing the user's goals and collaborating to reach them, the higher the user's perceived usefulness of that technology and ability to use it effectively will be.

2 RELATED RESEARCH

The need for understanding the relationship between how a person interacts with a system and her perceptions of the tasks that the system needs to accomplish is often noted in research on user interface development (see, for example, Stary, 1999). The importance of designing interfaces with the personal goals of the user in mind, rather than

Babaian T., Lucas W. and Topi H. (2004).
COLLABORATING TO IMPROVE ERP USABILITY.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 164-168
DOI: 10.5220/0002655101640168

the system-prescribed method for achieving those goals, is also emphasized by Cooper (1995). A review of current literature yielded few studies that discuss these issues in the context of ERP or other enterprise systems, and none specifically focusing on interface design or usability. Kleiner et al. (1999) do, however, raise some of the human factors implications of ERP systems, including the lack of attention paid to training and the maze of screens one has to navigate.

A recent annotated bibliography on ERP system research by Esteves & Pastor (2001) does not include any references to papers that directly discuss ERP usability. Similarly, a comprehensive collection of state-of-the-art articles on usability (Jacko & Sears, 2003) does not include any articles discussing usability issues of ERP systems, enterprise systems, or any other administrative organizational systems.

While there is little research addressing ERP usability, user interface research in general has made considerable advances, as evidenced by collections such as Jacko & Sears (2003) (see above), and a large number of innovative interface types. Although a number of experimental interfaces have even found their way into practice, they have not made inroads into ERP systems to the best of our knowledge. Applying the scientific and technological advancements that have been made in user interface research to these systems holds great promise for improving their usability.

3 COLLABORATION THEORY

In this paper, we contend that collaboration theory in the context of human-computer interaction could be applied as a set of guiding principles to enterprise-level administrative systems, such as ERP systems. It has been suggested (Grosz, 1996; Shieber, 1996) that to provide an adequate level of support to users in the increasingly complex environments of modern systems, human-computer interaction should move from a master-slave model, in which the human user issues commands to the system, to a model based on collaboration between the system and the user. In other words, the computer system should be viewed and act as the user's true partner in the process of goal achievement. This view of a system-collaborator, supported by a philosophical account of cooperative activity (Bratman, 1992) and by more formal mathematical frameworks of such a collaboration (Grosz & Kraus, 1996), has already been used in the design and implementation of collaborative systems (e.g., Babaian, Grosz, & Shieber, 2002; Rich, Sidner, & Lesh, 2001). None of these systems, however, have been of the organizational administrative ilk.

Before we can present a concrete example illustrating collaborative behavior, we need to specify the characteristics defining what constitutes that behavior at the conceptual level. As defined by Bratman (1992) and further elaborated for computational use by Grosz & Kraus (1996), collaborative behavior is identified by the following principles:

- •***Commitment to the joint activity.*** Each party recognizes and is committed to the joint activity. As part of this commitment, the parties need to be aware of the context surrounding their collaboration because it may be important in determining the finer details of that activity.
- ***Mutual responsiveness.*** Each participant seeks to adjust his behavior based on the behavior of the other and guided by the commitment of both to the joint activity. Combined with the commitment to the joint activity, mutual responsiveness entails that the parties may have to adapt their actions for the benefit of the more optimal joint outcome.
- ***Commitment to mutual support.*** Each party is committed to supporting the efforts of the other. When an agent knows the other party may need help in performing a subtask related to their shared activity and is able to provide such help, the agent is ready to assist and the other party recognizes and supports such assistance. Commitment to mutual support also implies communication with the purpose of sharing information important to the completion of the joint activity.
- ***Meshing subplans***. The parties should seek to decompose the task into mutually meshing, although independent, subplans. The parties must thus engage in communication to coordinate their independent subplans at certain times, as the need arises.

We believe that collaboration theory is an excellent foundation for usability design and evaluation because:

- It directly addresses the process of cooperative problem solving in a systematic way by describing a set of requirements and procedures that must be in place to achieve successful collaboration.
- It provides a framework and explains the benefits of many existing user interface

practices and developments that improve system usability. A simple example of a collaborative interface practice is the highlighting of an incorrectly spelled word. An example of a larger development is adaptive interfaces (Rogers, Flechter, & Langley, 1999) that learn over time to adjust their appearance/behavior based on the history of use, thereby implementing the mutual responsiveness principle of collaboration.

- As noted by John (1996), one of the challenges of usability research is to find evaluation frameworks that can simultaneously be used to guide design choices. Collaboration theory satisfies this criterion because it can serve as a design guide for interface development, as shown in Ortiz & Grosz (2002), Rich, Sidner, & Lesh (2001), and Babaian, Grosz, & Shieber (2002).

4 TRANSACTION TASK EXAMPLE

In order to illustrate the application of collaboration principles to the design of a user interface, we present an example of a typical ERP-user interaction. Based on these principles, we point out how the interface fails to support the user in achieving her goal. We also suggest how the system could be modified to be a better collaborator.

Pat is an engineer and a relatively new user of a large enterprise system. As part of her engineering assignment, Pat needs to order a certain hardware component. She tries to create a purchase requisition, but is stymied when she can't specify the item to be ordered because it is not listed in the Material Master.

The option of adding a new part to the Material Master is not available in the purchase order interface, although its implementation exists and is available elsewhere in the system interface. Interface design based on collaboration should recognize the broader context in which the task of creating a purchase requisition may occur. Based on the mutual support principle, the system should provide easy access to related or prerequisite tasks, such as adding a new part in the context of a purchase requisition.

Pat has to scrap the unfinished purchase requisition, enter the part into the Material Master, and then proceed to create the purchase requisition again. To create a new purchase requisition, Pat follows this menu path: Logistics – Material Master – Purchasing – Purchase Req. – Create. She enters information regarding the delivery date, the plant to which this part must be delivered, the storage location, and the purchasing group.
When Pat presses Ok to move to the next screen, the system complains: "Date period D is not valid." Pat goes back to the date field and tries to modify the date specification. Reading the system help on various formats fails to explain how the D, T, W or M options affect the format of the date to be entered (particularly since Pat does not recognize the use of the letter 'T' for 'Date' in German), so she is puzzled for a while until she stumbles upon the Possible Entries option that is available for the date field. Selecting this option results in the system displaying a calendar from which Pat selects a date, which is then correctly entered for her by the system into the date field.

Although the interface includes the very useful option of selecting the date from a calendar, this option is not offered and remains obscured even though the system has detected and reported the user's error. Commitment to mutual support and mutual responsiveness would require a system-collaborator that has the ability to offer such help when it can provide it, instead of merely informing the user about a failure.

A colleague then suggests that Pat select the Model service specifications option, which displays the actual names of all items listed in the form in addition to their numeric identifier. Pat finds this option very helpful for both clarity and verification purposes, and opts to use it.

Commitment to mutual support requires that the collaborating parties share the knowledge that is relevant to the success of the joint activity. In the previous example, even though displaying the item names in addition to the identifiers would be more informative from the perspective of a human user and is very easy for the system to do, the interface does not provide this information without a specific request. Typically, new users are not aware of all of the available options, and thus fail to take advantage of these types of capabilities.

Pat verifies that the information she has entered, including the destination plant for the part, is correct, Pat confirms this to the system and is

taken to the next screen, where she is asked to list the items to be ordered. Unfortunately, Pat has forgotten the exact ID number of the part she just entered into the Material Master.

If the system kept track of the steps Pat had taken previously and used this information to examine the context of the current interaction, it would be able to recognize that, having just entered a new part, Pat is likely to need to refer to this part's information when she follows with the purchase requisition.

Therefore, she tries to find it by reviewing the item descriptions using the Possible Values option for the item field. At some point during this review process, the information on the screen changes completely. Pat is unsure what she has done to cause this change and wonders whether or not the information on the purchase requisition is still available.

The rapid and drastic change on the screen creates an impression that the purchase requisition task has been abandoned. Pat is now unsure of whether the system is still committed to the joint activity of creating a purchase requisition. This situation demonstrates the need for the system to convey the future steps (plan) in performing the task as well as the history of the steps performed and the context of the most current interaction. Collaborators need to communicate in order to make sure their mutual plans for achieving the shared goal are coordinated.

After an initial moment of panic, she discovers that she can still get to the list of items in the purchase requisition by using the Go Back button, and heaves a sigh of relief.

There would be no panic if Pat knew exactly where she is in the process, in other words, if she was kept aware of the plan by the system-collaborator, and knew how to get back to the previous steps.

There are more than 12 available options for displaying the material lists – too many for Pat to make use of them all.

Pat has just provided the system with information regarding the plant for which the part is being ordered. The system should be able to infer that the list of parts for this plant should be most useful for the search, and perhaps rate that option higher than other searches for parts.

Feeling overwhelmed by choices, Pat finally notices an option for displaying parts by plant and, in reviewing the material list for the destination plant, locates the description of the part there. Upon specifying the quantity, Pat is done creating this document. She feels unsure, however, if the information she has entered is complete enough because there are a lot of other fields on the form that she has left blank. After consulting the help desk, Pat concludes that the purchase requisition is complete and saves it.

The system knows which fields are optional. It should communicate this knowledge clearly to the user, because this would help Pat complete the task with confidence.

Note that as the critique of the above scenario shows, the principles of collaboration influence both the design of the static components of the interface as well as the dynamics of human-computer interaction extending over the entire process of completing a purchase requisition for a new part. The effect on the static components of the interface, such as the content and layout of the screen, is manifested, for example, by including the option of adding a new part to the purchase requisition interface and by displaying information that is certainly going to help the human user (e.g., field optionality) without waiting for a specific request. The collaboration principles should also guide the system behavior over the process of working with the user on a task, by considering a broader context of each simple interaction, as when the system may recognize that the part number of the newly added part may be used in the purchase requisition that follows.

5 DISCUSSION

It has been argued by the usability researchers as well as the researchers in the collaborative interfaces community that design for usability cannot be achieved by a local change in the interface. Collaboration cannot be "patched on" and must be designed from the start. The influence on the design is not limited to the system front-end: to implement a collaborative nature of the interaction generally requires appropriate support in the data model and the algorithmic modules of a system. Investigating the design principles and the resulting representational and algorithmic needs stemming from the human-computer collaboration model of the interface is especially interesting and important in the context of the enterprise-wide

systems, not only because of the obvious shortcomings of the ERP interfaces. These systems span an enormously broad domain of organizational tasks, while most tasks involve multiple logical and physical system modules, and there are multiple users with varying demands and expertise levels. All of these issues present a challenge to the interface design for usability. We are currently working on the design and implementation of a prototype involving several categories of ERP tasks to demonstrate how collaboration principles can be used to address these issues.

6 CONCLUSIONS AND FUTURE WORK

Improving the usability of ERP systems provides benefits that extend well beyond meeting the needs of individual users; it benefits the organization as a whole by reducing the length of the training time, improving employee satisfaction, and providing valuable information on overall system usage. This paper has argued that collaboration theory is a highly relevant conceptual framework that can be used effectively to guide user interface development work in the context of large-scale enterprise systems. Future research on ERP and enterprise system usability should address both the technical issues related to user interface design as well as the overall impact of ERP interfaces on organizational decision-making.

ACKNOWLEDGMENTS

This work was funded by a grant from Bentley College. We gratefully acknowledge this support.

REFERENCES

Babaian, T., Grosz, B. J., & Shieber, S. M., 2002. A writer's collaborative assistant. In Y. Gil & D. B. Leake (Eds.), Proceedings of the 2002 International Conference on Intelligent User Interfaces (IUI-02) (pp.7–14). New York: ACM Press.

Bratman, M. E., 1992. Shared cooperative activity. The Philosophical Review, 101(2), 327–341.

Chew, J,. Orlov, L, & Herbert, L. (2003). App User Interfaces Still Need Work, a technology brief from Forrester Research. Accessed on 1/26/2004 at http://www.forrester.com/ER/Research/ Brief/Excerpt/0,1317,16184,00.html.

Cooper, A., 1995. About Face: The Essentials of User Interface Design (First ed.): IDG Books Worldwide, Inc.

Esteves, J. & Pastor, J., 2001. Enterprise Resource Planning Systems Research: An Annotated Bibliography. Communications of the AIS, 7(8).

Gilbert, A., 2003. Business apps get bad marks in usability, CNET News.com. Accessed at http://news.com.com/2100-1017-980648.html on 2/12/2003.

Grosz, B. J., 1996. Collaborative systems. AI Magazine, 17(2), 67–85.

Grosz, B. J. & Kraus, S., 1996. Collaborative Plans for Complex Group Action. Artificial Intelligence, 86(2), 269–357.

Jacko, J. A., & Sears, A. (Eds.), 2003. The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications. Mahwah, new Jersey, London: Lawrence Erlbaum Associates.

John, B. E., 1996. Evaluating usability evaluation techniques. ACM Computing Surveys, 28(4es), 139.

Kleiner, B. M., Bishu, R. R., Drury, C. G., Madhu, N., Getty, R., & Muralidhar, A., 1999. Human factors concerns in enterprise resource planning (ERP) solutions. Paper presented at the Human Factors Engineering and Ergonomics Conference, Houston, TX.

Ortiz, C. L. & Grosz, B. J., 2002. Interpreting information requests in context: a collaborative web interface for distance learning. Autonomous Agents and Multi-Agent Systems, 5, 429-465. Kluwer Academic Publishers.

Rich, C., Sidner, C. & Lesh, N., 2001. COLLAGEN: Applying Collaborative Discourse Theory to Human-Computer Interaction, AI Magazine, Special Issue on Intelligent User Interfaces, 22(4),15-26.

Rogers, S., Flechter, C.-N., & Langley, P., 1999. An adaptive interactive agent for route advice. In O. Etzioni, J. P. Müller, & J. M. Bradshaw (Eds.), Proceedings of the Third International Conference on Autonomous Agents (Agents'99) (pp. 198–205). Seattle, WA, USA: ACM Press.

Shieber, S. M., 1996. A call for collaborative interfaces. ACM Computing Surveys, 28(4es), 143–143.

Stary, C., 1999. Toward the Task-Complete Development of Activity-Oriented User Interfaces. International Journal of Human-Computer Interaction, 11(2), 153-182