Communications of the Association for Information Systems

Volume 41 Article 31

12-2017

A Conceptual Model for User-system Collaboration: Enhancing Usability of Complex Information Systems

Jennifer J. Xu
Bentley University, jxu@bentley.edu

Heikki Topi
Bentley University

Communications of the Association for Information Systems

Research Paper ISSN: 1529-3181

A Conceptual Model for User-system Collaboration: Enhancing Usability of Complex Information Systems

Jennifer J. Xu
Computer Information Systems
Bentley University
jxu@bentley.edu

Heikki Topi
Computer Information Systems
Bentley University
htopi@bentley.edu

Abstract:

A variety of organizations around the world use complex information systems (e.g., enterprise resource planning and supply chain management systems). However, poor usability caused by system complexity continues to frustrate users and damage the reputation of these systems. In this study, we address usability issues with complex information systems from the human-computer collaboration perspective by modeling user-system interaction as a joint activity between the system and its users. We propose a conceptual model for user-system collaboration, elaborate on the components in the model and the relationships between the components, derive the required capabilities for collaborative information systems, and establish conceptual relationships between system collaborative behaviors and usability. We use empirical evidence gathered from a qualitative field study on ERP systems to illustrate the model and the possible impact of system collaborativeness (i.e., the presence or absence of collaborative capabilities) on usability. We do so to provide a strong conceptual foundation for modeling user-system collaboration and to encourage designers to employ the collaboration metaphor during system design and, thus, help them develop future complex information systems with better usability.

Keywords: Complex Information Systems, Usability, User-system Collaboration, Conceptual Model, Enterprise Resource Planning Systems.

This manuscript underwent peer review. It was received 08/29/2016 and was with the authors for 10 months for 2 revisions. The Associate Editor chose to remain anonymous.

DOI: 10.17705/1CAIS.04131

1 Introduction

As their operating environment continues to change more rapidly than ever, organizations are increasingly relying on advanced digital technologies to adapt and gain competitive advantage. To effectively reduce costs; further enhance efficiency and productivity; and leverage their informational, material, and human resources fully, many organizations have adopted a diverse set of complex information systems to enable their daily operations. Hsieh and Wang (2007, p. 216) define complex information systems as "large organizational information systems that integrate and streamline business processes across various functional departments/areas". Enterprise resource planning (ERP) systems are an example of large-scale, enterprise-wide complex information systems that play a key role in organizational life (Davenport, 1998; Markus, Axline, Petrie, & Tanis, 2000). Examples of complex information systems can also be found in aerospace and marine operations, power production, and healthcare domains (Pascot, Bouslama, & Mellouli, 2011; Rouse, Geddes, & Curry, 1987-1988; Tretten & Karim, 2014). The complexity of these systems not only requires sophisticated engineering of the system architecture, software, hardware, data and processes but also leads to significant obstacles for modeling and designing highly usable systems (Albers, 2011). At the core of the problem is the way in which the systems interact with their users. In this paper, we address the usability design challenges facing complex information systems with a special focus on ERP systems.

During the past two decades, ERP technologies have advanced drastically (Marston, Li, Bandyopadhyay, Zhang, & Ghalasi, 2011; Pairat & Jungthirapanich, 2005). With an integrated data repository and standard front-end user interfaces, an ERP system can facilitate a broad spectrum of business functions and serve units ranging from purchasing and manufacturing to sales and human resource management (Al-Mashari, Al-Mudimigh, & Zairi, 2003; Davenport, 1998; Grant, Hwang, & Tu, 2013). Many organizations continue to invest substantial amounts of financial and human resources in ERP system acquisition, implementation, deployment, maintenance, and consulting services, making the ERP market one of the fastest-growing software markets. The advent of cloud computing further expands the ERP market. Sources have estimated that the global ERP sales will reach US$41.7 billion by 2020 (Chaudhari & Ghone, 2015). Despite the vast technological advances and functional and performance improvements, poor usability continues to negatively affect ERP systems' image and negatively affect these systems' users (Bueno & Salmeron, 2008; Hestermann, 2009; Lambeck, Fohrholz, Leyh, Šūpulniece, & Muller, 2014a; Matthews, 2016; Parks, 2012). Frequently reported usability problems include complex and unintuitive user interfaces (Parks, 2012; Singh & Wesson, 2009), lack of task support and guidance (Calisir & Calisir, 2004; Topi, Lucas, & Babaian, 2005), and poorly designed user-training manuals (Scott, 2005; Scott & Vessey, 2002).

Usability has long been a key research topic in the field of human-computer interaction (HCI). In this research, we posit that using the human-computer collaboration perspective to guide the design of complex information systems will result in improved usability. This perspective maintains that computer systems, even though they do not possess independent intentions, can be designed to serve as knowledgeable collaborators of human users. If a system, especially a complex one such as an ERP system, is designed to take a more active role than it currently does and to help overcome human limitations and enhance human abilities (Rouse et al., 1987-1988), users will be less burdened with the high cognitive load associated with learning and operating the system. Consequently, users will be better able to focus on their core goals instead of the instrumental process of interacting with the system. In turn, this focus will improve the users' perceptions of system usability. From a practical standpoint, this perspective encourages designers to more fully consider all types of interactions between systems and users. Applying the human-computer collaboration perspective to the design of user-system interactions realigns the roles of the user and the system by placing a greater emphasis on the system's responsibility for helping users accomplish their goals while ensuring that they remain in control. In the ERP context, this perspective provides an effective metaphor for designers to employ when architecting ERP systems.

To achieve our broad research objective—that is, to better understand and enhance user-system collaboration while designing complex information systems—we address three key research questions in this paper:

RQ1: How can we model user-system interaction as a collaborative activity?

RQ2: What capabilities should complex information systems maintain to successfully collaborate with users?

RQ3: Through which mechanisms could collaborative capabilities have an impact on the usability of complex information systems?

To address the first research question, we propose a conceptual model based on and derived from the collaboration theory (Bratman, 1992). Our model includes the required features for collaborative activity specified in the original theory. Moreover, the model identifies and incorporates additional intention and behavioral components and relationships, which are necessary for modeling user-system interaction as a collaborative activity. Based on this conceptual model, we address the second research question by proposing seven behavioral capabilities that complex information systems ought to possess in order to work as knowledgeable partners of users. We examine the third research question by relating collaborative capabilities with system usability. We use empirical examples that we gathered in a qualitative field study on ERP systems to illustrate the model's components and how the level of system collaborativeness (i.e., the presence or absence of collaborative capabilities) can influence users' perceived system usability.

We stress that, since the development of this model is only the first step toward our broader research objective, this paper is to a large extent conceptual. Specifically, we do not develop a grounded theory because we mainly derived and extended our model from prior theory about collaboration, which is aligned with the positivist view. On the other hand, however, we do not intend the empirical examples we cite to "test" the model quantitatively, which certainly is part of our plan for the future stages of the research. Positioned in the theory generalizability framework (Lee & Baskerville, 2003), our model can be seen as a type "TT" theory, which:

> *generalizes from theoretical propositions in the form of concepts (such as variable, an priori construct, or another concept) to the theoretical propositions that make up a theory (specifically, a set of logically consistent propositions that, pending the results of empirical testing, could qualify as a theory)* (p. 238).

This paper proceeds as follows. In Section 2, we review the literature on usability and the human-computer collaboration perspective. In Section 3, we propose our conceptual model for user-system collaboration, the seven collaborative capabilities for complex information systems, and the possible mechanisms through which collaborative capabilities may impact system usability. In Section 4, we use examples from the qualitative field study to illustrate this model. We discuss the connections of our model with existing IS research on system characteristics and enterprise systems. In Section 5, we discuss our study's implications for research and practice. Finally, in Section 6, we conclude the paper and suggest future research directions.

2 Literature Review

2.1 Usability of Complex Information Systems

Usability is a key research theme in the field of HCI (Nielsen, 1993; Sears & Jacko, 2008), and researchers have recognized it as an important system quality dimension (DeLone & McLean, 2003) and target of study (Agarwal & Venkatesh, 2002; Venkatesh & Agarwal, 2006). In general, usability research focuses on developing principles, techniques, and methods to enable the design of highly "usable" interactive computer systems (Agarwal & Venkatesh, 2002). The International Organization for Standardization has defined 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" (ISO, 1998, p. 2). The HCI field has widely accepted this definition (Hornbæk, 2006). The definition focuses on the "quality of use" (Bevan, 1995) and requires that one evaluate a system with respect to the specific context of use. In other words, the three dimensions of usability (i.e., effectiveness, efficiency, and user satisfaction) are meaningful only when considering who uses the system (user), in what situation they do so (context), and for what purposes (goal). The characteristics of the users (e.g., education, knowledge, experience, and job role), the tasks[1] (e.g., goals, frequency, duration, and complexity), and the technical, organizational, and social environments all have an impact on the system's usability (Maguire, 2001). There are a variety of other measures for usability such as learnability and memorability (Nielsen, 1993),

[1] The IS literature broadly defines tasks as "actions carried out by individuals to turning inputs into outputs" (Goodhue & Thompson, 1995). We use the same definition in this paper.

which one can also categorize as one of the three dimensions; namely, effectiveness, efficiency, or satisfaction (Hornbæk, 2006).

Although HCI research has studied usability for many years (Sears & Jacko, 2008), little research has addressed usability design in the context of large-scale, complex information systems. The complexity of an information system may manifest itself in many aspects: the multitude of functions and modules, the numerous possible navigational paths, the highly integrated multidimensional data, the closely coupled and dependent tasks and processes, the sheer number of users with varying roles and skills, and the dynamic environments and contexts in which the systems operate.

When one considers all these factors and, more importantly, their interconnections, designing a highly usable system becomes rather challenging. Albers and Still (2011) summarize that the design challenges result from four types of complexity: complex work environment (requiring collaboration among different people with a goal sustained over time), complex information context (dynamically changing data and information space), complex technology (a large number of applications and systems), and complex topic (requiring advanced technical or domain knowledge). Norman (1988) offers three general principles for designing complex systems: use common knowledge and do not require unique knowledge, use functions to help the user make natural decisions, and make options readily visible. Tretten and Karim (2014) make more specific recommendations regarding input, output, workflow design, and other system characteristics. Albers (2011) points out that the usability of a complex information system is more than the sum of the usability of its individual components. Some researchers have proposed architectural solutions to enable complex information systems to interact with users intelligently (Pascot et al., 2011; Rouse et al., 1987-1988). These approaches share a common assumption that high level of usability needs to be achieved beyond user interfaces and by restructuring various other related system components such as information and data management, error handling, adaptive aiding, and context monitoring. An important contribution of our research is that we use a holistic approach to address system usability and avoid the simplistic view of user-system interaction design as a process according to a check list for discrete user interface features, which many existing usability design heuristics or guidelines have achieved (Nielsen, 1993; Norman, 1983).

In the domain of ERP systems, a limited number of studies have investigated usability. Topi et al. (2005) conducted a field study and identified a number of usability issues that ERP users experience, including poor navigability, lack of task support and guidance, and complexity. Calisir and Calisir (2004) found a significant impact of user interface characteristics (system capability, user guidance, and learnability) on user satisfaction. Some recent studies have re-evaluated a variety of ERP systems and found that complexity, difficulty in finding functionality, and poor error handling remain major usability issues (Lambeck et al., 2014a; Lambeck, Muller, Fohrholz, & Leyh, 2014b). Singh and Wesson (2009) developed a set of criteria for evaluating ERP system usability in terms of the system’s navigation, presentation, task support, learnability, and customization characteristics. Through the lens of sociomateriality, O’Farrell, Letch, and Purchase (2012) identified several ERP usability challenges that relate to users’ resistance and accommodation behaviors when using ERP systems (see also Topi, Lucas, & Babaian, 2006).

ERP systems—a prototypical example of a complex information system—have been studied extensively in the information systems (IS) literature. A significant body of research examines the critical factors that cause the success (or failure) of ERP implementation at the organizational level (Botta-Genoulaz, Millet, & Grabot, 2005; Grossman & Walsh, 2004; Sarker & Less, 2003; Siau, 2004; Snider, da Silveira, & Balakrishnan, 2009). An IS research stream related to usability studies the acceptance and adoption of ERP systems in organizations. Unlike usability research in HCI that takes an engineering approach and examines the system’s design characteristics that users experience, this line of research focuses on the psychological processes that influence a user’s behavioral intention to adopt and use a system (Palmer, 2002; Venkatesh & Agarwal, 2006). For example, Bueno and Salmeron (2008) developed an ERP success model based on the technology acceptance model (Davis, 1989). They found that critical factors such as cooperation, user training, and technological complexity may influence users’ perceptions regarding an ERP system’s usefulness and ease of use, which further influence users’ attitudes toward use and behavioral intentions to use the system. Similarly, Amoako-Gyampah (2007) found that, in addition to perceived usefulness and ease of use, a user’s level of intrinsic involvement can also affect the user’s intention to use ERP systems. Overall, while the ERP implementation research has produced important results, it does not directly inform this work given our different focus on individual ERP user’s relationship with the system.

Our research differs from the acceptance and adoption literature because we focus not on users' psychological processes but on systems' usability, an important system design characteristic. A recent comprehensive review of the critical success factors of ERP adoption/implementation research (Shaul & Tauber, 2013) suggests that prior research has paid little attention to the design of ERP systems. Research on design related to the user-system interaction is even scarcer. We examine the design characteristics related to usability from the human-computer collaboration perspective by modeling user-system interaction as a collaborative activity.

2.2 Human-Computer Collaboration

The human-computer collaboration perspective is rooted in both the HCI and artificial intelligence (AI) fields (Grosz, 2005; Grosz & Kraus, 1996; Rich, Sidner, & Lesh, 2001; Terveen, 1995). In the context of this perspective, system use is treated as a collaborative activity performed jointly by a human user and a computer system. Collaborative activity refers to "a process in which two or more agents work together to achieve shared goals" (Terveen, 1995, p. 67). In human-computer collaboration, the system no longer works only as a passive servant that reacts to the commands and instructions issued by its human user. Instead, the system takes an active role and collaborates with its human user (Grosz, 1996; Shieber, 1996). In other words, the relationship between a computer system and its user becomes a partner-partner relationship (Oberquelle, 1984; Oberquelle, Kupka, & Maass, 1983) rather than a master-servant one, which is the dominant underlying model of human-computer interactions for most existing systems.

The human-complementary approach (Terveen, 1995) employed in the human-computer collaboration perspective recognizes that humans and computers have different strengths. Successful collaboration can be achieved by allocating different responsibilities to humans and computers based on their unique capabilities and strengths. This approach does not require a computer system to have a free will and a high level of (artificial) intelligence in order to be able to interpret and act on the intentions, beliefs, and goals of its human users. Instead, the system's collaborative capabilities can be achieved by a design guided by the principles of collaboration. These principles specify the appropriate division of labor between humans and computers. The human-complementary approach is a key tenet underlying this research.

Prior research in philosophy and computer science has proposed several important theoretical models of collaboration. Philosopher Bratman (1992) proposed a seminal theory on collaboration. In this paper Bratman refers to a collaborative activity as a shared cooperative activity (SCA) and identifies three critical features of collaborative activity (or an SCA): commitment to the joint activity (CJA), mutual responsiveness (MR), and commitment to mutual support (CMS). Bratman's theory and his later related work (Bratman, 1993, 1997, 1999) have influenced research in a variety of fields including psychology (Alonso, 2009; Knoblich, Butterfill, & Sebanz, 2011), sociology (Pellizzoni, 2001; Tomasello, 2006), computer science (Jennings, 1996; Panzarasa, Jennings, & Norman, 2002), and computational linguistics (Lochbaum, 1998).

In the field of computer science, researchers have proposed more specific models of human-computer collaboration in an attempt to develop collaborative computer systems. For example, some have proposed a mathematical model called SharedPlan to formulate the planning process in collaboration in the form of logic (Grosz & Kraus, 1996; Grosz & Sidner, 1990; Kamar, Gal, & Grosz, 2009). Grosz (1996) modeled collaborative activity as having three features: 1) commitment to the joint activity, 2) reaching consensus on the recipe, and 3) commitment to constituent actions. Terveen (1995) summarized prior models of human-computer collaboration and consolidated a list of five features of human-computer collaboration: 1) agreement on the shared goal(s) to be achieved; 2) planning, allocation of responsibility, and coordination; 3) shared context; 4) communication; and 5) adaptation and learning.

Although researchers have proposed the collaboration theory and several models related to it, none have used the human-computer collaborative perspective to address the usability problem of large-scale, complex information systems. We believe that this perspective provides a new and improved theoretical lens for examining usability of complex information systems by offering a comprehensive approach to improving the system design as a whole. It is particularly useful for ERP system design because the usability challenges imposed by the intrinsic complexity of ERP systems cannot be successfully addressed merely by scattered patchwork on the system interfaces. This approach requires a fundamental change in the user-system relationship from a master-servant type to a partner-partner one. In Section 3, we present our conceptual model for user-system collaboration, which we derived from Bratman's (1992) collaboration theory. We chose this theory as our theoretical framework because it has been successfully used as the foundation for many existing human-computer collaboration models as we review above.

3 The Conceptual Model for User-system Collaboration

The traditional approach to system design assumes that users are the only actors with active responsibilities and that they must maintain full knowledge about the process of completing a task. We envision a collaborative information system as one that would function more like a helpful and knowledgeable coworker, which would reduce the user's cognitive load associated with operating the system and allow the user to concentrate on her core task-related goals. In this section, we first elaborate on the original formulation of shared cooperative activity (Bratman, 1992) in detail. Based on Bratman's theory, we then present the conceptual model with several key aspects that extend the original theory and propose the capabilities that complex information systems ought to have to enable successful user-system collaboration and to improve system usability. We chose Bratman's philosophical theory of collaboration as the conceptual foundation of our work because researchers have used it, and Bratman's work in general, as a foundation for understanding concepts of collaboration, planning, and intention in the context of shared activity between human systems and computing-based systems (see Section 3.2 below). As we describe above in Section 2.2, researchers across a variety of fields have widely used Bratman's work, and it has been recognized with formal awards, such as an International Foundation for Autonomous Agents and Multi-Agent Systems Award in 2008 for Bratman, Israel, and Pollack (1988) and in 2007 for Grosz and Kraus (1996) who heavily use Bratman (1992). Bratman is not, of course, the only philosopher who has worked in this field, but, as we discuss in Section 3.2, his work has been highly influential in understanding human-computer collaboration.

3.1 The Collaboration Theory

As we discuss above, Bratman's (1992) collaboration theory provides a theoretical framework for modeling collaborative activity. In this framework, a collaborative activity (or shared cooperative activity in Bratman's terminology) refers to a joint activity with three required features: commitment to joint activity (CJA), mutual responsiveness (MR), and commitment to mutual support (CMS). CJA and CMS are intentions, and MR includes both intention and behavior. The first feature, CJA, represents the intentions that the participating agents maintain concerning the collaboration; that is, each agent intends to act collaboratively with others in the joint activity. Bratman further specifies that the subplans of the agents for achieving the success of the joint activity have to mesh to form an interlocking and reflexive set of intentions. Bratman's specification of CJA implies that a collaborative activity is possible only if three important components and conditions are in place:

- There is a *shared goal*. The definition of collaboration (Terveen, 1995) indicates that collaborative activity is goal oriented. Although the personal goals of the agents in a collaborative activity may not be the same, the agents must agree on a shared goal toward the success of the joint activity. The shared goal also becomes the criterion base on which the outcome of the activity is assessed. Note that the shared goal may not be completely and unambiguously defined before the collaborative activity begins. The goal is subject to refinement and reformulation as the activity progresses.
- There are *shared plans*. In order to achieve the shared goal, the participating agents must decide how to carry out the activity. Grosz (1996) refers to this component as the "consensus on recipe". The consensus has to be reached through a negotiation process in which the agents get to know each other's subplans, reconcile different ideas, and eventually agree on a set of shared, meshing subplans. An important outcome of reaching the shared plans is the *division of responsibility*. This responsibility allocation results from the negotiation process for working out the shared plans during which the assignment of constituent actions to each agent is agreed on (Grosz, 1996). This assignment specifies who is going to do what based on each agent's strengths, abilities, and constraints.
- There is *knowledge*[2] concerning the collaborative activity. Critical knowledge includes the agents' intentions, the shared goal, and the meshing subplans (Bratman, 1992). Additional

[2] The knowledge maintained by collaborative partners in general can be both tacit and explicit knowledge (Nonaka & Takeuchi, 1995). In the user-system interaction context, however, the knowledge (e.g., business process definition, organizational terminology, task progress and status, etc.) is mostly explicit and can be encoded and stored in the system. Even tacit knowledge, such as the

information such as the activity status and progress also becomes important contextual knowledge that the agents have to keep track of during the course of the activity (Silverman, 1992; Terveen, 1995). Thus, each agent is responsible not only for executing their own subplans and doing their own job but also for maintaining, communicating, and sharing collaboration-relevant contextual knowledge.

Because a joint collaborative activity requires these components, CJA is not an atomic construct but comprises several intention components, which we can express from an agent's perspective as “I intend (a) to achieve the shared goal, (b) to execute the shared plans based on my assigned responsibilities, and (c) to share knowledge”.

The second collaborative feature in Bratman's (1992) theory comprises both the intentions and behaviors of mutual responsiveness (MR). In Bratman's words (p. 328): “Each [agent] seeks to guide [their] behavior with an eye to the behavior of the other, knowing that the other seeks to do likewise”. Agents can only do so if they are aware of each other's behavior (and intentions) and change their behavior based on what they learn about the others. The MR intentions will lead to responsive behaviors. Bratman strongly emphasizes the importance of mutual responsiveness both in intention **and** in action (p. 339). Mutually responsive actions require mutual responsiveness in intention, which is reflected in agents’ willingness to *stay aware* of the state of the other agents and *adjust their own behaviors* based on what they learn about the others.

The third feature, commitment to mutual support (CMS), represents one's intention to offer assistance when any of the participating agents encounter problems. This commitment ensures that each agent not only performs their own responsibilities but also is willing and ready to provide help and support in order to optimize the likelihood of success of the collaborative activity.

When stated in a different form that separates intentions from behaviors, Bratman's (1992) theory specifies that a joint activity is collaborative if 1) each agent intends to participate in the joint activity (CJA), to respond to others' intentions and behaviors (MR intention), and to provide others with support if necessary (CMS); and 2) each agent is responsive, in action, to the others (MR behavior). Note that, although collaboration requires commitment to mutual support (the intention), it does not necessarily require that the action of mutual support (the behavior) be present every time. A collaborative activity may be carried out smoothly without any agent’s encountering any problems during the course of the activity. If, however, any agent encounters problems, the others must provide support (CMS behavior).

3.2 Extensions to the Collaboration Theory

We developed our conceptual model based on reviewing the literature on the collaboration theory and human-computer collaboration models in depth (Bratman, 1992; Grosz, 1996; Grosz & Kraus, 1996; Kamar et al., 2009; Silverman, 1992; Terveen, 1995). This conceptual model builds on Bratman’s formulation of collaborative activity, extends his theory by incorporating additional intention and behavioral components concerning collaborative activity, and specifies the underlying relationships between CJA, MR, and CMS. When applying the model to user-system interaction, we further conceptually separate the system designer’s intentions from system behaviors. In addition, we posit that, if used to help design collaborative behaviors and functions of complex information systems, the model can help enhance the effectiveness and efficiency of and user satisfaction with the systems.

This conceptual model extends Bratman’s (1992) collaboration theory as follows:

- We believe that agents should be able to *re-plan* during the course of a collaborative activity. The intention to re-plan is a natural part of the CJA feature such that the participating agents are willing to refine, adjust, or alter the initial plans when important factors concerning the joint activity (e.g., the environment and external conditions, the shared goals, and the agents’ subplans) change. Re-planning makes possible the alternative routes and recipes toward successful collaboration.

subtle nuances of how a user performs a task, can be converted into explicit knowledge by recording, logging, and analyzing the user-system interactions.

- Bratman’s (1992) theory focuses only on one-time collaborative activities in single sessions. It does not consider the nature of mutual responsiveness resulted from long-term collaboration between participating agents in multiple sessions. We posit that mutual responsiveness in long-term collaboration manifests itself in *learning* and *adaptation* (Silverman, 1992; Terveen, 1995). Through repeated interactions with others, one will learn the behavioral patterns of the others and adapt to these patterns over time in order to increase the likelihood of success in future collaboration. These learned behavioral patterns become part of the contextual knowledge that the participating agents maintain.
- Although Bratman’s (1992) theory considers the collective presence of CJA, MR, and CMS as the minimal requirements for a collaborative activity, it does not articulate the underlying relationship among these three features. In our model, we posit that:
  1) *MR is based on and guided by CJA*. For the relationship between CJA and MR, Bratman (1992, p. 328) simply states that MR is “in the pursuit of” CJA without elaborating on it further. We believe that the re-planning component in CJA enables MR intention. That is, individuals' intention to re-plan, if necessary, leads to their intending to keep an eye on others’ behaviors and to adjust their plans and behaviors accordingly.
  2) *MR intention is a prerequisite of CMS.* That is, an individual’s intention to provide assistance to others depends on the individual’s intention to respond to others’ intentions and behaviors.
  3) Because of the two reasons above, a hierarchical relationship exists between CJA, MR intention, and CMS in the sense that CMS requires the presence of MR intention, which, in turn, requires the presence of CJA. This position is consistent with the previous human-computer collaboration model that Grosz (1996) proposes in which both MR and CMS depend on the existence of CJA.
- We posit that the model needs to incorporate *additional behaviors* for collaborative activity. The three features in Bratman’s (1992) model are mainly intentions (i.e., CJA and CMS); only MR comprises both responsive intention and behavior. In other words, Bratman’s theory specifies only the minimum requirements for collaboration. We believe that concrete behaviors that result directly from CJA and CMS are also important components of collaborative activity. Specifically, CJA leads to agents’ behaviors of maintaining the shared goal, executing the shared plans, and sharing knowledge. The agents’ behavior to provide help to others when necessary is directly motivated by their CMS. Although some of these behaviors (e.g., providing help) may not have to be present in every single collaborative activity, these additional behaviors together with the related MR behaviors (staying aware of the others’ relevant intentions and behaviors and adapting one’s intentions and behavior accordingly) do contribute to the success of a collaborative activity.

3.3 The Conceptual Model

Building on the extensions that we specify above, we model user-system interaction as a collaborative activity. However, we first have to address an important challenge: Bratman’s (1992) original theory formulates collaborative activity between human agents, while our model applies to interactions between human agents and computer systems. We recognize that computer systems, such as ERP systems, cannot independently form and maintain intentions such as humans do. That is, one cannot expect a computer system to make a commitment independently. Consequently, CJA, MR, and CMS are all meaningless for a computer system, and one cannot apply the collaboration theory to user-system interaction without adjustments.

On the other hand, computer systems are artifacts designed by human designers, and these humans can and do form and maintain intentions. Therefore, a designer has the capability to intend the system to collaborate with users. Based on the intent, the designer can design and implement the system to make it capable of working with users as a partner. As a result, in our model, we separate the intentions and behaviors on the system side and distribute them between the two different types of agents: the designer and the system. Because of this separation, users cannot directly observe and perceive the intentions of designers during their interactions with the system. The users can, however, observe the system’s collaborative behaviors. This separation is quite important because it addresses the challenge of applying the conceptual model in the context of user-system collaboration.

Figure 1 presents our conceptual model, which separates collaborative intentions of the designer from the collaborative behaviors of the system. This model treats the collaborative features (CJA, MR, and CMS) in Bratman's (1992) theory are as latent, non-observable intention components (denoted with boxes with dashed borders). The vertical arrows represent the hierarchical relationships between them (see above discussion regarding these relationships). Each designer's intention leads to one or more system behaviors, which are a concrete result of the system design and implementation process. System behaviors reflecting CJA include maintain shared goal, execute shared plans, and share knowledge; one can observe MR intentions in behaviors such as *keep track of user's behaviors*, *adjust own behavior*, and *adapt to user's behaviors*; and, finally, CMS is realized through the *provide support* behavior. We directly derived six out of the seven system behaviors from Bratman's theory, and we explicitly describe them above in Section 3.1 and discuss them more in this section. The seventh, adapt to user's behaviors, concerns long-term collaboration (which we discuss earlier in this section). A*dapt to user's behaviors* differs from *adjust own behavior* because it is formed and manifested mainly in long-term, repeated interactions. One can illustrate this difference using the global positioning system (GPS) as an example. During a trip, if the GPS device detects that the driver deviates from the recommended route, it will automatically recalculate and recommend an alternative route to the destination. This demonstrates its *adjust own behavior* capability. On the other hand, if the GPS can detect that the driver always avoids toll roads when possible in many trips, it will automatically recommend routes with no toll roads. This is an *adapt to user* behavior.

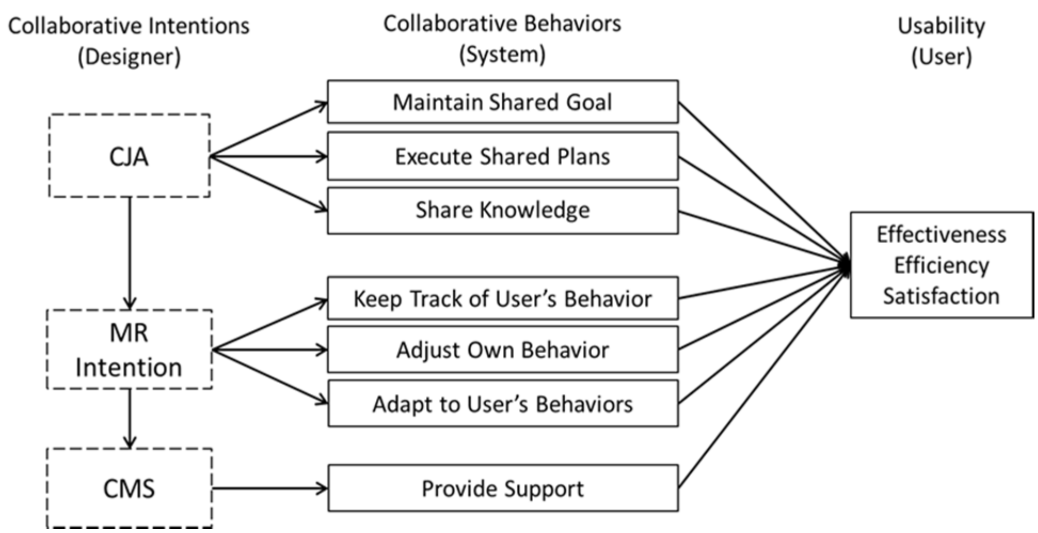

Figure 1. The Conceptual Model for User-system Collaboration

We emphasize that this conceptual model is not a structural model intended for measurement purposes but a specification of 1) the hierarchical relationships and dependencies between the intention components and 2) the differences between the latent (designer) intentions and concrete, observable (system) behaviors. One could, however, use the model as a foundation for a measurement model if one specified indicators for the behavior components.

The model is not yet complete in this form because it omits another underlying layer that one needs to specify between the designer intention and system behavior: the computer system must possess certain *capabilities* to enable its collaborative behaviors. By assumption, human agents who participate in a collaborative activity have the necessary abilities to perform collaboration-relevant actions. Agents can act on their intentions only with behavioral capabilities. For example, if two people intend to sing a duet (an example that Bratman (1992) introduces), they each must be able to make vocal sounds. In the context of user-system interaction, such behavioral capabilities in the system take the form of designed and implemented system functionality. Only with these capabilities (functionality) can a system successfully collaborate with the human user.

3.4 Capabilities of Collaborative Information Systems

Directly based on the conceptual model we introduce above, we propose seven behavioral capabilities that collaborative information systems should have. One can see these capabilities as behavioral indicators of the designer's collaborative intentions (see Figure 1 above).

1) *Capability of identifying and maintaining the shared goal.* When a user uses a system, the user-system interaction usually centers on a specific task (such as creating a purchase order) that the user intends to accomplish in a specific context at a specific time. The user's goal to successfully accomplish the task becomes the common goal that the system and the user must share. With current information technologies, most systems do not yet have a high enough level of artificial intelligence to automatically detect and capture a user's goal through a natural dialog. Therefore, the system must provide a mechanism through which the user can specify the task goal in a way that is intuitive and understandable for the user.
2) *Capability of executing the shared plans.* When working to achieve the shared goal, the system and the user typically have an initial, general plan of the actions required. For users, their mental model, which they develop or acquire through training, represents the plan for the tasks or processes. For example, the task of creating a purchase order usually requires two steps: 1) filling out the order header information (e.g., the issue date, the supplier name) and 2) inputting the order details (e.g., the names and qualities of the materials to be purchased). For systems, they have the plan built-in by design or configuration in the form of a prescribed business process model or workflow, which the system and users need to follow to accomplish the task. The prescribed models in a good system partner should be consistent with the users' mental models. In other words, the prescribed models in the system must be logical and make sense in the task contexts so that they can serve as the "shared" plans.
3) *Capability of sharing contextual knowledge.* An information system can be an effective, helpful co-worker only if it communicates with the user when appropriate to successfully complete the activity. Because the user may only have incomplete knowledge of the information embodied in the system, the system must share information required to complete the task (e.g., prescribed process models, required input, etc.). For example, SAP's Business Blueprint provides users with a diagram of the steps and actions necessary to accomplish a business process or task. Some ERP interfaces also identify which inputs are optional and which are mandatory. Indeed, the communication and knowledge sharing do not have to be accomplished using a natural language dialogue. Instead, the ordinary point and click-based user interface or other emerging user interface technologies can be skillfully designed to provide an effective means of communication.
4) *Capability of keeping track of the user's actions and status.* To be able to adjust to the user's behavior, the information system must keep track of the task's progress or a process based on the actions the user has taken. If the system notices that the user must complete another task in order to proceed with the current task, it should provide an easy way to navigate to and from that task.
5) *Capability of re-planning and adjusting in response to the user's short-term behaviors.* Computer systems are generally designed to respond to users' behaviors by prompting for and accepting user inputs, fulfilling user requests, generating outputs, and so on. In addition to the existing responsive mechanisms, the system should allow users to choose alternative paths that deviate from the predefined built-in process models (re-planning). Additionally, the system should be implemented to incorporate the vocabulary and terminology used by the particular organization where the users work.
6) *Capability of learning and adapting to the user's usage patterns over time.* The system should record the history of the interactions with the user in multiple sessions of system use. Over time, the system may extract each user's specific behavioral patterns and make corresponding adjustments to its own behavior. For example, the system can automatically recommend and populate user input based on previous interactions with the users. Further, if the system learns that a user always works on only a few modules, it may modify its menu layout to provide direct shortcuts to the frequently used modules while allowing the user to expand the menu for other modules if needed.
7) *Capability to provide users with assistance if needed.* This capability is most critical when users cannot complete their part without assistance. In such circumstances, the system should

be able to detect that the user has made a mistake, report the error, diagnose the possible causes, and provide guidance regarding one or several possible effective paths for moving forward.

If one designs a complex information system with these collaborative capabilities in mind, one can achieve a new division of responsibilities between the system and user. Considering the strengths of information systems in storing, integrating, retrieving, and presenting large quantities of data related to organizational processes and practices, the new division of labor shifts more responsibilities onto systems. This new division of labor allows systems to take a more extensive, active role than they currently do and to further overcome human limitations by keeping track of task and process status, maintaining and sharing task- and process-relevant knowledge, adapting to users' behaviors, and providing users with assistance. As a result, users will no longer have to spend a long period of training time in learning and memorizing the system's prescribed models and ways of performing tasks. In addition, users will not have the sole responsibility for maintaining contextual knowledge and diagnosing and solving errors. Consequently, we can reasonably expect that users' perceptions of systems usability will improve.

3.5 The Impact of Collaborative Behaviors on Usability

In this section, we provide the rationale for the collaborativeness-usability relations and, thus, for the relationship between collaborativeness and complex information systems. The definition of usability identifies three key dimensions: effectiveness, efficiency, and user satisfaction (ISO, 1998). Effectiveness refers to the accuracy and completeness with which users can achieve their goals. Efficiency refers to the ratio between effectiveness and the resources required to achieve the specific level of effectiveness. Satisfaction refers to the users' comfort with the system and positive attitudes toward its use. In this section, we discuss how the seven collaborative capabilities and behaviors may potentially influence the three dimensions of usability. We again emphasize that these proposed collaborativeness-usability relationships are largely conceptual, exploratory, and illustrative in nature. We propose them as mechanisms through which the level of system's collaborativeness impacts the dimensions of usability. As such, they form the link between a system's collaborative design and system's usability, which is particularly important in the context of complex information systems as Albers and Still (2011) have articulated.

1) A system that can identify and maintain a shared goal with a user in a specific task context is likely to be more usable than a system that does not have this capability. The system's functions that allow the user to easily communicate the task goal to the system will improve the likelihood that the user will succeed at the task. It is difficult—if not impossible—to achieve accuracy or completeness in the task execution if the system's and the user's goals are not aligned (i.e., if the user is attempting to use system functionality in a way that is not aligned with the user's real goal). This capability will also increase the user's level of satisfaction due to a stronger sense of shared commitment.
2) A system that has the capability to execute the shared plans performs in a way that is more compatible with the user's mental model of the task and its execution. It is very possible that users' mental models vary significantly. A system's ability to re-plan and adapt to specific users in specific situations will ensure that they do not need to learn models and execution logic that do not make sense to them. Avoiding conflicting perspectives will allow the user to stay on track and, thus, improve both effectiveness and efficiency.
3) If a system can fill the user's gaps in contextual knowledge, the user will be able to perform tasks more accurately because of a better understanding of the task context, which will improve the system's overall effectiveness. Sharing of contextual knowledge will also impact efficiency because the user can perform tasks faster due to the reduced need to seek information and confirmation regarding how to perform them in their specific contexts. Such conceptual clarity can also lead to a higher level of satisfaction due to an increased sense of contextual awareness.
4) Only if a system is aware of the user's actions does it have the potential to guide the user back to a more effective path of task execution in situations when the user makes ineffective or incorrect choices. If the system can prevent the user from taking unnecessary or incorrect actions in the context of the shared goal, it will lead to improved effectiveness and efficiency and potentially also increased satisfaction due to decrease in frustration.

5) Awareness of status has little use if it is not used to help a system re-plan and adjust. In other words, the true usability benefits of a system's awareness of a user's status can be realized only if system can react to the changes in the user's status.
6) The learning and adaptive capability of a system will help the user accelerate task performance times and avoid potentially costly errors, which will increase efficiency and potentially satisfaction.
7) A system's capability of providing assistance when needed will, if executed successfully, have a positive impact on the user's ability to complete the task accurately and completely, which will increase effectiveness. This capability is particularly important for less-experienced users. It is also likely that the user's satisfaction will improve if appropriate help is available and the user is not left alone in problem situations.

Note that, in many situations, the presence or absence of one collaborative capability may have impact on all three usability dimensions. For example, if a system can provide the user with necessary and timely assistance in error situations, it will help the user successfully complete tasks (high effectiveness) faster (high efficiency), which will lead to a higher level of satisfaction. On the other hand, a lack of several collaborative capabilities may cause specific usability issues. For example, a high data-entry error rate (low effectiveness) may occur because the system does not provide or share the information about the meaning of each field so that the user enters the data into wrong fields or because the system fails to detect the error and to provide necessary support at the time of the user's incorrect action. Similarly, a terminology problem (see the example in Section 4.3.2) can be attributed to the system's failure to maintain shared plans (i.e., the system's plan is inconsistent with the user's), share knowledge (i.e., the system does not provide adequate explanation of the meaning of terminology), or adapt to users (i.e., the system does not adopt the language used by the users in the organization). In other words, the associations between a system's collaborativeness and usability are potentially dense and multifaceted. Consequently, rather than expanding all possible relations between collaborative capabilities/behaviors with individual usability dimensions, we treat usability as a whole in the current model (see Figure 1).

In Section 4, we use two ERP system contexts as sources of examples to illustrate these collaborative capabilities and the ways in which the presence and absence of these capabilities may impact user's perceptions of system usability.

4 Illustrating Collaborative Behaviors in the Context of Usability

Based on Bratman's (1992) collaboration theory and related human-computer collaboration models, our model proposes and describes the design intentions and system capabilities and behaviors at a high level that enable a system to collaborate with its human users. We provide a set of examples regarding 1) the system capabilities and the resulting behaviors that users recognize and perceive as collaborative and 2) the ways in which users think these capabilities and behaviors (or the lack of them) would impact system usability. We use ERP systems as an example of complex information systems.

4.1 Research Methodology

We employed a qualitative method in order to fully explore the nature of the interactions between a system and its users and the context of these interactions. Although we used a qualitative method, philosophically, our ontological and epistemological positions are aligned with positivism. We believe that the phenomenon we studied exist independently of our beliefs regarding it, and we believe that it has been possible to articulate and will be possible to empirically evaluate the relationships between the constructs specified in our conceptual model. Our use of the case study material is compatible with Eisenhardt's (1989) "soft positivism". We specifically did not intend to use the grounded theory approach, which is the primary tool of interpretivists. Consequently, we specified and defined the constructs in the study **a priori**; they did not emerge only from the qualitative data. The examples we use in this section only illustrate the conceptual model and the impact of system collaborativeness on usability. We do not formally test the theory with the empirical evidence but use the empirical evidence to demonstrate the proposed conceptual model's feasibility.

The quotes in the examples come from a field study of 26 interviews with users in two companies located in northeastern US: a middle-sized software and service vendor (company A; 11 interviews) and a medical imaging device manufacturer (company B, 15 interviews). The two companies used two different ERP systems, both of which are among the leaders in the general ERP software market. Each of the two systems comprise literally hundreds of modules including production, inventory control, marketing and sales, human resources, finance and accounting, and so on. Neither system, as far as the modules used in each company are concerned, had artificial intelligence features that could make inferences about user-system interactions. The interviewees' organizational roles varied from entry-level jobs (e.g., assistant controller) to the top management (e.g., vice president of finance). Users' experience levels ranged from occasional users to regular users who used the system on a daily basis and, finally, to super users with extensive knowledge and expertise about system functionality and features. At the time of data collection, company A had used the system for about two years (after a 20-week implementation process), and company B for about six-and-a-half years (after an implementation process that lasted for approximately a year).

We conducted semi-structured interviews based on a set of 16 questions (which we can provide on request). We designed these questions to explore users' perceptions regarding the level and type of their system's collaborative capabilities and behaviors in user-system interactions. For example, we phrased a question regarding their system's adaptation capability as: "[A]s you work with the system on this [task], can you give examples of the system taking any of your previous actions into account? Is that helpful?". We first tested and revised these questions in a small pilot study before we used them in the formal interviews. Two investigators were present for each interview. By using multiple investigators, we could view from different perspectives (Eisenhardt, 1989). Typically, one of the investigators had the primary responsibility for the interaction with the interviewee and the other observed the interview and took notes.

We extracted the empirical illustrations via a formal coding process in which we first divided the total of 1,155 pages of transcribed interview data into multiple-sentence chunks. We then assigned these chunks to zero, one, or many coding categories, which included both the high-level collaborative features (CJA, MR, and CMS) and usability dimensions (effectiveness, efficiency, and satisfaction). We further broke down each of the three high-level collaborative feature categories into subcategories based on the model in Figure 1 (e.g., maintaining shared goals, executing shared plans, sharing knowledge for the CJA category). For the usability categories, we also distinguished between positive and negative perceptions by analyzing the evaluative words and phrases that our participants used. In the examples that we cite in the following sections, we embed these usability codes in the quotes. For example, we coded a chunk as: "It's quick {+Efficiency}, we could just open this thing up, and we can just click on this, that's the transaction code which is probably another pet peeve {-Satisfaction} with people, sort of these long cryptic transaction codes that aren't all that meaningful.".

We employed QSR NVivo, a widely used tool for organizing and coding transcribed interviews. Four researchers, including the authors and two graduate students, were divided into two groups of two coders. One group was responsible for coding the collaborativeness aspect and the other for the system usability aspect. The two groups worked independently. In each group, the two coders coded each interview transcript independently. They identified and addressed coding discrepancies via correspondence and face-to-face meetings. They resolved final disagreements about disputed text passages through in-person discussion until they reached agreement. Prior to this resolution, coding reliability was within the acceptable range of 70 percent inter-coder reliability (Miles & Huberman, 1994).

4.2 Collaborative Intentions of Designers

When modeling user-system collaboration, we separate designer intentions and system capabilities and behaviors. From the user's perspective, designers' intentions are latent and unobservable (see Section 3.3). Although we do not test the model per se, we still hope to find qualitative evidence for the designer's collaborative intensions. As part of exploring the role of collaborativeness, we searched extensively though the documentations that the vendors of the two ERP systems provided. We found no indication in either one of the cases that suggested that the collaborative perspective might have been used to design and develop the ERP systems.

We also analyzed the views of the super users and developers who participated in the ERP implementation and customization projects in our case organizations. Their comments shed light into their views regarding the role of ERP systems in user-system interactions. A super user from company B commented on the relationship between the system and users:

> *They [the end users] feel like they're probably enslaved by the system, whereas I look at it as a tool. And I know it can only do so much. It's just a database; it's not a brain…. In some*

> *relationships you like to be the one with the brain. That way, that thing doesn't talk back to you, doesn't complain.*

It is fairly clear that this super user treated the ERP system simply as a tool and the user-system relationship as a master-servant one. Another super user pointed out that the goal of the implementation was not to help users:

> *This was implemented for financial management. So, it had a huge financial bias when it was set up. So, as a result it wasn't operational centric. It wasn't customer service centric. It was finance centric. So, [ERP] orbited around there.*

A developer from company A, however, hinted at his intentions to facilitate end users' work:

> *You have to have the user's perspective and to understand that what their job is, they want to do it with the easiest you know or less pain possible. And you have to have a little bit of empathy to see their point.*

In terms of the system's adaptability, this developer indicated that it was not part of their implementation:

> *Adaptability, no. And this has to be designed to be.*

Our examination of a number of other comments from these super users and developers revealed that, although the implementation teams might have considered user perceptions, they did not intend to implement collaborative ERP systems that could work as knowledgeable partners for end-users. Combined with the lack of this perspective in underlying systems design, users perceived that the resulting systems lacked many collaborative capabilities and exhibited many usability problems. In Section 4.3, we provide examples about the collaborative capabilities and their impacts.

4.3 Collaborative Capabilities and their Impact

We organize this section based on the seven collaborative capabilities we present in Section 3. For each capability, we present examples of system behaviors that demonstrate either the presence or absence of the capability and user perceptions of these system behaviors. The examples also demonstrate the links between the collaborative capabilities and users' perceptions of system usability (which comprises, according to the ISO definition, effectiveness, efficiency, and satisfaction).

4.3.1 Capability of Identifying and Maintaining a Shared Goal

The ERP systems we examined provided two major mechanisms for users to specify a task (or transaction) goal; namely, menu trees[3] and transaction codes[4]. Although their interfaces had a task search box, the users had to use a transaction code or exact task name to find a task. A convenient task search mechanism "like a Google field" (a term that one of the users used) that allows for partial, fuzzy searches did not exist. Because a menu tree was quite deep with multiple levels, the users found it difficult to locate an intended task while navigating a menu tree. They were afraid of getting lost in the middle of the tree, especially when the task was new or unfamiliar. Although transaction codes allowed direct access to transactions, they often were cryptic and required memorization. Therefore, because both system mechanisms lack the ability to effortlessly identify and maintain the shared goal, they appear to be complicated for users. The users in company A noted as much in the following quotes (we also include our usability category codes in curly brackets in these quotes):

> *Some of these menu paths are a little cryptic. And so it's hard {-Efficiency} to get the lay of the land just by looking at it. And if you were brand new to this, I suspect you could drill down many of these paths wondering where this ends. And then if you found what you wanted you might think, "I'm never going to find that ever again.* (Accounting manager)

3 A menu tree is a hierarchical structure in which tasks (or transactions) are nested deeply in multiple layers of modules. In the ERP system used in company A, for example, the menu path for creating a purchase order had eight levels: root > logistics > materials management > purchasing > purchase requisition > follow-on functions > create purchase order > via assignment list.

4 A transaction code is a sequence of characters that a user can use to access a specific task directly without having to navigate the menu tree. As an ERP system may cover a whole range of business tasks, the number of transaction codes is often quite big. For example, there were more than 100,000 built-in transaction codes in the ERP system that company A used.

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 {-Effectiveness}. (Accounts receivable and collection user)

[N]ot everything is intuitive in this system. It is all code based and if you don't know the code and you don't know where to look. If I was looking for one particular thing [like] this Z-report, I would have to look through every one of these {-Efficiency}. I wouldn't necessarily know where to go {-Effectiveness}. (Order entry user)

4.3.2 Capability of Executing the Shared Plans

This collaborative capability requires that the system's task-related plans be consistent with the users' plans. If the two types of plans are incompatible, users would perceive the system to be non-logical or "making no sense". Such plans may take the form of a workflow for a business process (e.g., order fulfillment) or a navigation path that comprises a sequence of steps for a task (e.g., purchase order creation). If the workflow or navigation path is not intuitive from the user's perspective, it will cause great confusion and frustration. The vice president of finance in company A noted:

[S]o for folks like me who aren't in the system all the time, you can see I struggle {-Satisfaction, -Efficiency} because it just doesn't make logical sense. Versus something that's more of a basic system. I mean those systems tend to be a lot easier. And you fumble your way through it because they are intuitive. And, they do follow business logic. And, they make a lot more sense.

The users also expressed confusion when the terminology that the system used differed from the business language that users employed in their daily work. To the users, the terminology that the system used was not logical and made no sense. An accounting manager in company A with 12 years' experience of using ERP gave an example:

[T]hings are buried under something. You'll think it's an accounting thing but it will be under IS, and then it's another thing they call environment {-Effectiveness}.

4.3.3 Capability of Sharing Contextual Knowledge

The users commented on the system's inability to share knowledge and information relevant to the task context. First, the ERP systems did not provide guidance along the system-prescribed navigation paths for tasks that required multiple steps and screens[5]. Users noted that it was often not straightforward to move through the screens. As a result, they had to remember the sequence and ordering of the screens along the navigation path, the purpose of each screen, and the result from running each screen. A user handling sales orders in company B commented on the system's inability to provide task guidance:

It doesn't tell you what steps to take next. You have to basically, like I said, know what the next step is for your process, for what your job title is to do {-Efficiency}.

Second, the ERP systems often did not provide contextual information that could help users perform their tasks, such as progress toward completing a task: where the user is along the navigation path, what the user has done, what remains to be done, and so on. A user in charge of order entry in company A noted:

[When entering a sales order], there is nothing that is going to tell you you've entered in the header and now you go into the line item and you update this and you update that. The system doesn't do any of that. It's more of its user having to know the steps {-Efficiency}.

Another example of useful context knowledge that the system can potentially share with users is the information regarding the larger business process of which a task is part. A user working on accounts receivable and collection in company A noted:

With the cash accounts receivable side of things, I am the absolute last line in that whole process. It begins in sales, the deals close, [they go to] the billing folks. They bill. And then it shows up as money that I need to go collect. So I'm the last line in that whole process. So that will be one where I have to look backwards all the time to figure out what is happening.

[5] We again use the example of the purchase order creation task in the system used in company A here. The navigation path comprises multiple screens in addition to the main creation screen: purchasing organization, material, vendor, plant, and delivery date.

> *[Interviewer: Is it obvious to the system how to look backwards?] No {-Efficiency}, that was, again, more from getting the crash course and asking questions, feeling like I'm not going anywhere {-Effectiveness} and then raising the flag.*

However, in some cases, the users were pleased about the system's capabilities related to sharing, which a quote by an account payable specialist in company A illustrates:

> *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 {+Efficiency, +Satisfaction}. It's like a story when you go through them.*

4.3.4 Capability of Keeping Track of User's Actions and Status

Unlike many other capabilities, we found this one in a number of cases in our field data. The users were consistently pleased with the features in both ERP systems that allowed them to keep working on several tasks at the same time and save the incomplete jobs. A sales order specialist in company B commented:

> *It saves everything. It saves the steps, the status, where it's at; it saves everything that you've made the change on. If you were to change it from step 10 to 20, it saves all that. And then the next time you go in it, it will pop up and show {+Effectiveness, +Efficiency}.*

4.3.5 Capability of Re-planning and Adjusting Behavior in Response to User's Behavior

Regarding this capability, many of the examples from the interviews were both positive and negative. The positive ones related to the system's flexibility to re-plan and adjust its behavior, illustrated through its ability to detect user actions and act accordingly, and changes in system behavior based on access rights. The users consistently favorably mentioned the auto-population feature in the ERP system. For example, the system could automatically fill out the product price field if the user entered the product number in a sales order.

We also found examples of cases where this capability was absent. In these cases, negative user comments related to, for example, the systems' poor handling of cancellation of operations, always showing controls that a user never accessed, high risk of accidentally repeating a transaction, and lack of intelligent default values. The following comments by users in company B illustrate two of these cases:

> *Every time I pull up the list of responsibilities, it lists it the same way all the time. It's in alphabetical order. It doesn't adjust to what you're doing {-Efficiency}.* (Product line group leader)

> *I have to change it [the date] every time I used it {-Efficiency}. The default is in 2007. (*Sales order specialist)

4.3.6 Capability of Learning and Adapting to User's Behavioral Patterns over Time

The systems lacked in their ability to learn and adapt to users' behavioral patterns over time. For example, they could not record users' repetitive usage patterns (e.g., preferred layout, the frequently used navigation paths, etc.), and they forced them to expend unnecessary manual effort when performing their tasks as several users said:

> *What I do every time is right away I hide these three [fields], I hide these two, and I hide this one. So I do this every time I get in here {-Efficiency}. [I]f there were a way that (ERP) recognized that I do that every single time I go in, it would be very nice just to say, "I noticed that you have these tendencies!"* (Accounts receivable and collection user in company A)

> *(I)f it could recall everything that you've done and things that you use the most, your last 20 responsibilities that you've gone into, if it could adjust to the user and, "OK, these are the fields that are most common to you," it could simplify a lot of things [-Efficiency].* (Product line group leader in company B)

Very commonly, an ERP system's inability to adapt to users' actions and respond to users' needs made users turn to external systems, particularly in reporting. Users in company B explained how they used Microsoft Excel or Access for reporting purposes:

> *[E]verything that we do report wise, we export onto Excel and we edit it from there. And we export it into Excel {-Effectiveness} only because there are certain things on the report that I*

> *need, and then there are certain things that other departments, like shipping needs, that I don't need. (*Sales order specialist)

> *What I do is—I vet this data—and then I pull it into an Access database {-Effectiveness} that I write because I find that [ERP] doesn't give me exactly what I need. (*Director of technical support)

4.3.7 Capability of Providing Support

We found a number of examples of users' perceiving the system to be effective when they ran into trouble in completing their tasks. Most of these situations related to the guidance that error messages provided; for example, one user said:

> *Yes, it can give you error signs {+Effectiveness}, saying something is not right, or this field is incorrect, or this has already been inputted.* (Accounts payable specialist in company A)

However, the absence of the support capability was demonstrated by, for example, the fact that users often found the system responses unclear and error messages cryptic. They had to seek help from other coworkers or IT specialists as the following two examples illustrate:

> *The system just doesn't know what the problem is. It doesn't know how to diagnose those {-Effectiveness}. So this one says, "No suitable documents found. Message, ME260.".* (Assistant controller in company A)

> *The error messages are horrible {-Satisfaction}. If it's a simple error message like "this part doesn't exist in the database" and then that's pretty simple. But when something fails, I just get some bizarre code that I have no idea what it means. I have to take a screen shot of it and send to MIS {-Efficiency}.* (Receiver in company B)

Sometimes, a user got stuck in the middle of a task and could not proceed. In this case, the system should be responsible for recognizing the user's need for help and offering appropriate assistance. However, the ERP systems did not have such capabilities, which the following example in which the accounts receivable and collection user in company A encountered a problem during a transaction illustrates:

> *Now, the thing is that in this case, the system is not reaching out to you saying that you obviously need help. It's me having to go find it {-Efficiency}. Just to go back to that GL account scenario [in which I didn't know what the account number was], rather than just telling me you had to put something in, if it knew automatically what that one was supposed to be, and once you failed, say three times, or X times putting in the wrong one and then at that point, it would query you "You obviously need help here". And then it would send you to a help desk function.*

The system capabilities and behaviors illustrated in these examples impacted the users' perceptions of system usability. On one hand, the presence of collaborative capabilities often led to positive perceptions toward the system. For example, the capabilities of sharing transaction history information and the features of automatic population of previously entered data helped users complete their tasks faster (high efficiency) and resulted in positive attitudes toward the ERP systems (better satisfaction).

On the other hand, the absence or lack of collaborative capabilities negatively affected the users' perceptions of system usability. Because of the complexity of the menu trees, transaction codes, process models, and the ERP terminology—all violations of the shared goal and shared plan components of CJA—users ended up spending significant amounts of time memorizing required details in order to complete their tasks as the accounting manager in company A commented:

> *They memorized it {-Efficiency}. They've memorized what needs to happen too. So, you need to know the screens. And, you need to be trained in what the process is.*

Similarly, the need to memorize the sequences of screens required to perform tasks and to undergo extensive training is a strong indicator of the systems' poor efficiency because it added to the users' cognitive load and the time they spent completing their tasks with the systems. Moreover, training is time-consuming and, for many, a burden; yet, without it, users would be unable to use the system effectively.

The systems' inability to adapt to users' behavioral patterns led to their using external systems (e.g., MS Excel) as a workaround, which suggests that users perceived the ERP systems as ineffective for achieving their goals.

In terms of support capability, the lack of intuitive error handling mechanisms manifested by cryptic error messages, a failure to detect users' need for assistance, and an absence of guidance regarding possible solutions prevented users' from accomplishing their tasks (low effectiveness), caused delays to their tasks (low efficiency), and made them feel negative about the system (poor user satisfaction).

Table 1 summarizes the collaborativeness-usability relationships that our examples illustrate. Note that an absence of a relationship in this table does not necessarily mean that no such relationship exists. It simply means that the relationship is absent from the limited set of examples we selected for this paper.

Table 1. Summary of the Relationships between Collaborative Behaviors and Usability

Collaborative behavior	Effectiveness	Efficiency	Satisfaction
Maintain shared goal	√	√	
Execute shared plans	√	√	√
Share knowledge	√	√	√
Keep track of user's behavior	√	√	
Adjust own behavior		√	
Adapt to user's behaviors	√	√	
Provide support	√	√	√

5 Discussion

The human-computer collaboration perspective (Bratman, 1992; Grosz & Kraus, 1996) provides a useful framework for examining user-system interactions in the context of complex information systems. Our research has several implications for research and practice.

5.1 Implications for Research

With our conceptual model, we make three main research contributions. First, we extend Bratman's (1992) collaboration theory by including *additional intention and behavioral components* (e.g., re-planning and adaptation). As a result, the conceptual model applies to not only short-term, one-time activities but also long-term, repeated interactions. In addition to the collective presence of the three features that Bratman's original theory requires, we posit *a hierarchical relationship* among the three features: CJA guides MR, which, in turn, CMS depends on. Although prior research on human-computer collaboration has proposed some similar components (e.g., adaptability by Terveen, 1995) and relationships (e.g., the dependency of MR and CMS on CJA (Grosz, 1996)), our research is the first to synthesize and organize all the features and relationships into a complete, comprehensive, and logical framework.

Second, our model separates the *designer's intentions* from the *system's behaviors*. We formulate user-system interaction as collaborative activity in which one agent is a computer system. Recognizing a system agent's inability to form and maintain independent intentions and make commitments, we separate each of the three collaborative features into the designer's intentions and the system's behaviors. The designer's intentions for user-system collaboration, although latent and unobservable, can be manifested by the system's collaborative behaviors and perceived by users. Such separation makes it possible to build a measurement model in the future to quantitatively assess the levels of collaborativeness of complex information systems.

Third, we propose *seven core system capabilities* that one must design and embed into complex information systems to enable them to work as knowledgeable partners of users. In other words, there is a logical chain connecting *intentions*, *capabilities*, and *behaviors*: a designer's intentions lead to the designer's designing (and implementing) the system's collaborative capabilities (in the form of system functionality), which system's collaborative behaviors during user-system interactions further manifest. As Bratman's (1992) original theory models collaborative activities among human agents and assumes humans to possess the necessary capabilities to perform any collaboration-relevant behaviors, he has never explicitly discussed the importance of the capabilities. We believe our explicit specification of collaborative capabilities for the system is important because these capabilities are the direct (and only) mechanism for enabling a link between a designer's intentions and a system's behaviors. From a practical standpoint, they can bridge the gap between the abstract components in the conceptual model and concrete design guidance that designers can follow when developing collaborative systems.

Furthermore, although the model is primarily descriptive, includes mainly collaborative components, and cannot yet be used as a predictive model for system usability, it forms a foundation for a more sophisticated understanding of the *relationships between the system collaborative capabilities and usability*. Our empirical illustrations gathered from our field study on ERP systems show that the systems in our target organizations had not been designed in a way that would have allowed them to fully collaborate with their users. This lack of collaborative capabilities has been associated with various usability problems, which one could potentially address if one shifted more responsibilities from the user to the system. Our conceptual model calls for *a new division of responsibility* based on which the system is designed to work toward its maximum strengths and to take more responsibility than it currently has. As users are less burdened with learning how to use the system to accomplish business tasks, finding and maintaining contextual information of tasks and processes, or diagnosing and fixing errors, they may experience a lower cognitive load and be able to accomplish their business goals more effectively and efficiently and with better satisfaction. We next plan to systematically investigate and model the relationships between the seven collaborative capabilities and the three dimensions of system usability.

Viewed more broadly, our research bears intrinsic connections with research on various IS phenomena. We believe that the insights we provide through this research help illustrate the impact of a system's design characteristics on users' key beliefs regarding the system (e.g., perceived usefulness and perceived ease of use) as Benbasat and Barki (2007) suggest. The human-computer collaboration perspective also has the potential to stimulate a more in-depth exploration and investigation of the links between the collaboration model and other theories regarding information system use, such as TAM (Davis, 1989), UTAUT (Venkatesh, Morris, Davis, & Davis, 2003), and the task-technology fit model (Goodhue & Thompson, 1995). For example, we need to understand whether and how users' perceptions of a system's collaborative behaviors might affect their behavior of accepting and using the system. It is also possible that the extent to which a system acts as a collaborative partner might affect users' effort and personal performance expectations. We need a more systematic and comprehensive analysis of a larger body of data to explore these ideas further and to evaluate the internal and external validity of our framework.

5.2 Implications for Practice

Our conceptual model also builds a foundation for developing guidance for designing collaborative capabilities in future complex information systems. As the empirical examples illustrate, some simple user interface features (e.g., the transaction/task search function) may increase an ERP system's collaborativeness to an extent. However, we must emphasize that one cannot achieve the full range of collaborative capabilities via merely patching user interfaces. One must use a holistic approach to design the system to enable it to effectively collaborate with users. Such a holistic approach may include new data models (Lucas & Babaian, 2012) and better user action-logging mechanisms, which require much deeper and broader improvement in the system architecture than on the user interface.

As we discuss at the beginning of the paper, however, in this paper, we do not develop another set of usability principles or guidelines (such as those proposed in Gould & Lewis, 1985; Gulliksen et al., 2003; Nielsen, 1994; Norman, 1983) at this stage. Others can use the seven collaborative capabilities that we identify in Section 3 as a foundation for practical design principles; indeed, Babaian, Lucas, Xu, and Topi's (2010) work represents one existing attempt to do so. We hope that researchers can use our descriptions of the model and empirical illustrations as a solid, conceptually justified foundation for developing design principles in the future.

Our conceptual model also paves the way for several other research endeavors that are part of the broader research project to which this study belongs. This project as a whole uses a number of methods to examine the user-system collaboration phenomenon from different perspectives, including studying a range of organizations through cross-case analysis, developing and conducting surveys of user perceptions regarding system collaborativeness, implementing a walkthrough protocol based on collaboration concepts, and developing a proof-of-concept prototype using the collaborative design approach.

Achieving a high level of usability for complex information systems is critically important for both the organizations that build the core of their business around these systems and for the users who often spend a significant portion of their time using these systems. Our research specifically focuses on understanding user-system collaboration and its linkage to usability in order to provide guidance about designing future versions of information systems (e.g., large-scale enterprise systems). We hope that the study we report on in this paper and the broader research project will ultimately lead to more usable information systems.

5.3 Limitations

Our research has several limitations. First, in our field study, we studied only two ERP systems used in two organizations. Other ERP products and configurations of these same products may have different levels of collaborative capabilities from what we found in these two systems. Moreover, although these two organizations represent two different industry sectors (IT and medical device manufacturing), technical, organizational, and cultural characteristics may be unique to the specific settings in the two organizations. As a result, our empirical evidence is limited only to the two ERP systems as configured and used in the two organizations in our study. They may not be generalizable to a wide variety of complex information systems that different organizations use. We do, however, believe that the data we collected serves the purpose for which we used it well—to provide practical examples of collaborative capabilities and their impacts. Moreover, although our research results may not be generalizable based on statistical, sampling-based inferences, we offer a set of logically consistent propositions based on the prior conceptualization of collaborative activity. In this sense, we believe our model has TT (from concept to theory) generalizability (Lee & Baskerville, 2003).

Second, we present examples only about the system behaviors as users perceive them. We do not illustrate how users perceived their own collaborative behaviors and whether such self-perceptions would affect their perceptions of system usability. As we note in Section 2.1, system usability leads to perceptions of "quality of use" (Bevan, 1995), and it highly depends on the context of use, including the user, goal, and environment. Similarly, according to the task-technology fit theory (Goodhue, 1995; Goodhue & Thompson, 1995), the characteristics of both the task and the individual, in addition to those of technology, have an impact on the user evaluations of the system and further on user performance. It is possible that, when a user becomes more collaborative by taking a series of adaptation and learning actions (Burton-Jones & Grange, 2013), the task-technology fit will improve, and the user will perceive the system more positively. Thus, we need future research that investigates the impact of user's own collaborativeness perceptions on system usability.

Third, we note that our research is still at an early stage. The empirical evidence from our qualitative field study offers a limited window to the key collaborative capabilities associated with design characteristics of complex information systems. The interviews only generally describe users' perceptions regarding the system and its design. Although the coders carefully coded and analyzed the interview data and although our on-site observations of real use supported the data, a thorough evaluation of system design documents, interviews with designers, and/or a structured, formal walkthrough of the system would certainly provide additional insights and significant opportunities for future research. In addition, the conceptual model focuses only on user-system interactions at the individual level and does not include a variety of organizational-level factors (e.g., organization policy and culture) that may impact the user-system relationship, designers' intentions, and system behaviors.

6 Conclusion

The complexity of many organizational information systems will continue to scale up as they continue to integrate more business functions. Consequently, companies and organizations that adopt complex information systems must invest more time and effort in implementing and customizing the systems and in training end users to use the systems if their usability remains unsatisfactory. In this paper, we address complex information systems' usability issues from the human-computer collaboration perspective. Based on Bratman's (1992) collaboration theory, we propose a conceptual model that specifies the requirements for successful collaboration between systems and users. Based on this model, we derive seven capabilities for collaborative information systems. As our empirical examples demonstrate, the presence and absence of these collaborative features can have great impact on users' perceptions of system usability in terms of effectiveness, efficiency, and satisfaction. We believe that our research provides a holistic framework for addressing complex information systems' usability in an innovative way and lays the foundation for both future research on collaborative systems and improved design of these systems.

Acknowledgments

This research was supported by the National Science Foundation (Grant No. SRS 0819333). We gratefully acknowledge the origins of the thinking underlying this work in collaboration with our colleagues at Bentley University. We are particularly indebted to Tamara Babaian and Wendy Lucas. The results of this collaboration have been reported, for example, in Babaian, Lucas, and Topi (2007), Babaian et al. (2010), and Cooprider et al. (2010). Cooprider et al. (2010), presented at HICSS 2010, can be considered an early version of this paper. We also thank Anna Karpovsky and Martin Dias for their assistance with the content coding tasks.

References

Agarwal, R., & Venkatesh, V. (2002). Assessing a firm's Web presence: A heuristic evaluation procedure for the measurement of usability. *Information Systems Research*, *13*(2), 168-186.

Al-Mashari, M., Al-Mudimigh, A., & Zairi, M. (2003). Enterprise resource planning: A taxonomy of critical factors. *European Journal of Operational Research*, *146*(2), 352-364.

Albers, M., & Still, B. (Eds.). (2011). *Usability of complex information systems: Evaluation of user interaction*. Boca Raton: Taylor & Francis Group.

Albers, M. J. (2011). Usability of complex information systems. In M. J. Albers & B. Still (Eds.), *Usability of complex information systems: Evaluation of user interaction* (pp. 3-16). Boca Raton, FL: Taylor & Francis Group.

Alonso, F. M. (2009). Shared intention, reliance, and interpersonal obligation. *Ethics*, *119*(3), 444-475.

Amoako-Gyampah, K. (2007). Perceived usefulness, user involvement and behavioral intention: An empirical study of ERP implementation. *Computers in Human Behavior*, *23*(3), 1232-1248.

Babaian, T., Lucas, W., & Topi, H. (2007). Visualizing the process: A graph-based approach to enhancing system-user knowledge sharing. In *Proceedings of the 9th International Conference on Enterprise Information Systems* (pp. 123-128).

Babaian, T., Lucas, W., Xu, J., & Topi, H. (2010). Usability through system-user collaboration: Deriving design principles for greater ERP usability. In *Proceedings of the 5th International Conference on Design Science Research in Information Systems and Technology.*

Benbasat, I., & Barki, H. (2007). Quo vadis TAM. *Journal of the Association for Information Systems*, *8*(4), 211-218.

Bevan, N. (1995). Usability is quality of use. In *Proceedings of 6th International Conference on Human Computer Interaction*.

Botta-Genoulaz, V., Millet, P.-A., & Grabot, B. (2005). A survey on the recent research literature on ERP systems. *Computers in Industry*, *56*(6), 510-522.

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

Bratman, M. E. (1993). Shared intention. *Ethics*, *104*, 97-113.

Bratman, M. E. (1997). I intend that we J. In R. Tuomela & G. Holmstrom-Hintikka (eds.), *Contemporary action theory* (pp. 49-63). Dordrecht: Kluwer.

Bratman, M. E. (1999). *Faces of intention*. Cambridge: Cambridge University Press.

Bratman, M. E., Israel, D. J., & Pollack, M. E. (1988). Plans and resource-bounded practical reasoning. *Computational intelligence*, *4*(3), 349-355.

Bueno, S., & Salmeron, J. L. (2008). TAM-based success modeling in ERP. *Interacting with Computers*, *20*(6), 515-523.

Burton-Jones, A., & Grange, C. (2013). From use to effective use: A representation theory perspective. *Information Systems Research*, *24*(3), 632-658.

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.

Chaudhari, S., & Ghone, A. (2015). *World ERP software market—opportunities and forecasts, 2013–2020.* Allied Market Research.

Cooprider, J., Topi, H., Xu, J., Dias, M., Babaian, T., & Lucas, W. (2010). A collaboration model for ERP user-system interaction. In *Proceedings of Proceedings of the 43rd Hawaii International Conference on System Sciences.*

Davenport, T. H. (1998). Putting the enterprise into the enterprise system. *Harvard Business Review*, *76*(4), 121-131.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, *13*(3), 319-340.

DeLone, W., & McLean, E. (2003). The Delone and Mclean model of information systems success: A ten-year update. *Journal of Management Information Systems*, *19*(4), 9-30.

Eisenhardt, K. M. (1989). Building theories from case study research. *Academy of Management Review*, *14*(4), 532-550.

Goodhue, D. L. (1995). Understanding user evaluations of information systems. *Management Science*, *41*(12), 1827-1844.

Goodhue, D. L., & Thompson, R. L. (1995). Task-technology fit and individual performance. *MIS Quarterly*, *19*(2), 213-236.

Gould, J. D., & Lewis, C. (1985). Designing for usability: Key principles and what designers think. *Communications of the ACM*, *28*(3), 300-311.

Grant, D., Hwang, Y., & Tu, Q. (2013). An empirical investigation of six levels of enterprise resource planning integration," *Computers in Human Behavior*, *29*(6), 2123-2133.

Grossman, J., & Walsh, J. (2004). Avoiding the pitfalls of ERP system implementation. *Information Systems Management*, *21*(2), 38-42.

Grosz, B. J. (1996). Collaborative Systems: AAAI Presidential Address," *AI Magazine* (17:2), pp. 67-85.

Grosz, B. J. (2005). Beyond mice and menus. *Proceedings of the American Philosophical Society*, *149*(4), 529-523.

Grosz, B. J., & Kraus, S. (1996). Collaborative plans for complex group action. *Artificial Intelligence*, *86*(2), 269-357.

Grosz, B. J., & Sidner, C. (1990). Plans for discourse. In P. Cohen, J. Morgan, & M. Pollack (Eds.), *Intentions in communications* (pp. 417-444). Cambridge, MA: MIT Press.

Gulliksen, J., Göransson, B., Boivie, I., Blomkvist, S., Persson, J., & Cajander, A. (2003). Key principles for user-centred systems design. *Behaviour and Information Technology*, *22*(6), 397-409.

Hestermann, C. (2009). Key issues for enterprise resource planning 2009. *Gartner*.

Hornbæk, K. (2006). Current practice in measuring usability: Challenge to usability studies and research. *International Journal of Human-Computer Studies*, *64*, 79-102.

Hsieh, J. J. P., & Wang, W. (2007). Explaining exployee's extended use of complex information systems. *European Journal of Information Systems*, 16, 216-227.

ISO. (1998). *ISO 9241-11: Ergonomic requirements for office work with visual display terminals (VDTs)—part 11: Guidance on usability.*

Jennings, N. R. (1996). Coordination techniques for distributed artificial intelligence. In G. M. P. O'Hare & N. R. Jennings (Eds.), *Foundations of distributed artificial intelligence* (pp. 187-210). New York: John Wiley & Sons.

Kamar, E., Gal, Y., & Grosz, B. J. (2009). Incorporating helpful behavior into collaborative planning. In *Proceedings of the 8th International Joint Conference on Autonomous Agents and Multiagent Systems* (pp. 875-882).

Knoblich, G., Butterfill, S., & Sebanz., N. (2011). Psychological research on joint action: Theory and data. In B. Ross (Ed.), *Psychology of learning and motivation-advances in research and theory* (pp. 59-101). Burlington: Academic Press.

Lambeck, C., Fohrholz, C., Leyh, C., Šūpulniece, I., & Muller, R. (2014a). Commonalities and contrasts: An investigation of ERP usability in a comparative user study. In *Proceedings of the 22nd European Conference on Information Systems.*

Lambeck, C., Muller, R., Fohrholz, C., & Leyh, C. (2014b). (Re-) evaluating user interface aspects in ERP systems: An empirical user study. In *Proceedings of the 47th Hawaii International Conference on System Science* (pp. 396-405).

Lee, A. S., & Baskerville, R. L. (2003). Generalizing generalizability in information systems research. *Information Systems Research*, *14*(3), 221-243.

Lochbaum, K. E. (1998). A collaborative planning model of intentional structure. C*omputational Linguistics*, *24*(4), 525-572.

Lucas, W., & Babaian, T. (2012). Implementing design principles for collaborative ERP systems. In *Proceedings of the 7th International Conference on Design Science Research in Information Systems and Technology* (pp. 88-107).

Maguire, M. (2001). Context of use within usability activities. *International Journal of Human-Computer Studies*, *55*(4), 453-483.

Markus, M. L., Axline, S., Petrie, D., & Tanis, C. (2000). Learning from adopter's experience with ERP: Problems encountered and success achieved. *Journal of Information Technology*, *15*, 245-265.

Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., & Ghalsasi, A. (2011). Cloud computing: The business perspective. *Decision Support Systems*, *51*(1), 176-189.

Matthews, D. (2016). Usability as an ERP selection criteria. Retrieved from http://ifs.datahost.com/shop/images/wp-usability.pdf

Miles, M., & Huberman, A. (1994). *Qualitative data analysis: A sourcebook of new methods.* Beverly Hills, CA: Sage.

Nielsen, J. (1993). *Usability engineering*. Boston, MA: Morgan Kaufman.

Nielsen, J. (1994). Enhancing the explanatory power of usability heuristics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 152-158).

Nonaka, I., & Takeuchi, H. (1995). *The knowledge creating company*. London, UK: Oxford University Press.

Norman, D. (1988). *The design of everyday things*. New York, NY: Doubleday.

Norman, D. A. (1983). Design principles for human-computer interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

O'Farrell, J., Letch, N., & Purchase, S. (2012). ERP usability and the mangle of practice: Improving work processes and employee performance. In *Proceedings of the Eighteenth Americas Conference on Information Systems*.

Oberquelle, H. (1984). On models and modelling in human-computer co-operation. In *Proceedings of 2nd European Conference on Readings on Cognitive Ergonomics: Mind and Computers* (pp. 26-43).

Oberquelle, H., Kupka, I., & Maass, S. (1983). A view of human—machine communication and co-operation. *International Journal of Man-Machine Studies*, *19*(4), 309-333.

Pairat, R., & Jungthirapanich, C. (2005). A chronological review of ERP research: An analysis of ERP inception, evolution, and direction. In *Proceedings of 2005 IEEE International Conference on Engineering Management* (pp. 288-292).

Palmer, J. W. (2002). Web site usability, design, and performance metrics. *Information Systems Research, 13*(2), 151-167.

Panzarasa, P., Jennings, N. R., & Norman, T. J. (2002). Formalizing collaborative decision-making and practical reasoning in multi-agent systems. *Journal of Logic and Computation*, *12*(1), 55-117.

Parks, N. E. (2012). Testing & quantifying ERP usability. In *Proceedings of the 13th Annual Conference on Information Technology Education* Calgary (pp. 31-36).

Pascot, D., Bouslama, F., & Mellouli, S. (2011). Architecturing large integrated complex information systems: An application to healthcare. *Knowledge and Information Systems*, *27*, 115-140.

Pellizzoni, L. (2001). The myth of the best argument: Power, deliberation and reason. *The British Journal of Sociology*, *52*(1), 59-86.

Rich, C., Sidner, C. L., & Lesh, N. (2001). Collagen: Applying collaborative discourse theory to human-computer interaction. *AI magazine*, *22*(4), 15-26.

Rouse, W. B., Geddes, N. D., & Curry, R. E. (1987-1988). An architecture for intelligent interfaces: Outline of an approach to supporting operators of complex systems. *Human-Computer Interaction*, *3*, 87-122.

Sarker, S., & Less, A. S. (2003). Using a case study to test the role of three key social enablers in ERP implementation. *Information & Management*, *40*(8), 813-829.

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

Scott, J. E., & Vessey, I. (2002). Managing risks in enterprise systems implementations. *Communications of the ACM*, *45*(4), 74-81.

Sears, A., & Jacko, J. A. (Eds.). (2008). *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications*. New York City, NY: Lawrence Erlbaum Associates.

Shaul, L., & Tauber, D. (2013). Critical success factors in enterprise resource planning systems. *ACM Computing Surveys*, *45*(4), 1-39.

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

Siau, K. (2004). Enterprise resource planning (ERP) implementation methodologies. *Journal of Database Management*, *15*(1), 1-6.

Silverman, B. G. (1992). Human-computer collaboration. *Human-Computer Interaction*, *7*, 165-196.

Singh, A., & Wesson, J. (2009). Evaluation criteria for assessing the usability of ERP systems. In *Proceedings of South African Institute of Computer Scientists and Information Technology*.

Snider, B., da Silveira, G. J. C., & Balakrishnan, J. (2009). ERP implementation at SMEs: Analysis of five Canadian cases. *International Journal of Operations & Production Management*, *29*(1), 4-29.

Terveen, L. (1995). An overview of human-computer collaboration. *Knowledge-Based Systems*, *8*(2-3), 67-81.

Tomasello, M. (2006). Why don't apes point? In N. J. Enfield & S. C. Levinson (Eds.), *Roots of human sociality: Culture, cognition and interaction* (pp.506-524). New York: Berg.

Topi, H., Lucas, W., & Babaian, T. (2005). Identifying usability issues with an ERP implementation. In *Proceedings of the 7th International Conference on Enterprise Information Systems*.

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.

Tretten, P., & Karim, R. (2014). Enchancing the usability of maintenance data management systems. *Journal of Quality in Maintenance Engineering*, *20*(3), 290-303.

Venkatesh, V., & Agarwal, R. (2006). Turning visitors into customers: A usability-centric perspective on purchase behavior in electronic channels. *Management Science*, *52*(3), 367-382.

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. *MIS Quarterly*, *27*(3), 425-478.

About the Authors

Jennifer J. Xu is an Associate Professor of Computer Information Systems at Bentley University. Her research interests include business intelligence, data mining and knowledge discovery, social network analysis, human-computer interaction, and enterprise systems, with a particular interest in developing and applying analytics methods and approaches in a variety of domains including business, accounting and finance, healthcare, and security. She has published more than 60 papers in Information Systems journals, books, and conference proceedings. She is currently serving on the editorial boards of *Communications of the Association for Information Systems* and *Journal of Security Informatics*. She received her PhD degree in Management Information Systems from the University of Arizona.

Heikki Topi is Professor of Computer Information Systems at Bentley University. His research focuses on systems development methodologies, information systems education, and human factors and usability in the context of enterprise systems. His research has been published in journals such as *European Journal of Information* Systems, *JASIST*, *Information Processing & Management*, *International Journal of Human-Computer Studies*, *Journal of Database Management*, and others. He is co-author of *Modern Database Management*, co-editor of *IS Management Handbook* and *Computing Handbook: Information Systems and Information Technology,* and co-editor of *Education Department of CAIS*. He has been actively involved in global computing curriculum development and evaluation efforts since early 2000s (including *IS 2002*, *CC2005 Overview Report,* and as task force co-chair of *IS 2010* and *MSIS 2016*, the latest IS curriculum revisions). He currently serves as AIS VP of Education. He received his PhD in Management Information Systems from Indiana University.