

## A Collaboration Model for ERP User-System Interaction

Jay Coopriider  
Bentley University  
Waltham, MA  
jcoopriider@bentley.edu

Heikki Topi  
Bentley University  
Waltham, MA  
htopi@bentley.edu

Jennifer Xu  
Bentley University  
Waltham, MA  
jxu@bentley.edu

Martin Dias  
Bentley University  
Waltham, MA  
dias\_mart@bentley.edu

Tamara Babaian  
Bentley University  
Waltham, MA  
tbabaian@bentley.edu

Wendy Lucas  
Bentley University  
Waltham, MA  
wlucas@bentley.edu

### Abstract

*Enterprise Resource Planning (ERP) systems have received well-deserved attention from both the industry and the research communities, yet the usability of these systems remains a challenge. This paper proposes a model of the usability of ERP systems based on collaboration theory. In this model, Commitment to Joint Activity (CJA), Mutual Responsiveness (MR), and Commitment to Mutual Support (CMS) are properties of Shared Cooperative Activity between the user and the ERP system. The strength of this collaboration (conceptualized as the interaction of the individual properties: CJA, MR, and CMS) impacts the system's usability. The proposed constructs are illustrated with data from a field-based case study. Our findings provide initial support for our model of human-computer collaboration in ERP systems.*

### 1. Introduction

The widespread adoption of Enterprise Resource Planning (ERP) systems has received well-deserved attention from the research community. With an estimated market of \$38 billion [1], organizations are looking to maximize their ERP software investment. Industry reports suggest that firms are increasingly relying upon these systems to meet basic information management needs, focusing efforts on increasing the degree of integration across data sources, and enhancing business agility by moving ERP implementations to more flexible architectures [2]. However, while reliance upon ERP systems is on the rise, the usability of these systems has not improved accordingly [3-5]. This reality continues despite the potential for greater productivity and reduced training by increasing usability. A higher level of usability

can also reduce errors and improve user attitudes towards the system. Finally, usability in ERP and other large-scale enterprise level systems is increasing in importance as individuals experience more usable software interfaces in other applications (e.g., desktop and browser-based applications) [3].

This study considers these issues in the context of a collaboration perspective of human-computer interactions [6], in which a system and user collaborate or partner to accomplish their task (rather than other interaction models such as master-servant). The benefit of this collaboration view is that it encourages designers to more fully consider all types of interactions between the system and its users, including error situations, while providing an effective metaphor for designers to employ in architecting ERP systems. This notion of human-computer collaboration is motivated by calls for focusing Information Systems (IS) research attention on the IT artifact [7] and is in line with suggestions for new research rationalities in the examination of ERP use [8]. The human-computer collaboration perspective also appeals to a sociomateriality view – the notion that artifacts and the individuals using them are fully intertwined in terms of their effects on one another and should therefore be closely examined together [9]. Finally, it leverages concepts from the computer science models of collaboration [6, 24, 25].

This model of ERP system-user collaboration is based upon Bratman's [10] concept of shared cooperative activity (SCA), where he proposed the following features of such activities: commitment to joint activity (CJA), commitment to mutual support (CMS), and mutual responsiveness (MR). This conceptualization of collaboration provides the philosophical framework for the collaborative action of intelligent systems [11]. Examining ERP user-system collaboration from this perspective continues

a stream of studies examining ERP systems – evaluating ERP end user satisfaction by perception of participatory design as well as system understanding [12], evaluating the role of emotions in ERP user adoptions [13], and assessing a new method of evaluating ERP implementation strategies [14].

In this paper, we introduce a conceptual model that specifies key properties of the core construct of collaboration (shared cooperative activity) and we illustrate these properties using interview data from a case study of ERP usage.

This paper begins with a review of the theoretical background for the human-computer collaboration perspective, describing its core concepts and properties. Then, we discuss potential implications of this collaboration perspective on usability in general – and ERP system usability in particular. Next, illustrations of the core concepts and properties of our model are provided through motivating examples from our case study. The paper concludes with discussions regarding research and practical implications of this new perspective and our future research agenda.

## 2. Collaboration theory

Our research intends to develop design principles for improving the collaborative capabilities of ERP systems. By designing ERP systems that are capable of collaborating with human users, many usability problems associated with such systems can be addressed. Grosz [24] and Shieber [25] propose a paradigm shift from human-computer interaction to human-computer collaboration. Collaboration is defined as “a process in which two or more agents work together to achieve shared goals” ([6], p. 67). In system-user collaboration, the system partners with the user by being aware of the context of their interaction. They are both aware of the overall goal, and share knowledge of how the goal can be achieved through their interactions.

This new paradigm is rooted in both philosophical and computational models of collaboration [10, 26]. In these models, collaboration is conceptualized as shared cooperative activity (SCA) and is thought of as having three features:

- **Commitment to the Joint Activity (CJA):** Each party in the interaction commits to the joint activity and is aware of the context surrounding their collaboration. The parties may commit for different reasons (i.e., they may have different

ultimate goals), but they are committed to collaborating in the activity. Without intending to cooperate, the two parties might interact, but it would not be a shared cooperative activity [10].

- **Mutual Responsiveness (MR):** In a shared cooperative activity, each party responds to the intentions and actions of the other by adjusting his own behavior based on the behavior of the other party's. Without such responsiveness, the interaction might be what Bratman refers to as "prepackaged cooperation" [10], but it is not SCA.
- **Commitment to Mutual Support (CMS):** Each party is committed to supporting the efforts of the other when that party needs help and will recognize the need and provide assistance. Bratman [10] thinks of this as "If I know a partner is having trouble with performing their part and I am able to provide help in a way that does not jeopardize my individual responsibilities regarding the collective action, I must offer help."

Based on this perspective, we propose a collaboration model of system-user interaction. In this model, the three features (CJA, MR, and CMS) are seen as properties of the system-user SCA. These properties provide an approach for considering ERP use as a shared cooperative activity.

## 3. A collaboration model of ERP use

For the first property of shared cooperative activity, commitment to the joint activity, both the user and the system must in some sense know, share, and commit to the context of the collaboration and the necessary tasks to accomplish the joint activity. Both are ready to do their own part, are ready to share and communicate task-relevant knowledge, and are willing to adjust and even restart if needed. For example, SAP's Business Blueprint provides users with a diagram of the steps and processes necessary to accomplish a system task, allowing the system to share that knowledge with the user. Similarly, if the user attends training on the ERP system use, she is demonstrating commitment to the joint activity of accomplishing her tasks with the system. Other illustrations of this and the other properties will be given in the section on Empirical Illustrations.

The property of mutual responsiveness (MR) implies that both the user and the system will recognize each other's behavioral state and be willing and able to adjust their own behavior in response. Over time, in fact, mutually responsive systems and users will identify the other's behavioral patterns and adapt to these patterns. For example, an ERP system might be able to automatically complete information on some screens – responding to the previous behavior of the user. Similarly, if the user runs into an ERP system problem on one of his tasks, he should be able to respond by finding an alternative solution – even undoing previously done parts of the task.

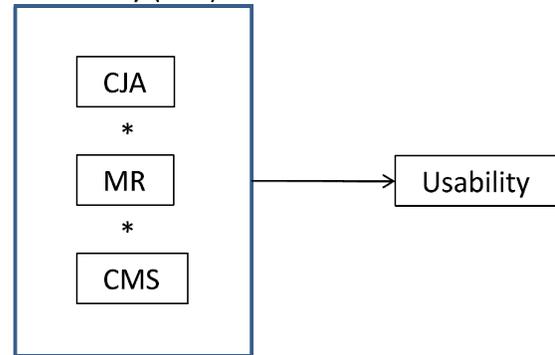
The third property – commitment to mutual support (CMS) – is most evident where the user of a system needs help to complete her planned task. In a user-system collaboration, both the user and the system should be ready to provide needed support and help to the other. An ERP system providing context-sensitive help for the user is an obvious example of this property, but another example would be a user defining his own functions that encapsulate frequent paths or processes for a task.

A distinction should perhaps be made between collaborative systems in our research and adaptive systems [28, 29]. Adaptive systems are designed to adapt to specific types users based on certain models regarding those users, such as their online purchasing behaviors. Collaborative systems are qualitatively different from adaptive systems in that they act cooperatively with the user in an environment of mutual commitment to the collaboration (CJA), are mutually responsive (MR), and include the intent and ability to mutually support the users (CMS) as indispensable components of the collaboration with respect to specific tasks.

These three properties not only determine if a joint activity is collaborative but also determine the strength of the collaborative relationship between system and user. Figure 1 presents the conceptual view of our collaboration model.

It is not immediately clear what the nature of the relationship is among the three properties of shared collaborative behavior. However, Bratman [10] clearly considers these properties as a minimal requirement for SCA. That is, all three properties must be present for the behavior to be considered a shared collaborative activity. At the same time, he maintains that each property – and SCA as a whole – comes in "degrees."

### Shared Cooperative Activity (SCA)



**Figure 1. Conceptual model of system-user collaboration.**

Conceptualizing the relationship among the properties as a multiplicative interaction implies that (1) if any of the properties has a value (or "degree," in Bratman's terms) of zero, then the total amount of collaboration – the SCA – is also zero; and (2) the amount or value of SCA will increase much more if multiple properties are high than they will if only one of them is high, for example.

## 4. ERP system usability

A variety of definitions exist for system usability. The ISO defines usability as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" [15]. Nielsen [16] identifies five attributes of usability: learnability, efficiency, memorability, errors, and satisfaction. These two definitions overlap on effectiveness (measured by metrics such as error rate), efficiency (measured by metrics such as task completion time), and satisfaction. The ISO definition emphasizes the importance of user, goal (and task), and the context of system-user interaction. The Nielsen definition provides two additional attributes, learnability and memorability, which are important factors in achieving effectiveness, efficiency, and user satisfaction. A highly usable system should also be easy to learn so that the training required can be minimal, and easy to remember so that casual users do not have to relearn how the system works every time they use it.

Usability problems of software systems can have a negative impact on user performance and

productivity. They can also lead to poor technology acceptance by users, the wasting of money and resources, and diminished return on investment in information technology [17, 18]. User-centered design has been proposed to address usability issues [19, 20]. This design philosophy requires a deep understanding and serious consideration of the characteristics, needs, and limitations of users in all stages of the software design lifecycle.

Prior research has identified a number of usability problems in ERP system use, such as difficulty in identifying and accessing the correct functionality, lack of transaction execution support, system output limitations, inadequate support in error situations, and incompatibility between the users' and the system's terminology [22]. These problems weaken several aspects of usability. The incompatible terminology problem, for example, makes ERP functions hard to learn and memorize - thus lowering the effectiveness, efficiency, and satisfaction associated with system use. As a consequence of these and other poor usability characteristics of ERP systems, users rely on both formal and informal support mechanisms (see, e.g., [23]) and external applications, such as Microsoft Excel®, to get their work done [22].

However, empirical evidence shows that the various usability problems with ERP systems can be attributed, at least partially, to the lack of collaboration between a system and its users [22]. The poor support in error situations, for example, is an example of a system's lack of mutual support (MS).

## 5. Empirical illustrations

This section provides illustrative examples to further clarify the nature of the constructs and properties introduced earlier. The quotes in these examples come from a set of fourteen interviews with users of a leading ERP system in a middle-sized software and service vendor in the northeastern U.S. The participants' organizational roles varied from an entry-level accounting job to a Vice President. The ERP implementation is less than five years old, and most of the interviewed users had experience with the same ERP system before their current employment with this organization. The interviews were semi-structured and based on a set of questions (available from the authors by request) intended to explore the users' perceptions regarding the level and type of

collaboration in their interactions with the ERP system.

The section is organized by the three properties of collaboration (i.e., shared collaborative activity): commitment to the joint activity, mutual responsiveness, and commitment to mutual support. We drew the examples of each property to illustrate system and user behavior that contributed either positively or negatively.

### 5.1 Commitment to the Joint Activity

Commitment to the Joint Activity (CJA) is illustrated first. From the users' perspective, there were many examples of ways that the user and system might demonstrate knowledge, sharing, and/or commitment to the necessary tasks for accomplishing the joint activity (in this case, the business task/process for which the system is being used). For example, not surprisingly, most users were confident that they knew how to complete the tasks that they perform often – and how to use the system as a tool in doing so.

*User 1: What it means very simply is, think about conceptually what you're trying to do before you even touch the keys.*

*User 1: How do you know that this is your debit piece and this is your credit piece? You wouldn't know that until you walked through it all and then you understood conceptually. It worked on that screen and then you looked at the screen and said, 'Okay. Yeah. This is what I was trying to do.'*

Training was often cited as critical in developing this understanding of the task processes on the part of the users.

*User 2: They [other users] memorized it. They've memorized what needs to happen too. So, that's the one drawback I would say versus some other systems that's not very intuitive. So, you need to know the screens. And, you need to be trained in what the process is.*

Characterizing commitment and sharing of an ERP system was understandably more difficult for users to describe, as the users normally had not thought about their systems as collaborators or partners. However, they were still able to come up

with ways that an ERP might share its knowledge about the task at hand.

*User 1: Again, back to a search of a database that [captures] common tasks, and says, 'If you're trying to do this, here's the transaction code you'd use.' And it would walk you through screen (trials) or something.*

Users acknowledged at times that there were responsibilities that simply were theirs and that we should not expect the system to take their role:

*User 1: No. No. I think it's just (up to) you to understand what you're trying to do and say 'Oh, yeah. Your (journal actions are going to have to be) balanced, of course.'*

Similarly, there were a number of user statements regarding the system's intention (or mostly the lack of it) to share the knowledge and relevant details of collaboration. However, in some cases, the users were pleased about the system's capabilities related to sharing, illustrated by these quotes:

*User 3: Yes, in looking up the history – I mean it allows you to see all of the transactions. And so, you can drill in here and see why you short-paid someone, or why you paid the amount you did. It makes sense. It's like a story when you go through them.*

*User 4: I think that the system is actually very helpful. This little help button right here, which will launch into either the definitions for where you are, if you're on a particular field, but then it will also launch into help with the overall applications, so I actually think it's very helpful.*

These positive statements regarding the ERP system's intent or ability to share with the user were often related to the use of terminology consistent with the user's domain jargon, simple process guidance, error messages, drill-down capabilities, consistency between versions, and the help function. There were, however, many negative statements about the system's unwillingness/inability to share either important knowledge or essential collaboration details. Many of the points of frustration were relatively simple, such as lack of clarity in identifying required fields, late or ambiguous error messages, unclear terminology, unclear visual symbols, and

lack of navigational clarity. Sometimes users were simply confused about where they were in the system and what functions they were performing, as in this quote from an expert user with over 10 years experience with the ERP system:

*User 4: If I executed the transaction, I think it would come back and tell me that this one – actually, now I look at it, I'm probably in a bad transaction to do it on. If we went in here – Customer accounts. Depending on what I'm going to try to do here. I forget if I'm in account payable or accounts receivable here.*

*INT: How can you tell if you need to know?*

*User 4: I actually don't know. I'd probably just go back and then I'd look at the menu path that I was in. So here I'm in accounts receivable.*

Many user comments related to the system's inability to share either transaction information or process details in an intuitive way that made sense to them:

*User 1: When you first see this layout stuff, it doesn't make sense. But then once you know what it is it does make sense.*

*User 3: Honestly, for me personally, every time I've gone and tried to drill in through the folders, I can never get to where I think I need to be.*

## 5.2 Mutual Responsiveness

For mutual responsiveness (MR), the users consistently mentioned a specific ERP behavior that changed based on the users' prior action – the system allowing the user to reuse previously entered data very easily. This is a relatively simple form of behavioral adjustment, and the users discussed additional ways in which the system could be responsive in a beneficial way. These examples included broader types of system process changes based on repetitive user actions, allowing better options in situations when the user has to cancel an operation, removing fields that the user never uses, preventing accidental repetition of a transaction, and having more intelligent default values.

On the user side, some users felt that they were constantly adjusting their behavior because of the system's actions:

*User 1: If you're used to doing an upload of the same thing, you might have to manually type that*

*in, if the upload doesn't work. You always have to adapt to the system.*

At least for some users, their adaptation to the system's behavior had gone so far that they no longer even considered alternative ways of performing their task:

*INT: Do you find yourself having to change the way you do things to match the way the system wants you to do it?*

*User 1: It's hard to say because I think I've just always done it according to the system.*

*INT: Are there any examples of situations where the system forces you to do things in a certain way that is different from your preferred way of doing it, accomplishing a certain goal?*

*User 6: Geez, I don't know. I think after all these years I think the system has just beaten me down.*

*INT: Is it that way or the other way?*

*User 6: Yes, I think it's beaten me down. I just, I had to capitulate.*

Users even adapted to the terminology used by the system:

*User 7: We get in meetings that are comical. They're comical when people are like 'I was working on [XXXX], but you didn't [YYYY] first' – it's a whole different language, and our guys can do that.*

Many of the examples from the interviews were about system capabilities, both positive and negative. The positive ones related to the system's ability to identify simple repetitions in user actions and act accordingly, sufficiency of most error messages, and changes in system behavior based on access rights. Error messages were, however, also a significant source of negative comments, such as the following:

*User 4: The system just doesn't know what the problem is. It doesn't know how to diagnose those. So this one says, no suitable documents found. Message [XXXXXX].*

Another common negative user perception was that the system did not support default values consistently across all functions. Moreover, users commented on the system's inability to adapt to the specific task for the interaction. When answering a question regarding how the ERP system could be

improved to be more responsive, a user commented that:

*User 8: I would say just getting rid of some of the fields that you don't need to look at [for this task] instead of having to go through them all. It's just a waste of time.*

### 5.3 Commitment to Mutual Support

For Commitment to Mutual Support (CMS), there were a number of examples of users perceiving the system to be helpful when they ran into trouble in completing their tasks. Most of these situations related to the guidance that error messages provided, such as:

*User 5: Yes, it can give you error signs, saying something is not right, or this field is incorrect, or this has already been inputted.*

However, users also criticized the ERP system's inability to help resolve (and not just identify) problems. For example:

*User 2: JK: No, it doesn't give you like error messages that say hey, you need to do x, y, z.*

In several of these cases, the users' natural reaction was to seek help from either an immediate colleague or a super-user. Personal interaction with other users appears to be the most typical source of help in situations when the system fails to provide assistance.

In one situation where the user got lost and did not know how to move forward with a task, the system did not recognize the need for help and provided no response or assistance, laying the burden of seeking solutions completely on the user:

*User 3: The thing is that in this case [of getting lost], the system is not reaching out to you saying that you obviously need help. It's me having to go find it.*

## 6. Discussion and conclusions

The contribution of this paper is twofold:

(1) theoretically, building on our earlier work [32], we propose and further elaborate a conceptual model of collaboration, with the properties of shared

cooperative activity (SCA) being described as commitment to a joint activity (CJA), mutual responsiveness (MR), and commitment to mutual support (CMS), and explain why such collaboration should improve the usability of ERP systems.

(2) practically, we provide examples demonstrating that ERP systems currently lack many collaboration capabilities and suggest that principles of collaboration should be considered during system design to improve the usability of enterprise systems.

In this paper, we examine user-system interactions in ERP systems by building on a theoretical approach of collaboration theory [10, 26] taken from the fields of philosophy and computer science. Through the use of empirical illustrations from a case study based on field interviews of ERP users in a variety of organizational roles within the same company, we demonstrated the feasibility of our theoretical model as a framework for evaluating the usability characteristics of a complex enterprise system.

The empirical illustrations demonstrate that instances of all these properties are present in the interview data, mostly in statements that the users made regarding the system. These statements suggest that the ERP system in the studied organization was not designed to strongly collaborate with its users and that, in practice, it violated many collaborative principles.

While our study provides some evidence that the collaboration capabilities as well as the usability of ERP systems is inadequate, the main contribution of this study is not in the aggregated empirical findings but in the conceptual model that forms a foundation for a more sophisticated understanding of the relationships between the core collaboration properties and their specific linkages to system usability.

It is important to acknowledge that our research approach currently provides only a limited window to the key properties from the system's perspective. The interviews generally describe users' perceptions regarding the system and its design. A thorough evaluation of system design documents, interviews with designers, and/or a structured, formal walkthrough of the system would certainly provide additional perspectives. However, the user perceptions are valuable to study themselves, as it is these perceptions of the system's behavior and design that will have the most direct impact on the users' ability, motivation, and willingness to act collaboratively in their interactions with the ERP

system, and, in some cases, to even use the system at all.

In addition, we found evidence that user perception of the ERP system is based upon a broad definition of *information system* that includes the vendor-default information and communication artifact, the customized artifact, as well as the technical and operations staff supporting the artifact. As a result, any future perception-based research will need to clearly specify the boundaries of the system that respondents have in view.

Furthermore, future studies should explore the potential of this model to link to other theories regarding information system use such as UTAUT [18]. For example, it will be important to understand whether and how a user's perception of a system's intentions (CJA and CMS) might lead to the user's behavior of accepting and using the system. A more systematic and comprehensive analysis of a larger body of data is required to demonstrate internal and external validity of our framework.

The concept of usability is complex and requires evaluation through multiple means and from multiple perspectives [31]. Empirical evidence suggests that various usability problems with ERP systems can, in many situations, be attributed to the lack of collaboration between system and users [22], and many usability issues could likely be addressed if the systems were designed following principles that are compatible with human-computer collaboration. In this model, CJA is a critical element as it implicitly creates a mutually understood division of labor or responsibility between the system and user [11], so that both the user and the system are able to "commit" to do their own part. Designing a system as a collaborator means that more responsibility is shifted to the system so that less cognitive load will be required for the user to learn or memorize how to use the system to accomplish the business tasks effectively and efficiently.

The collaborative perspective provides a useful framework for identifying technology characteristics of complex enterprise systems that influence (either positively or negatively) the strength of the collaborative relationship between the system and its human user. However, this case study is only the first step in articulating the relationships between these constructs. At the current stage, it is still unclear exactly how the properties of shared cooperative activity in the model will affect the various attributes of usability and how other factors (e.g., user's computer experience, user's prior experience with

similar systems, and task complexity) might moderate these effects.

The field study described in this paper has strengthened the theoretical model which, in turn, forms a foundation for a number of other research approaches that are part of the broader research project. In the future, this project will examine a range of organizations through cross-case analysis, conduct surveys of ERP collaborativeness perception, implement a cognitive walkthrough protocol incorporating collaborative concepts, and ultimately identify a set of design principles to improve ERP usability. The development of a proof-of-concept prototype design using a collaborative design approach is an integral part of the research project.

Achieving a high level of usability for large-scale enterprise systems is critically important for both the organizations building the core of their business around these systems and for the employees who often spend much of their work time using these systems. Our hope is that the research reported in this paper and the entire research program will ultimately build a foundation on which enterprise systems with better usability characteristics can be designed.

## 7. Acknowledgements

This material is based in part upon work supported by the National Science Foundation under Grant No. 0819333. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## 8. References

- Hamerman, P.D., "ERP Applications 2008: The Battle Goes Vertical." 2008, Forrester Research.
- Woods, J., "Modernizing ERP: How to Make Users Fall in Love With ERP All Over Again." 2008, Gartner.
- Hestermann, C., "Key Issues for Enterprise Resource Planning 2009". 2009, Gartner.
- Hestermann, C. and J. Woods, "Key Issues for Enterprise Resource Planning, 2008." 2008, Gartner.
- Otter, T., "Case Study: Ness Combines Consumer Application Ease of Use With ERP Robustness." 2008, Gartner.
- Terveen, L., "An overview of human-computer collaboration," *Knowledge-Based Systems*, 1995. 8(2-3): pp. 67-81.
- Orlikowski, W.J. and C.S. Iacono, "Research Commentary: Desperately Seeking the 'IT' in IT Research--A Call to Theorizing the IT Artifact," *Information Systems Research*, 2001. 12(2): pp. 121.
- Lee, A., "Researchable directions for ERP and other new information technologies," *MIS Quarterly*, 2000. 24(1).
- Orlikowski, W.J., "Sociomaterial Practices: Exploring Technology at Work," *Organization Studies*, 2007. 28(9): pp. 1435-1448.
- Bratman, M.E., "Shared Cooperative Activity," *The Philosophical Review*, 1992. 101(2): pp. 327-341.
- Grosz, B.J., "Beyond Mice and Menus," *Proceedings of the American Philosophical Society*, 2005. 149(4): pp. 529-523.
- Wu, J.-H., et al., "An Examination of ERP User Satisfaction in Taiwan," in *Hawaii International Conference on System Sciences*. 2002: Hawaii.
- Nelson, K., "Exploring Emotions during ERP Adoption: A Stakeholder Analysis," in *Hawaii International Conference on System Sciences*. 2005: Hawaii.
- Guido, C. and R. Pierluigi, "A methodological proposal to assess the feasibility of ERP Systems Implementation Strategies," in *Hawaii International Conference on System Sciences*. 2008: Hawaii.
- ISO, "Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 11: Guidance on usability." 1998.
- Nielsen, J., *Usability engineering*. 1993, Boston, MA: Morgan Kaufman.
- Davis, F.D., "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology," *MIS Quarterly*, 1989. 13(3): pp. 319-340.
- Venkatesh, V., et al., "User acceptance of information technology: Toward a unified view," *MIS Quarterly*, 2003. 27(3): pp. 425-478.
- Norman, D., *The design of everyday things*. 1988, New York, NY: Doubleday.
- Maguire, M., "Methods to support human-centred design," *International Journal of Human-Computer Studies*, 2001. 55(4): pp. 587-634.
- Lazar, J., A. Jones, and B. Shneiderman, "Workplace User Frustration with Computers: An Exploratory Investigation of the Causes and Severity," *Behaviour and Information Technology*, 2006. 25(3): pp. 239-251.
- Topi, H., T. Babaian, and W. Lucas, "Identifying usability issues with an ERP implementation," in *Seventh International Conference on Enterprise Information Systems*. 2005: Miami, FL.
- Topi, H., W. Lucas, and T. Babaian, "Using informal notes for sharing corporate technology know-how," *European Journal of Information Systems*, 2006. 15(5): pp. 489-499.
- Grosz, B.J., "Collaborative Systems: AAAI Presidential Address," *AI Magazine*, 1996. 17(2): pp. 67-85.
- Shieber, S.M., "A call for collaborative interfaces," *ACM Computing Surveys*, 1996. 28(A(electronic)): pp. 143.

26. Grosz, B. and S. Kraus, "Collaborative Plans for Complex Group Action," *Artificial Intelligence*, 1996. 86(2): pp. 269-357.
27. Cohen, P.R., H.J. Levesque, and J.H.T. Nunes. "On acting together." in *Proceedings of the 8th National Conference on Artificial Intelligence (AAAI-90)*. 1990. Menlo Park, CA.
28. Jameson, A., "Adaptive interfaces and agents," in *The Human-Computer Interaction Handbook*, A.S.a.J.A. Jacko, Editor. 2008, CRC Press: New York, NY. p. 433-458.
29. Benyon, D., "Adaptive systems: A solution to usability problems," *User Modeling and User-Adapted Interaction*, 1993. 3(1): pp. 65-87.
30. Calisir, F. and F. Calisir, "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*, 2004. 20(4): pp. 505-515.
31. Agarwal, R. and V. Venkatesh, "Assessing a firm's Web presence: A heuristic evaluation procedure for the measurement of usability," *Information Systems Research*, 2002. 13(2): pp. 168.
32. Babaian, T., W. Lucas, and H. Topi (2006) "Improving ERP Usability Through User-System Collaboration," *International Journal of Enterprise Information Systems*, 2006. 2(3): pp. 10-23.