Communications of the Association for Information Systems

INTEGRATING HUMAN-COMPUTER INTERACTION DEVELOPMENT INTO THE SYSTEMS DEVELOPMENT LIFE CYCLE: A METHODOLOGY

Ping Zhang
Syracuse University
pzhang@syr.edu

Jane Carey
Arizona State University West

Dov Te'eni
Tel-Aviv University

Marilyn Tremaine
New Jersey Institute of Technology

ABSTRACT

Incorporating a human computer interaction (HCI) perspective into the systems development life cycle (SDLC) is necessary to information systems (IS) success and, in turn, to the success of businesses. However, modern SDLC models are based more on organizational needs than human needs. The human interaction aspect of an information system is usually considered far too little (only the screen interface) and far too late in the IS development process (only at the design stage). Thus, often a gap exists between satisfying organizational needs and supporting and enriching human users. This problem can be addressed by carefully integrating HCI development into the SDLC process to achieve a truly human-centered IS development approach. This paper examines the roles of HCI in systems development, justifies the importance of considering HCI through out the entire systems development life cycle, presents a methodology for human-centered IS development, and demonstrates how to apply this methodology to develop human-centered information systems.

Keywords: systems development life cycle (SDLC), human-computer interaction (HCI), human factors in information systems (HFIS), information systems development methodology, user-centered design, human-centered systems development, information systems

I. INTRODUCTION

In his AMCIS 2003 keynote speech on "The Future of the Internet," Patrick listed numerous frustrations and difficulties of using corporate websites from a consumer's perspective [Patrick, 2003], indicating the significance and importance of human computer interaction considerations for business applications in today's world. Patrick's call for an emphasis on the usefulness and usability of information systems from the perspective of the consumer is just the most recent in a long line of such suggestions. In various degrees, the software shapes work organization, job content, job design, and decision latitude [Clegg et al., 1997; Eason, 1997]. Therefore, incorporating a human computer interaction perspective into the systems development life cycle (SDLC) is critical to information system (IS) success and in turn to the success of organizations and businesses. As early as the first volume of *MIS Quarterly*, Bostrom and Heinen [1977] suggest that information systems failures could be attributed to “faulty design choices” (p.17) resulting from the lack of emphasis on the human/social aspects of system use. The Technology Acceptance Model (TAM) demonstrates the importance of both the perceived usefulness and the perceived ease of use on user acceptance of IS [Davis, 1989; Venkatesh et al., 2003]. In addition, as consumers handle more of their own services, HCI becomes increasingly critical to business success [Carey et al., 2004]. Despite the importance of considering humans and their interactions with the computer systems, modern SA&D approaches consider the human computer interaction aspect too little and too late in the systems development process. In practice of systems development, there is still a lack of attention to the HCI issues, yielding frustrating software systems that control the work pace and task order, leave users little or no control over their work or tasks, and increase the users’ cognitive workload and mental stress [Boivie et al., 2003; Patrick, 2003].

This paper provides both the ‘why’ and the ‘how’ of building HCI into the systems development process. In this paper, we first examine some misperceptions about HCI that may have contributed to its current roles in systems development. This discussion is further augmented by a brief examination of several popular Systems Analysis and Design (SA&D) textbooks, in which HCI issues, if considered at all, are covered with too little information and too late in the systems development process. Next, we introduce several important concepts and discuss multiple concerns or goals of human interaction with technologies. Building on the multiple HCI concerns, we propose a Human-Centered Systems Development Life Cycle (HCSDLC) model for developing information systems that considers both organizational and human needs thus streamlines the modern SA&D and HCI approaches. The HCSDLC methodology emphasizes the systematic and theory-based application and operationalization of human-centeredness during all stages of SDLC. A philosophy and a set of strategies are laid out, along with activities and methods for each of the main stages of the HCSDLC model. Our goal is that the methodology should be instrumental for developing information systems that meet both organizational and human needs because the ultimate concern of humans interacting with technology is for supporting human holistic experiences with technology for life enrichment and personal goals (job related or others). The term ‘human-centered systems development’ includes both basic user-centered systems functionalities and encompassing human-centered human-computer interaction development. Because of the maturity of modern SA&D approaches and limited space in this paper, we further limit our focus on the HCI development part of the HCSDLC methodology and refer to the modern SA&D counterparts when necessary. An e-Commerce website development is used as an example to illustrate the step-by-step procedure of applying the methodology.

II. INFORMATION SYSTEMS DEVELOPMENT AND HCI

INFORMATION SYSTEMS DEVELOPMENT METHODOLOGIES

The development of computer-based information systems began in the 1950s. It went through several revolutionary stages owing to the advancement of technological capabilities of computers and the organization’s IT needs. Revolutionary advancements in systems development include the structured approach in the 1970s, the object-oriented approach in the 1980s, and current agile approaches [Fowler and Highsmith, 2001] such as eXtreme Programming [Beck, 2000] and short cycle time systems development [Baskerville and Pries-Heje, 2004]. An information systems development methodology (ISDM) is a collection of particular systems development assumptions, a set of strategies, principles and guidelines, a multi-step procedure of what to do and how to do things, and associated techniques and methods.

Hirschheim and Klein [1989] argued that developing information systems necessarily involves implicit and explicit assumptions that affect not only the development processes but also the developed systems. They applied Burrell and Morgan's [1979] four paradigms or fundamental sets of assumptions to information systems development approaches. These four paradigms are based on knowledge acquisition methods (epistemological assumptions) and worldviews regarding society and technology (ontological assumptions). The two dimensions of knowledge are subjectivist-objectivist and the two dimensions of worldviews are order-conflict [Burrell and Morgan, 1979]. The objectivist applies methods and models, derived from the natural world to the study of human-based systems and treats the social world as if it were the study of the natural world. Conversely, the subjectivist seeks to understand human life by exploring the subjective experiences of individuals. In the order-conflict dimension, the order view sees the social world as one of stable order and functional coordination. The conflict view sees the social world as one of constant change, conflict and disintegration. These four dimensions form a two by two matrix, which yield the four paradigms: functionalism (objective-order); social relativism (subjective-order); radical structuralism (objective-conflict); and neohumanism (subjective-conflict). The functional paradigm seeks to provide explanations of the status quo and is based on rational choice and a belief in an integrated whole. The social relativist paradigm looks inward to individual consciousness and social roles to discover meaning and understanding. The radical structuralist paradigm sees the status quo as something that needs to be transcended or abolished. The main focus of analysis is on economic power relationships. The neohumanist paradigm seeks radical change by overcoming social constraints.

It is worth noting that Hirschheim and Klein's topology of ISDM paradigms is not without problems or criticism. One aspect is that it is not parsimonious and sometimes hard to apply to categorize existing approaches. Fortunately, these issues are not a problem in applying the theory to support our arguments. We think that Hirschheim and Klein's topology is a useful tool simply to set the perspective that many of the concerns of HCI have been addressed before in some form or other.

Applying this topology, we found that, in practice, information systems development approaches are influenced by assumptions from more than one paradigm, although the influence from one paradigm is typically dominant [Hirschheim and Klein, 1989]. For example, the traditional structured approach is within the functionalist paradigm. The modern structured approaches, as covered by the popular textbooks on systems analysis and design (Table 1), are influenced by more than just the functionalist paradigm and with an emphasis on the subjectivity and evolutionary nature of requirements by using prototyping, joint application development (JAD), and other techniques. More recently, several methodologies were developed to address the pressures from short time development and chaotic conditions. These methodologies include Scrum [Rising and Janoff, 2000], eXtreme programming [Beck, 2000], amethodical systems development [Truex et al., 2000], and short cycle time system development [Baskerville and Pries-Heje, 2004].

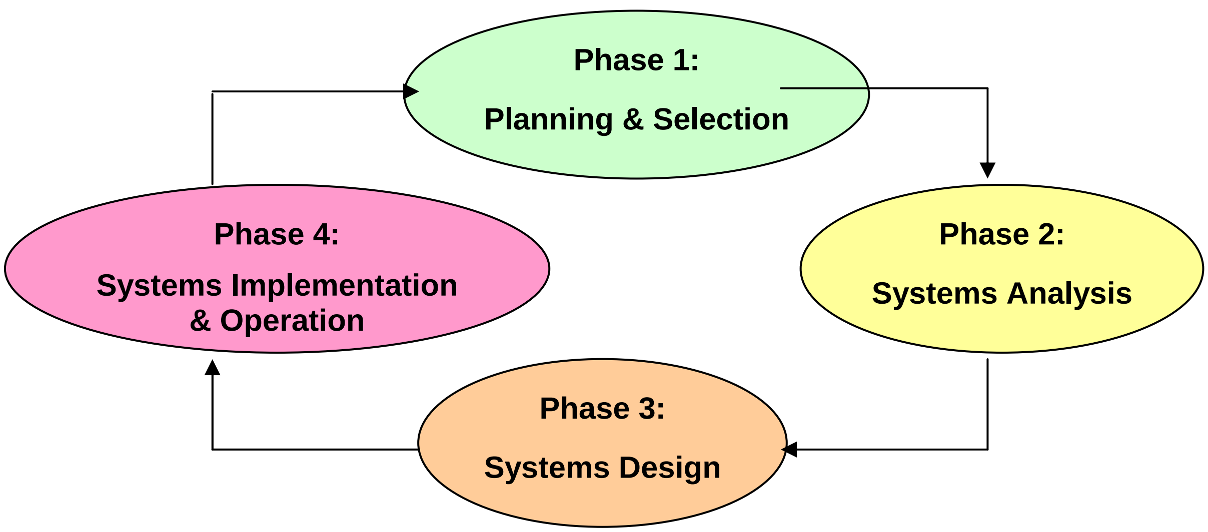
Various attempts to integrate human or user aspects into systems development have been proposed and some were adopted by industry with varying degrees of success. The more prominent approaches include the ETHICS methodology [Mumford, 1983], soft systems methodology [Checkland and Scholes, 1990], the Scandinavian approach [Bjerknes et al., 1987], and the approach of understanding human cognition in developing computer systems [Winograd and Flores, 1986]. The neohumanism approaches attempt to improve human understanding and the rationality of human action through emancipation of suppressed interests and liberation from unwarranted natural and social constraints [Hirschheim and Klein, 1989]. From this perspective, human or user aspects, especially the political, power or social aspects, are considered to some extent in the systems development processes.

Despite the variety of systems development approaches that are in practice, in classrooms, we teach students the basic components and techniques that function as the building blocks or ingredients of various systems development methodologies. For example, the basic considerations and techniques for analysis, design and coding can be used in many systems development methodologies, from the traditional waterfall model to the modern structured approach, and to the eXtreme programming or other current approaches. From a training perspective, the use of building blocks is appropriate because, unless the fundamentals are learned, developers do not have the "generational experience" or be able to develop a more abstract approach [Baskerville and Pries-Heje, 2004] to address various systems development situations.

Due to this consideration, in this paper we examine and then discuss how to build HCI considerations into the fundamentals of systems development approaches. Instead of considering a number of different systems development approaches, we consider a well covered one in the classrooms and textbooks (that is, the modern structured approach), and illustrate the issues and techniques of integrating HCI into this approach.

Among many systems development approaches, the systems development life cycle (SDLC) model is a commonly accepted modern structured approach for describing the complex processes and issues involved in information systems development. It captures the spirit of the systems development process [Hoffer et al., 2005] and is a general framework that can be found in many different systems development methodologies. Because of this nature, it dominates the current popular textbooks on SA&D (Table 1).

Figure 1 depicts one version of this model [Valacich et al., 2004]. Four phases are identified: Project Planning and Selection, Systems Analysis, Systems Design, and Systems Implementation & Operation.



From: Valacich, George, Hoffer [2004]

Figure 1. Modern SDLC

The directional relationships among phases in Figure 1 are for high-level project management purposes. Iterations among stages are typical in real IS development projects. Figure 2 reflects the key ideas of the modern SA&D approach: iteration, fast feedback (such as developing prototypes and soliciting user feedback), accuracy, and user-centeredness. The key point is that, because they are used for project management, the four phases in Figure 1 are at a higher level. Each phase could include multiple rounds of smaller scales of analysis, design, and implementation. For example, during Phase 1 project selection and initiation/planning, analysts may need to do a quick mock-up or prototype to have some concrete ideas of system functionalities and gain feedback from users and on market potential (such as user acceptance or needs tests on mockups/prototypes). In Phase 2 the analysis stage, system requirements can be specified in more details with a prototype of the system for both analysts and users to gain accurate understanding of the system functions. In Phase 3 the design stage, certain design options or results from formative evaluations may prompt a re-analysis of certain aspects, and such rework may better be demonstrated by another round of prototyping. All these examples

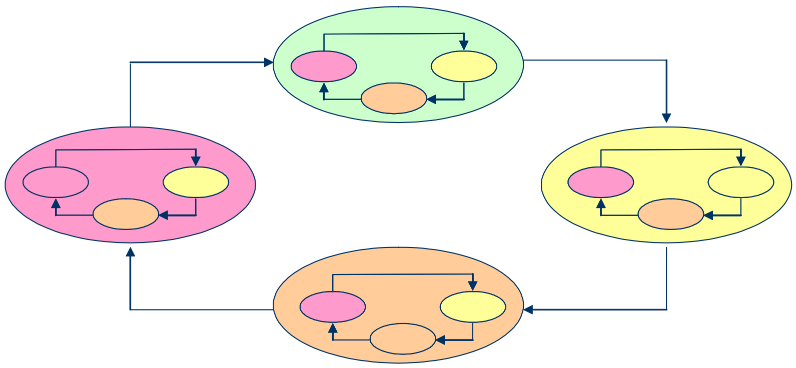


Figure 2. Modern SDLC: Iteration, Fast Feedback, Accuracy, and User-Centered

indicate that each stage may include smaller scale analysis, design and implementation activities within it. Figure 2 is an attempt to illustrate this iterative idea that is embedded in the modern SDLC model. The specific activities inside each phase of the SDLC model will be explained later when the modern SDLC is presented in contrast to the proposed HCSDLC model.

REALITY OF HCI IN SYSTEMS DEVELOPMENT

It is well realized that although usability engineering is making headway in industry, HCI exerted only a minor influence on the current generations of object-oriented development methods. While HCI created structured methods from both academic research and industrial authors, these ideas were largely ignored by software engineers [Sutcliffe, 2000]. Argument for an engineering approach to HCI that complemented and integrated with software engineering proved elusive [Sutcliffe, 2000].

In many systems development approaches including the recent ones, HCI issues and concerns, if ever covered, are not considered systematically. Attempts were made in the past to tie usability and user factors into the systems development life cycle [Hefley et al., 1995; Mantei and Teorey, 1989]. Still, we as educators and researchers did not provide a clear methodology for integrating HCI into the systems analysis and design processes. Such a methodology can help us to prepare our students to develop truly human-centered organizational information systems that benefit the human users and contribute to successful organizations and businesses. This lack of integration is reflected in many popular modern SA&D textbooks that contain only some chapters in the design stage of SDLC on some user interface issues. Table 1 is a summary of several most recent SA&D textbooks that are written by MIS scholars and are often adopted by MIS programs. Compared to the textbooks several years ago, these books all cover user interface issues to some extent. They demonstrate the realization that user interfaces are important in the success of information systems. Yet, as we will discuss later in the paper, HCI issues are covered too little and too late in these books.

The lack of HCI considerations in modern SA&D are related to some major misperceptions of what HCI is and what its roles are in systems development. One misperception is that HCI is only about the final user interface design, such as form design, menu layout, colors, icon design, and screen layout of display interfaces. This view coincides with the coverage in popular SA&D textbooks, as shown in Table 1. Undeniably, screen layout, menu design, buttons and colors and other interface features are HCI considerations in information systems development. But they are far from being exhaustive or even the most important ones. Very often, users of an information system are most frustrated or annoyed by problems that are beyond the computer screen level. Illogical organization of data/information in the system, lack of task support, misfit between the nature of the task and the support provided, lack of control over the system, difficulty in

Table 1. HCI Issues Covered in Popular SA&D Textbooks

Book	**Key Methodology/Approach**	**HCI Issues Covered**
[Dennis and Wixom, 2003]	4-stage life cycle model	One chapter in the Design stage on User Interface Design
[Hoffer et al., 2005]	5-stage life cycle model (with some coverage on OO and Agile/eXtreme programming approaches)	Two chapters in the Design stage on Designing Forms and Reports, and Designing Interfaces and Dialogues
[Kendall and Kendall, 2005]	4-stage life cycle model (with some coverage on OO and eXtreme Programming)	Four chapters in the Design stage on Input, Output, User Interface, and Data-Entry Procedure.
[Satzinger et al., 2004]	3-stage life cycle model (analysis, design and implementation, with coverage on OO)	One chapter in the Design stage on User Interface
[Valacich et al., 2004]	4-stage life cycle model (with some coverage on OO)	One chapter in the Design stage on Human Interface
[Whitten et al., 2004]	A life cycle model that supports multi-goals (knowledge, process, and communication) and has multi-views	The communication goal includes some HCI concerns that run through the entire lifecycle

navigation, and inconsistency between mental models and system operations are among the major problems or difficulties. These incompatibilities affect user reactions to, acceptance of, and effective use of the information system. These problems may be rooted in the neglect of complex human cognitive, affective, and behavioral factors and the dynamics of human interactions with technologies. These issues can be addressed during HCI development processes. A better understanding of various human ergonomic, cognitive, affective, and behavioral factors involved in user tasks, problem solving processes and interaction contexts is required to address these problems. Just as it is important to understand systems requirements as early as possible, it is important that human technology interaction should be addressed at the beginning and throughout the entire process of SDLC.

Another erroneous perception is that HCI is only about usability. Although usability has been a dominant part of the HCI field, many empirical studies on user technology acceptance prove that usability is neither the only nor the most important predictor of system acceptance and usage [Davis, 1989; Venkatesh et al., 2003]. Recent research and practice in IS, HCI, and other related disciplines go beyond usability and explore other factors affecting human interactions with technologies. User's affective reactions and their holistic experiences with technology are gaining more attention and becoming more important [Agarwal and Karahanna, 2000; Webster and Martocchio, 1993; Zhang et al., 2002]. This shift from a basic user-centered to a much richer human-centered perspective requires more understanding about humans and their interactions with tasks and technologies.

The misperceptions sometimes come from unclear or conflicting definitions of some key concepts. To facilitate understanding and discussion for the rest of the paper, we define the following concepts: user interface, human computer interface, usefulness, utility, usability, usability engineering, and human computer interaction.

IMPORTANT CONCEPTS

User Interface

User interface, or human computer interface, is an evolving concept [Grudin, 1993] and different people define it differently for different purposes. For the sake of discussion in this paper, we define User Interface or Human Computer Interface as an aspect of a computer that enables communications and interactions between humans and the computer. It is the layer of the computer that is between humans and the computer. It is not people's emotional response to computers such as anxiety, and it is not a user's physical movement such as moving or clicking a mouse.

Usefulness, Utility, and Usability

Usefulness has different meanings in different contexts. Nielsen defined usefulness of a computer system as the issue of whether the system can be used by users to achieve some desired goals [Nielsen, 1993]. It can be broken down into two categories: utility and usability [Grudin, 1992; Nielsen, 1993]. Utility is the question of whether the functionality of the system in principle can do what is needed. This utility idea is similar to the concept of usefulness in many technology acceptance studies in the IS discipline [Davis, 1989; Venkatesh and Davis, 2000; Venkatesh et al., 2003]. It is about the functions provided by a computer system that support a user's tasks or goals. To avoid terminology confusions, unless otherwise noted, we will use the term "usefulness" to mean the functionality of the system. Usefulness is a HCI concern because users will not use or interact with a system if it does not provide useful functions. Many technology acceptance studies found that perceived usefulness (that is, perceived utility) of a system is the most dominant factor for system acceptance and adoption [Davis, 1989; Venkatesh and Davis, 2000; Venkatesh et al., 2003].

ISO (International Standards Organization) defines usability as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" [ISO, in Bevan, 2001]. Usability is considered as part of system acceptability and is about how well users can use the functionality or utility of a system [Nielsen, 1993]. Usability is not a single, one-dimensional property of a system but contains multiple components. It is traditionally associated with five attributes: learnability, efficiency, memorability, errors, and satisfaction [Nielsen, 1993].

Usability Engineering

Usability Engineering is a process through which usability characteristics are specified, quantitatively and early in the development process, and measured throughout the process [Hix and Hartson, 1993]. Usability engineering is a set of activities that ideally take place throughout the lifecycle of the product, with significant activities happening at the early stages before the user interface is ever designed [Nielsen, 1993]. Usability engineering is a major movement in industry. Discount usability engineering and usability engineering lifecycle models guide interactive systems development practice [Mayhew, 1999; Nielsen, 1993]. As pointed out earlier, these models and practice do not seem to influence the software engineering camp. A gap exists between these two fields of practice.

Human-Computer Interaction

As many definitions for Human-Computer Interaction may exist as for user interface. Because of the advances of technology development and use, human-computer interaction (as a discipline and as a concept) is also an evolving target. During the HCI curriculum development process sponsored by the ACM special interest group on Computer-Human Interaction, Hewett and colleagues defined Human-Computer Interaction as a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them [Hewett et al., 1992]. Preece and colleagues stated that Human-Computer Interaction is about designing computer systems that support people so that they can carry out their activities productively and safely [Preece et al., 1994]. In their book, *Interaction Design: Beyond Human-Computer Interaction*, Preece and colleagues consider that interaction design differs from human-computer interaction because "(a book on interaction design) is concerned with a broader scope of issues, topics, and paradigms than has traditionally been the scope of human-computer interaction." Interaction design is "designing interactive products to support people in their everyday and working lives." [Preece et al., 2002]

Realizing the importance of contexts in the interaction between humans and technologies, Zhang and colleagues defined Human-Computer Interaction, especially human factors in IS, as the ways humans interact with information, technologies, and tasks within various contexts [Zhang et al., 2002]. This last definition emphasizes the point that HCI issues and concerns involve all possible interactions between a user and a system during its lifecycle, including the development stage, use in context, and the impact of such use on individuals, organizations, society, and future systems development. We use this last definition for our further discussions in this paper.

MULTIPLE CONCERNS OF HUMAN-COMPUTER INTERACTION

The recent development and work on HCI from several related disciplines call for a re-examination of the fundamental goals or concerns of human interaction with technologies. For example, Zhang and colleagues suggested revisiting Maslow's basic need hierarchy to ask what humans want or what they need in their lives, then to use technologies to support humans' higher needs in the needs hierarchy [Zhang et al., 2002]. From a slightly different approach, Maxwell also suggested using Maslow's needs hierarchy as an analog for an HCI maturity model to represent a progression in the types of human needs and goals that the HCI discipline supports [Maxwell, 2002]. Specifically, from the perspective that HCI is primarily a discipline focused on people, Maxwell identified three levels of HCI maturity:

- Level 1 is basic usability;
- Level 2 collaborative, organizational and role-based interaction; and
- Level 3 individualized and holistic interaction [Maxwell, 2002].

Overall, the message is that human interaction with technologies should be driven by human's different levels of needs and goals. Thus HCI can be viewed as a progression moving from supporting the basic needs and goals of users toward supporting higher-level human needs and goals with technologies [Maxwell, 2002; Zhang et al., 2002]. All these concerns go beyond the rather traditional usability concerns of HCI, and call for a true human-centered view that takes a holistic angle to examine the concerns and goals of HCI [Agarwal and Karahanna, 2000; Preece et al., 2002; Zhang et al., 2002; Zhang and Li, Forthcoming].

The fundamental difference for this new approach is that the concern is human-centered or human-oriented rather than task or technology oriented. The focus of individual interaction with everything in his or her environment, including IT, is on the potential impact of such interaction on the individual's well-being and personal growth [Maxwell, 2002; Zhang et al., 2002], or the human's self-actualization [Maslow, 1962]. This shift from a basic user-centered to an encompassing human-centered perspective prompts for great attention to identify and address individual factors that include physical and cognitive capabilities, emotional needs, personality traits, and situational factors. This shift is also largely contributed by the advancement of the computing environment that is increasingly ubiquitous, invisible, embedded, tangible, virtual, active, integrated, interconnected, interoperable, and mobile [Maxwell, 2002].

Human interaction with technology is goal-oriented behavior that constitutes two main questions: what causes users to use technology, and why the use of technology is different. These questions fall in the general area of modern motivation studies which attempt to answer two questions: what causes behavior? and why does behavior vary in its intensity? [Reeve, 2005]. Reeve suggests four sources of motivation: external events, needs, cognitions, and emotions. The latter three form internal motives that are internal processes that energize and direct behavior.

- External events are environmental incentives that can energize and direct behavior.
- Needs (biological and psychological) are conditions within the individual that are essential and necessary for the maintenance of life and for the nurturance of growth and well-being.
- Cognitions refer to mental events, such as beliefs, expectations, and self-concept. Cognitive sources of motivation revolve around the person's ways of thinking.
- Emotions are short-lived subjective-physiological-functional-expressive phenomena that orchestrate how we react adaptively to the important events in our lives [Reeve, 2005].

The efforts and results in related disciplines (such as human factors, ergonomics, HCI, and MIS disciplines in studying human interaction with technologies) can be examined within the motivation framework using the human-centered view. For example, ergonomics or human factors studies consider the physical aspect of human interacting with devices including computers. The key issue is to design systems to achieve physical fit between human and machines. Fit is based on the understanding of human physical constraints, limitations, and potentials. Most concerns are studied around human sensors that interact with computers. For example, eyes should not become discomfortable due to color uses or brightness of the computer screen displays; audio signals should be within the comfortable range of our normal hearing; muscle should not be hurt due to the operational demand the system imposes on its users, and people with certain disabilities should be considered properly in system designs. In general, the system should be safe for our health.

Cognitive psychology plays an important role in HCI, primarily on the cognitive side. Perception, memory, mental models and metaphors, knowledge representations, problem solving, errors and learning are all topics under cognitive psychology that have direct implications to HCI design. Usability engineering is largely built on cognitive psychology studies and applications in practice. Usability, or basic usability in Maxwell's term [Maxwell, 2002], includes aspects such as ease of use, ease of learning, error protection, error recovery, efficiency of performance, those that are discussed earlier. Usability involves a strong cognitive component in that users need to comprehend the system to some extent in order to utilize it. Basic usability considerations are continuously needed for any systems to be used by humans. These low-level or basic qualities of the system can be considered to be necessary or hygiene factors [Herzberg, 1966; Zhang et al., 2000]. This usability level of concerns is most mature owing to more than two decades work, especially in usability engineering.

A significant movement in the psychology discipline in recent decades is that the affective or emotional aspect is moving to the mainstream of psychology [Forgas, 1995; Russell, 2003] with the realization that a realistic human has more than just the physical and cognitive aspects. This realization is also reflected in studies in HCI [Brave and Nass, 2003] and in MIS [Sun and Zhang, 2005]. Beautiful things are easy to use [Tractinsky et al., 2000], pleasant things work better [Norman, 2004], and fun things make time fly [Agarwal and Karahanna, 2000]. These things meet our emotional needs.

In the MIS discipline, it is well understood that some extrinsic motivation, such as usefulness of IT, plays an important role in user's IT behavior [Davis, 1989; Venkatesh et al., 2003].

Usefulness, or utility in Grudin and Nielsen's term [Grudin, 1992; Nielsen, 1993], is an HCI concern. Users interact with, adopt and use technologies largely because they perceive that the technology can be used to achieve some desired goals they have. Technology should extend their capabilities, be physical, cognitive, emotional, or behavioral, and allow them to do things and in certain ways that they could not do otherwise. In other words, no matter how easy or how attractive an IT may appear to the potential users, few people will use it if its functions are not perceived to be useful to help fulfill some needs or goals.

While the above discussions all deal with the direct layer between human and IT, another external event can play an important role in the ways humans behave around IT. This event is the sociological, organizational, and cultural impact of computing [Maxwell, 2002]. In other words, the organizational, social and cultural context in which humans interact with IT. This context is largely the result of the broad adoption of IT by organizations and society to support organizational functions and goals and to enhance society's development. For example, organizational efficiency may be expected due to redesign of workflows among critical business units that is affected by the implemented IT; satisfaction and retention of customers/clients are anticipated due to accurate and fast information gathering and presentations, to name a few. It is noteworthy that some of the organizational or societal impacts may not be tangible or directly attributed to HCI considerations. This assertion is in line with the issues of determining IT values in organizations and societies. To make HCI concerns clear to the students and designers to guide their practice, in our methodology, we consider the direct layer concerns between human and IT.

While each of these HCI concerns may have its own importance in different situations in relation to human motivation, it would be helpful for students and designers to see an overview picture of the potential HCI concerns and goals. The purpose of this picture is not to force every IT to be compliant with all the HCI concerns, but to provide an overall framework so that designers can use it as a roadmap and to apply it according to different situations.

Table 2 is a list that considers HCI concerns. These concerns are clustered into four groups: physical, cognitive, emotional (including affective and intrinsically motivational), and extrinsically motivational aspects. These clusters can be relatively easy for designers to map these HCI concerns to some existing measures and concerns (in brackets in the table) such as usability standards. To illustrate each aspect, some items are listed as sample measures. These measures will be realized in the HCI development process, which will be discussed in Section V. A human's holistic experience with technology depends on satisfying these concerns.

These clusters of HCI concerns may depend on each other or influence one another. For example, usability is less relevant if usefulness issue is not resolved: as stated before, few people would use some technology that is easy to use but useless. Pleasant interface may make IT function better only if the usability part is not a concern. It is also noteworthy that certain concerns are more important to some type of IT than to others. For example, for an ERP system being used by an organization, aesthetically pleasing may be less important than for a touring information system at an airport that tries to attract attention of tourists passing by.

When designing HCI, certain design elements can address more than one type of concern. For example, color selection and combination can be of concern for not causing eye discomfort. They can also address the affective and emotional concerns by increasing the aesthetic value of the interface.

Multiple HCI concerns guide the development of our Human-Centered Systems Development Life Cycle (HCSDLC) model (Section V), including various activities and processes. The multiple concerns are particularly relevant to the development of HCI evaluation metrics, to be discussed later. Table 3 lists HCI concerns and some of the ways they can be measured.

IV. MODERN SA&D AND HCI DEVELOPMENT: DIFFERENT EMPHASES

In the development of organizational information systems, the modern SA&D approach focuses on system functionalities and data requirements to meet organizational needs. For example, Hoffer and colleagues consider information systems analysis and design as a complex, challenging, and stimulating organizational process that a team of business and systems professionals uses to develop and maintain computer-based information systems [Hoffer et al., 2005]. Modern SA&D and HCI overlap with the concerns of system utility or functionality (that is, usefulness), although their approaches are different.

The HCI approach focuses on human-machine interactions and collaborations, and defines what a system should do from a user's perspective. It considers user's constraints (physical, cognitive, affective and behavioral) and their impacts on system development and use. HCI development

Table 3. HCI Concerns

HCI Concern	Description	Sample Measure Items
Physical (ergonomic)	System fits our physical strengths and limitations and does not cause harm to our health.	Legible Audible Safe to use
Cognitive (usability)	System fits our cognitive strengths and limitations and functions as the cognitive extension of our brain.	Fewer errors and easy recovery Easy to use Easy to remember how to use Easy to learn
Affective, Emotional, and Intrinsically Motivational (pleasing and enjoyable)	System satisfies our aesthetic and emotional needs, and is attractive for its own sake.	Aesthetically pleasing Engaging Trustworthy Satisfying Enjoyable Entertaining Fun
Extrinsically Motivational (usefulness)	Using the system would provide rewording consequences	Support individual's tasks Can do some tasks that would not so without the system Extend one's capability Rewarding

distinguishes between the user's responsibilities and the system's responsibilities during user interaction with the system and how users can interact with the system. Ultimately, HCI is concerned with how systems can fit with user's needs, lifestyles, well being, and other concerns. To develop information systems to meet both organizational and individual needs, modern SA&D concerns and HCI concerns should be integrated in a unified methodology for information systems development.

V. A HUMAN-CENTERED SYSTEMS DEVELOPMENT METHODOLOGY: HCSDLC

We limit our methodology to developing organizational information systems, which is similar to many modern SA&D textbooks. Figure 3 shows the proposed methodology in contrast to the modern SA&D methodology: the left side (a) is a typical SDLC model while (b) is the HCSDLC model that covers both SA&D and HCI concerns and activities. Note that on the (a) side, user interface design is one task inside the design stage and is typically covered as one or two chapters in a modern SA&D textbook for a one-semester course. Modern SDLC and some systems development methods, such as RAD, JAD, and prototyping, attempt to capture systems requirements (that is, systems functionalities) as early and accurately as possible. These methods, however, are not typically used to capture HCI factors that affect user interaction designs.

The vertical line in the middle of the (b) HCSDLC side of Figure 3 roughly divides the different emphasis of modern SA&D and the HCI development. The four boxes that run across by the vertical line, Project Selection/Project Planning, Requirements Determination, Alternative Selection, and Prototyping are about the same activities that occur in both SA&D and HCI development. Note that for the SA&D side of (b), user interface design activity is removed and should be replaced by the entire HCI side of (b). HCI development thus involves all phases of the SDLC. The HCSDLC methodology indicates that a successful development of an information system should consider all the activities as depicted in (b).

Our philosophy for HCSDLC is that information systems development should meet both organizational and individual's needs. Several strategies under this philosophy are:

1. Early focus on users and their tasks (at the beginning of SDLC)

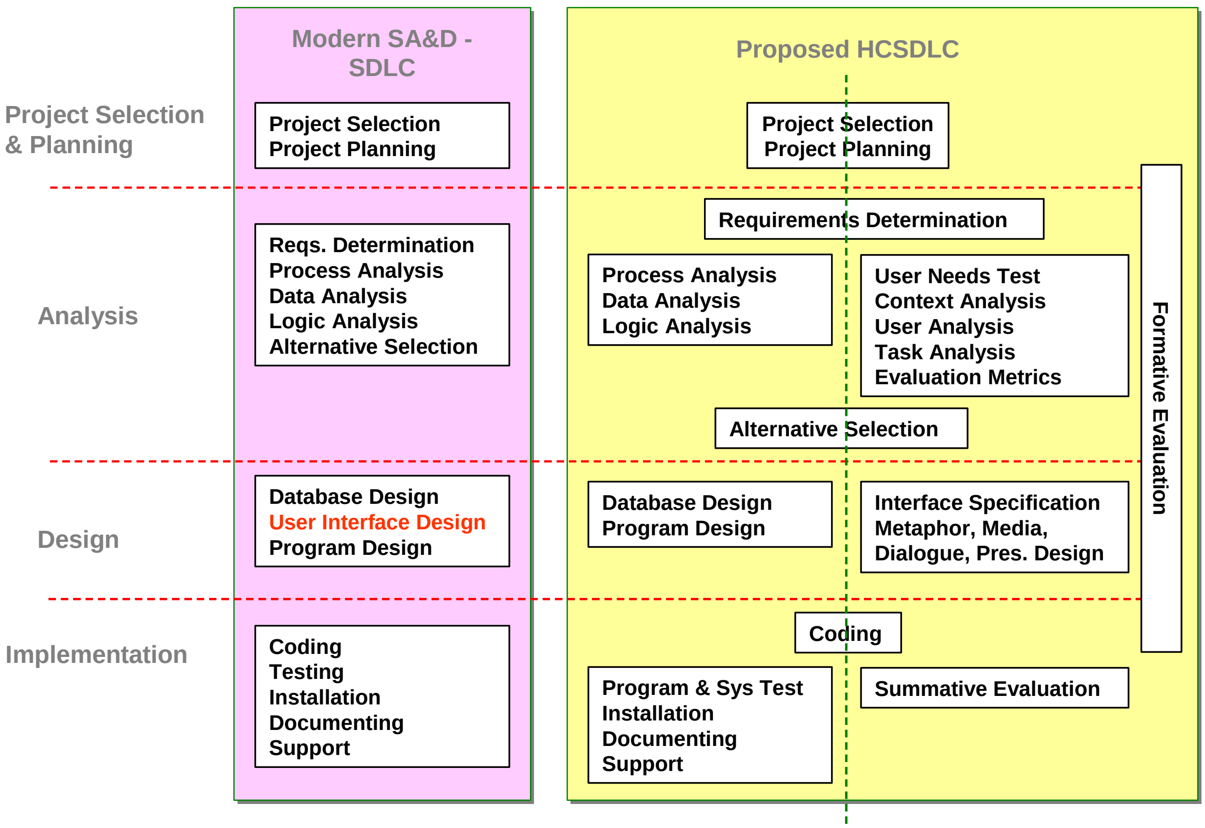


Figure 3. Modern SDLC vs. Proposed Human-Centered Systems Development Lifecycle Methodology

2. Parallel HCI development with modern SA&D activities
3. Evaluations through out the entire system development process
4. Iterative process
5. Consider all four types of HCI concerns

Figure 4 focuses on the HCI part of the HCSDLC methodology model. The main activities in HCI analysis, design and implementation, are guided by HCI principles and guidelines. Activities in each of the four SDLC phases are discussed below in detail. Like modern SA&D shown in Figure 2, the HCSDLC is iterative in nature. Thus each of the four phases may involve multiple iterations of the smaller scale interaction analysis, design, and implementation.

THE PROJECT SELECTION AND PLANNING PHASE

In this phase, the HCI and SA&D issues and activities are the same. The organization's total information needs are analyzed and arranged, a potential information system project is identified and an argument for continuing or not continuing the project is presented [Valacich et al., 2004]. A decision to continue with the project must be made at this phase in order to go ahead with the rest of the methodology.

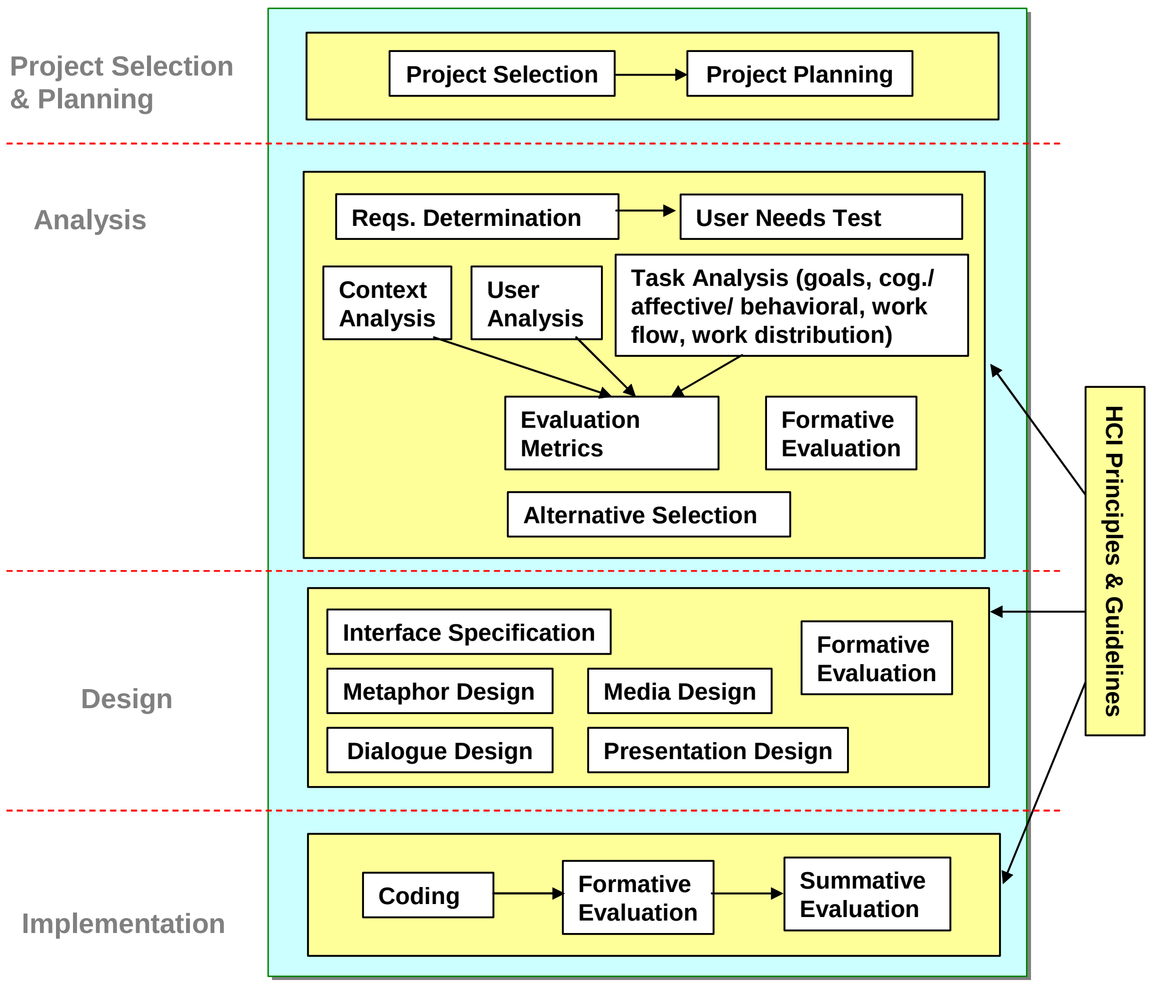


Figure 4. HCI Development Methodology in HCSDLC

THE INTERACTION ANALYSIS PHASE

In modern SA&D, the analysis phase involves determining the system requirements, structuring requirements according to their interrelationships (normally conducted by process analysis, data analysis, and logic analysis), and generating and selecting design alternatives [Valacich et al., 2004]. From the HCI perspective, requirement determination is still one of the most important activities, and alternative generation and selection are also necessary before subsequent design is conducted. In addition, HCI analysis includes user needs tests on the system requirements (which may be demonstrated by mockups or prototypes), and HCI evaluation metrics that are derived from context analysis, user analysis, and task analysis.

Requirement Determination and User Needs Tests

To determine the likelihood of target users' accepting a system's functionalities, user needs tests should be conducted right after the requirements are determined. Errors in requirements specifications are a major contributor to costly software project failures. Verifying requirements of a new system based on user evaluation of specifications measured during the earliest stages is beneficial [Davis and Venkatesh, 2004]. In two longitudinal field experiments, Davis and Venkatesh found that pre-prototype usefulness measured by target users, who received information about a system's functionality without direct hands-on experience, can closely approximate hands-on usefulness measures, and predict usage intentions and behavior up to six months after implementation [Davis and Venkatesh, 2004]. This distinction is key because, compared to ease-of-use, usefulness is generally much more strongly linked to future usage intentions and behaviors. A paper-based survey and paper-based prototypes or mock-ups, can be administered to target users by using Davis and Venkatesh's (2004) instrument. Based on the testing results, designers and managers can decide whether to

- go forward as planned,
- modify or refine requirements to improve acceptability, or
- abandon to avert major losses [Davis and Venkatesh, 2004].

User needs tests can be conducted once or multiple times during this stage. A similar test can also be administered during the project selection and planning stage even though the system requirements are at a higher level and less detailed.

Context Analysis

Once user needs tests are passed, three major analyses are conducted and will determine the HCI evaluation metrics. Context analysis includes understanding the technical, environmental and social settings where the information systems will be used. It examines whether and how the interaction between physical and social environment and the physiological and psychological characteristics of the user would impact users interacting with the system. There are four aspects in Context Analysis: physical context, technical context, organizational context, and social and cultural context. Overall, context analysis can provide ideas for design factors such as metaphor creation/selection and patterns of communications between users and the system.

1. Physical context: Where are the tasks carried out? What entities and resources are implicated in task operation? What physical structures and entities are necessary to understand observed task action? For example, an ATM machine can be used in a mall, outside a bank office, or in a night club. These environments provide different levels of lighting, crowdedness, and noisiness. Thus legibility of the screen, use of audible devices for input or output, or even the size of the working space to prevent people nearly to see the screen could be designed differently.

2. Technical context: What are the technology infrastructure, platforms, hardware and system software, network/wireless connection? For example, an E-commerce website may be designed to allow access only to people with certain browser versions. The website may also be designed to allow small screen devices such as PDA or mobile phone to access.

3. Organizational context: Organizational context may play different roles in internal and external situations. For an organizational information system to be used by the organization's own employees, organizational context analysis answers questions such as: What is the larger system where this information system is embedded? What are the interactions with other entities in the organization? What are the organizational policies or practice that may affect individual's attitude and behavior towards using the system? For example, assuming that Lotus Note is used by an organization as a communication and collaboration tool, management may depend on using the tool to set up meetings by checking employees' calendars on mutually available time slots. The effectiveness of

setting up meetings depends on whether employees use the tool, and how they use it. The whether and how questions can be enforced by organizational policies.

For an organizational information system that is used by people outside the organization, this analysis emphasizes what the user's own organizational factors may come to play when the user uses the system. The significance of organizational context may be less than that in the internal use situation and the role of such organizational context may be less controllable by the system developers. For example, in E-Commerce where customers order products via a system, the customer's organizational context may put certain constraints on using the system. If the system is to be used by a broad range of customers, such organizational issues may be less controllable and less clearly identified. Nevertheless, realizing this uncertainty of customer's organizational environment can help developers to put HCI development in perspectives.

4. Social and cultural context: What are the social or cultural factors that may affect user attitudes and eventual use of the information system? In an E-Commerce website example, the website can be accessed from all over the world. It thus is a design consideration that the website allows access by people with any language and culture background that can provide credit cards with USD exchange, or it is only accessible to people who speak certain languages (such as English, Spanish, and French) and are from certain cultures (such as America).

User Analysis

User analysis identifies the target users and their characteristics including

1. demographic data, such as age, gender, education, occupation, cultural background, any special needs, computer training and knowledge, and experience with similar systems/products;
2. traits and intelligence, such as cognitive styles, affective traits, and skill sets or capability; and
3. job or task related factors, such as job characteristics, knowledge of application domain and job familiarity, frequency of computer use for the job, and usage constraints.

Task Analysis

Task analysis is concerned with understanding what people do to achieve their goals. In developing organizational information systems, it is useful to analyze tasks at two levels: organizational level and tool level. Task analysis should start by identifying the tasks or goals that are meaningful to one's job or work within the organizational context. These tasks can be named Organizational Level Tasks (OLT). Then the task analysis should progress toward understanding OLT by decomposing them into the tasks or actions that users must do to interact with the information system or tools. We name these tasks Tool Level Tasks (TLT). The user interface should be designed to support the TLT directly with the OLT and the organization as their high level contexts.

Task analysis includes scenarios and conditions under which humans perform the tasks. Task analysis reveals patterns of information processing, information needs and representations that users currently use to perform work. It also discovers patterns of exceptions. The objective of task analysis is to identify opportunities to support user activities. For example, sound may be used to draw attention on a visually loaded screen, or sequence of presentation may be altered to help ameliorate biases caused by primacy and recency effects. In HCI, task analysis also distinguishes between what computers do, and what humans do. It examines the task workflow and the distribution of work and work skills among users. A key issue in building new systems is to realize that the new systems change skill sets and obstruct current workflow. Development of a new system must take into account the movement from one type of structured work environment to another.

There should be alignment and consistency between HCI task analysis and the process analysis (such as using Data Flow Diagrams) in SA&D. High level processes in a Level-0 DFD can be considered as organizational level tasks. Certain techniques such as use cases and scenarios can be used for both process and task analyses. In addition, existing techniques on task analysis (e.g., Hierarchical Task Analysis, cognitive task analysis) may be applied at this stage to address certain concerns and aspects. It is worth noting that task analysis in HCI is a challenging and time consuming activity and there is no one-fit-all method or technique to cover the entire spectrum of task analysis concerns. Task analysis may also depend on factors such as the nature of the system being developed.

To illustrate some possible aspects of a HCI task analysis, consider the example of developing a website for selling international foods over the Internet (we call it the International Foods example). The task analysis may identify the following four aspects:

1. User goals and use cases identify five cases or OLT: (Task 1) buy particular foods or ingredients that users already know about, (Task 2) look for ingredients that make a known dish, (Task 3) learn about a particular dish, its ingredients, and how to make it, (Task 4) browse to decide what to cook for a particular occasion, and (Task 5) recommend the site to others. For Task 3, a further analysis may indicate that one of the TLT could be "examining an ingredient on screen." The system then should provide support to this TLT by displaying relevant information about this ingredient in an ergonomic, usable, attractive and interesting way, which eventually supports Task 3.

2. Cognitive, affective, and behavioral analysis of user tasks shows that (1) in Task 1, a user may forget the official name but remembers the characteristics of some food (thus may need to do a query on certain attributes of food to find it first); (2) when examining an ingredient, users may need to refer to the dishes where this ingredient is used. The same is true when examining a dish where ingredients/receipt would be needed; (3) esthetically pleasing presentation would encourage browsing (Tasks 3 & 4) and eventually purchasing foods (Tasks 1 & 2) and recommending the website to others (Task 5); and (4) users may use the forum for peer recommendations and exchange of receipts or cooking experiences (Task 5).

3. Workflow analysis finds that Task 1 would need a sequence of actions to be finished; abandoning the task can occur at any stage of the sequence; and users may want to go back to any of the previous stages; and Task 4 may lead to any of Tasks 1-3.

4. General work distribution between users and the website/machine suggests that users make selections, and the website provides options and all related and relevant information for each choice.

Evaluation Metrics

Evaluation metrics specify the expected goals of human system interaction for the designed system. Such metrics, often quantified into specific measures, guide the rest part of the HCI development process and provide benchmarks for the formative and summative evaluations through out the entire development process. The evaluation metrics correspond to the multiple concerns of HCI as summarized previously in Table 2. The specific measures or quantitative aspects of the metrics come from the analysis results (context, user, and task analyses), formative evaluation tests on mockups or prototypes, industrial or international standards if any, as well as the goals and constraints of the information system being developed. This last consideration may come from the "Alternative Selection" activity that is the last activity in the interaction analysis phase (see below). Basically, the higher the HCI expectations (such as no users should make any errors), the more costly it will be to develop the system. Thus trade-offs may be necessary to achieve reasonable HCI goals (e.g., less than 10% of users would make some mistakes on certain tasks when using the system) within feasible development constraints.

Table 4 lists the evaluation metrics in general and the International Foods example to illustrate the evaluation metrics. A complete example in a later section will provide additional details on establishing evaluation metrics. The first column of Table 4 reviews the general goals of the evaluation metrics that correspond to those in Table 2. The second column of Table 4 demonstrates the possible evaluation metrics for the International Foods example.

Table 4. Evaluation Metrics and an Example

Evaluation Metrics Template	**International Foods Example**
Physical/ergonomic Concern	
Legibility Audibility Safety in use	• 85% of the potential customers can read the text and image with ease. • 85% of the potential customers feel that the sound produced by the system is audible and not hurting • 85% of the potential customers think that using the system does not impose health concerns
Cognitive/Usability concern	
Fewer errors and easy recovery Easy to use Easy to remember how to use Easy to learn	• New customers are able to navigate and use the main functions within 5 minutes. • Customers are able to get to the main tasks with one click • Ordering a type of foods should be done within 1 minute in normal network traffic and with no more than 4 clicks/actions. • Error rate should be less than 1 in every 10 customers for each main task. • The complaining rate of usability problems should be less than 1 in every 10 customers
Affective, Emotional, Intrinsic Motivation Concern	
Aesthetically pleasing Engaging Trustworthy Satisfying Enjoyable Entertaining Fun	• 85% of the tested shoppers should have (a) aesthetic, (b) enjoyable, (c) engaging and (d) satisfactory rating of at least 4 out of 5 with 5 the highest • At least 85% of the potential target users would trust the website for credit card use • Relaxed atmosphere for ordering foods • No unnecessary anxiety imposed by the interface design such as “customers have to complete purchasing in 10 seconds.”
Extrinsic Motivation/Usefulness Concern	
Support individual’s tasks Can do some tasks that would not so without the system Extend one’s capability	• Customers can order the types of foods that they normally cannot get from a local store. • Customers can order small amount of foods with an affordable price and shipping. • Customers can learn new ways of cooking international gourmet meals.

Alternative Selection

Consistent with SA&D, before transforming all gathered and structured information from the analysis phase into design ideas, the organization must select the final alterative design strategy for the proposed information system because (1) different users offer competing ideas on what the system should do, and (2) multiple alternatives are available for an implementation environment for any new system [Valacich et al., 2004]. Although SA&D emphasizes functionality in selecting design strategies, the approach of generating and selecting best alternatives can also be applied to HCI design strategies. The deliverables include (1) three substantially different design strategies (low, middle and high range) that come from different requirements specifications and HCI evaluation metrics, and, (2) a design strategy judged most likely to lead to the most desirable system, from functionality/usefulness, ergonomic, usability, and intrinsic motive perspectives, given all of the organizational, economic and technical constraints that limit what can be done. This alternative selection activity will help shape the final HCI evaluation metrics, as mentioned above. This is another example of iterations among activities within the same stage of SDLC. Other issues to consider when generating design strategies include examining different ways of constructing the system such as outsourcing, off the shelf, or in-house development. If in-house development is chosen, the interaction design and implementation phases will continue.

THE INTERACTION DESIGN PHASE

In this phase, the user interface is specified, sketched, developed, and tested. The goal is to support the identified issues during context, task and user analyses and to meet the HCI evaluation metrics requirements. Design is also based on accepted conventions and experience. The main activities are interface specification and formative evaluations. Interface specification includes semantic understanding of the information needs to support systems requirements and HCI analysis results, and syntactical and lexical decisions including metaphors, media, dialogue, and presentation designs.

Metaphor and Visualization Design

Metaphor and visualization design helps the user develop a mental model of the system. It is concerned with finding or inventing metaphors or analogies that are appropriate for users to understand the entire system or part of it. Well accepted metaphors include a shopping cart for holding items before checking out in E-Commerce context, and light bulbs for online helps or daily tips in productivity software packages.

Media Design

Media design is concerned with selecting appropriate media types for meeting the specific information presentation needs and human experience needs. Popular media types include text, static images (e.g., painting, drawing or photos), dynamic images (e.g., video clips and animations), and sound. The bandwidth needed for transmitting information depends on the media type. In addition, some media types contain affective qualities [Zhang and Li, Forthcoming] that can make presentations more interesting and stimulating, or annoying and distasteful.

Dialogue Design

Dialogue design focuses on how information is provided to and captured from users during a specific task. Dialogues are analogous to a conversation between two people. Many existing interaction styles [Shneiderman and Plaisant, 2005] can be used such as menus, forms, natural languages, dialog boxes, and direct manipulation.

Presentation Design

Presentation design concerns the decisions on information architecture and display layout incorporating metaphors, media, and dialogue designs with the rest of the displays.

Commonly established user interface design principles and guidelines may be applied during the design stage. For example, the following presentation design principles were suggested by Sutcliffe [Sutcliffe, 1997]:

1. maximize visibility – all necessary information should be immediately available;
2. minimize search time with minimum keystrokes;
3. provide structure and sequence of display;
4. focus user attention on key data – important information should be salient and easily comprehended;
5. provide only relevant information; and
6. no overloading user's working memory.

Shneiderman and Plaisant [2005] provide detailed design guidelines for each of the commonly used interaction styles.

Formative Evaluation

Formative evaluations identify defects in designs thus inform design iterations and refinements. A variety of different formative evaluations can occur several times during the design stage to form final decision decisions. In fact, we propose that formative evaluations occur during the entire HCI development life cycle, as depicted in Figure 3.

THE IMPLEMENTATION PHASE

HCI development in this phase includes

1. coding that is also part of SA&D,
2. formative evaluations to fine tune the system,
3. summative evaluation before system release, and
4. use evaluation after the system is installed and being used by target users for a period of time.

Summative evaluation takes place after the system is developed to confirm whether the evaluation metrics or other industry standards are met. Use evaluation collects feedback in understanding users' attitude and actual behavior toward system use. This understanding helps in developing new versions or other similar systems.

A TEMPLATE TO DOCUMENT HCI DEVELOPMENT ACTIVITIES AND DELIVERABLES

It is necessary to communicate to the clients or teammates about the HCI development activities and results. Although there are many details that can be and should be documented, a rather standard format that gives an overview of the entire project will facilitate communication and understanding. Other detailed documents, such as task analysis results (could be many pages and levels), and design alternative sketches, can be attached to the overview report.

We present a format that is based on our HCSDLC model, or specially, the HCI development methodology part. Table 5 lists the template for the HCI development report. This template can be easily streamlined with the Common Industry Format (CIF) that is designed for summative usability tests and is currently used in industry [Bevan et al., 2002].

A SAMPLE APPLICATION

To help the reader understand the HCI methodology for systems analysis and design, Appendix I presents an example of a simple scenario and how HCI is applied.

Table 5. HCI Development Life Cycle (HCI-DLC) Report Template

ID	HCI Development Activity	Deliverables
1.1	Project Selection and Planning	Schedule of IS projects development: Cost-benefit analysis: Other feasibility analyses:
2.1	Requirements Determination	The specific system functionalities:
2.2	User Needs Test	Sample profile: Data collection time and setting: Sketches or mockups used: Test results: Suggestions on revising system functionalities:
2.3	Context Analysis	Physical context: Technical context: Org context: Social/cultural context:
2.4	User Analysis	Demographic: Traits/skill sets: Job or task related factors:
2.5	Task Analysis	User goals and use cases (OLTs and TLTs): Cognitive, affective, behavioral analysis of user tasks: Workflow analysis: General work distribution between users and the system:
2.6	Evaluation Metrics	Ergonomic concerns: Usability concerns: Emotional concerns: Usefulness concerns:
2.7	Alternative Selection	Three alternatives: The main constraints: The chosen alternative:
2.8	Formative Evaluation	Evaluation target, method, timing and results:
3.1	Interface Specification	Metaphor and visualization design: Media design: Dialogue design: Presentation design:
3.2	Formative Evaluation	Evaluation target, method, timing and results:
4.1	Prototyping	Tools used:
4.2	Formative Evaluation	Evaluation target, method, timing and results:
4.3	Summative Evaluation	Sample profile: Data collection time and setting: Test results: Conclusions in light of evaluation metrics:

VII.SUMMARY AND CONCLUSION

Methods and techniques in both the SA&D (including software engineering) and the usability engineering disciplines matured over the years and are used for education, training, and guiding practice. However, little effort was invested in providing integrated methodologies for developing human-centered information systems that consider both organizational and human needs. This lack of integration is problematic to our students who often take different courses with different emphases. The same problem applies to information systems developers who are responsible for delivering both organizationally effective and human-centric systems but often find reference books with one emphasis or the other. Diverse approaches with different perspectives may help to isolate different issues but they do not help with overall effectiveness and efficiency of systems development. The result of this situation is that developed information systems often either lack well-defined systems requirements to support organizational needs, or lack human understanding and thus are frustrating to use.

The proposed human-centered SDLC model in this paper is an integrated methodology that emphasizes human-centeredness and considers HCI issues together with SA&D issues throughout the entire system development life cycle. The methodology uses the parsimony of the SDLC model that is helpful from the project management perspective and as a training wheel. It lays out the connections and differences between SA&D and HCI concerns and activities, and provides a step-by-step procedure for transformations between activities at different stages. This methodology can be used for courses on human-centered information systems analysis and design (the whole methodology), HCI and user interface design (the HCI development part of the methodology), and IS project management courses where all factors including human factors in IS development should be considered. We hope that the methodology presented will be instrumental in providing more successful information systems and thus more successful businesses and better human experiences.

ACKNOWLEDGEMENTS

The authors appreciate the feedback from the audience during the presentation of early ideas of this paper at the Americas Conference on Information Systems (AMCIS 2004) in New York City.

Editor's Note: This article was received on November 5, 2004. It was with the authors for approximately 2 months for 2 revisions. The article was published on April 10, 2005

REFERENCES

EDITOR'S NOTE: The following reference list contains the address of World Wide Web pages. Readers who have the ability to access the Web directly from their computer or are reading the paper on the Web, can gain direct access to these references. Readers are warned, however, that

1. these links existed as of the date of publication but are not guaranteed to be working thereafter.
2. the contents of Web pages may change over time. Where version information is provided in the References, different versions may not contain the information or the conclusions referenced.
3. the authors of the Web pages, not CAIS, are responsible for the accuracy of their content.
4. the author of this article, not CAIS, is responsible for the accuracy of the URL and version information.

Agarwal, R. and E. Karahanna (2000) "Time Flies when You're Having Fun: Cognitive Absorption and Beliefs about Information Technology Usage," *MIS Quarterly* (24)4, pp 665-694.

Baskerville, R. and J. Pries-Heje (2004) "Short cycle time systems development," *Information Systems Journal* (14), pp 237-264.

Beck, K. (2000) *eXtreme Programming eXplained*, Upper Saddle River, NJ: Addison-Wesley.

Bevan, N. (2001) "International standards for HCI and usability," *International Journal of Human-Computer Studies* (55)4, pp 533-552.

Bevan, N., N. Claridge, M. Athousaki, and M. Maguire (2002) "Guide to specifying and evaluating usability as part of a contract," *http://www.usability.serco.com/prue/*.

Bjerknes, G., P. Ehn, and M. Kyng (eds.) (1987) *Computers and Democracy - a Scandinavian Challenge*. Aldershot, England: Avebury, Gower Publishing Company Ltd.

Boivie, I., C. Aborg, J. Persson, and M. Lofberg (2003) "Why usability gets lost or usability in in-house software development," *Interacting with Computers* (15), pp 623-639.

Bostrom, R.P. and J.S. Heinen (1977) "MIS Problems and Failures: A Socio-Technical Perspective. Part I: The causes," *MIS Quarterly* (1)3, pp 17-32.

Brave, S. and C. Nass (2003) "Emotion in Human-Computer Interaction," in Jacko, J. and A. Sears (eds.), *The Human-Computer Interaction Handbook,* Mahwah, NJ: Lawrence Erlbaum Associates, Inc., pp. 81-96.

Burrell, G. and G. Morgan (1979) *Sociological paradigms and organizational analysis*, London: Heinemann.

Carey, J., D. Galletta, J. Kim, D. Te'eni, B. Wildermuth, and P. Zhang (2004) "The Role of HCI in IS Curricula: A Call to Action," *Communication of the AIS* (13)23, pp 357-379.

Checkland, P. and J. Scholes (1990) *Soft Systems Methodology in Action*, Toronto: John Wiley & Sons.

Clegg, C., C. Axtell, L. Damodaran, B. Farbey, R. Hull, R. Lloyd-Jones, J. Nichols, R. Sell, and C. Tomlinson (1997) "Information Technology: a Study of Performance and the Role of Human and Organizational Factors," *Ergonomics* (40)9, pp 851-871.

Davis, F. (1989) "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology," *MIS Quarterly* (13)3, pp 319-340.

Davis, F. and V. Venkatesh (2004) "Toward Preprototyping User Acceptance Testing of New Information Systems: Implications for Software Project Management," *IEEE Transactions on Engineering Management* (51)1, pp 31-46.

Dennis, A. and B.H. Wixom (2003) *Systems Analysis Design*, (2nd ed.), New York: John Wiley & Sons.

Eason, K. (1997) "Understanding the Organisational Ramifictions of Implementing Information Technology Systems," in Helander, M.G., T.K. Landauer and P. Prabhu (eds.), *Handbook of Human-Computer Interaction,* Amsterdam: Elsevier, pp. 1475-1494.

Forgas, J.P. (1995) "Mood and Judgment: The Affect Infusion Model (AIM)," *Psychological Bulletin* (117)1, pp 39-66.

Fowler, M. and J. Highsmith "The Agile Manifesto," 2001.

Grudin, J. (1992) "Utility and Usability: Research Issues and Development Contexts," *Interacting with Computers* (4)2, pp 209-217.

Grudin, J. (1993) "Interface: An Evolving Concept," *Communication of the ACM* (36)4, pp 110-119.

Hefley, W.E., E.A. Buie, G.F. Lynch, M.J. Muller, D.G. Hoecker, J. Carter, and J.T. Roth (1995) "Integrating Human Factors with Software Engineering Practices," in Perlman, G., G.K. Green and M.S. Wogalter (eds.), *Human Factors Perspectives on Human-Computer Interaction: Selections from the Human Factors & Ergonomics Society Annual Meetings 1983-1994,* Santa Monica, CA: Human Factors and Ergonomics Society, pp. 359-363.

Herzberg, F. (1966) *Work and the Nature of Man*, New York: World Publishing, pp. 71-91.

Hewett, T., R. Baecker, S. Card, T. Carey, J. Gasen, M. Mantei, G. Perlman, G. Strong, and W. Verplank (1992) *ACM SIHCHI Curricula for Human-Computer Interaction*, New York: Association for Computing Machinery.

Hirschheim, R. and H.K. Klein (1989) "Four Paradigms of Information Systems Development," *Communication of the ACM* (32)10, pp 1199-1216.

Hix, D. and H.R. Hartson (1993) *Developing User Interfaces: Ensuring Usability Through Product and Process*, New York: John Wiley.

Hoffer, J.A., J.F. George, and J.S. Valacich (2005) *Modern Systems Analysis and Design*, (4th ed.), Upper Saddle River, NJ: Prentice Hall.

Kendall, K. and J. Kendall (2005) *Systems Analysis and Design*, (6th ed.), Upper Saddle River, NJ: Prentice Hall.

Mantei, M. and T. Teorey (1989) "Incorporating Behavioral Techniques into the System Development Life Cycle," *MIS Quarterly* (13)3, pp 257-274.

Maslow, A.H. (1962) *Toward a psychology of being*, Princeton, NJ: D. Van Norstrand.

Maxwell, K. (2002) "The Maturation of HCI: Moving Beyond Usability toward Holistic Interaction," in Carroll, J.M. (ed.), *Human-Computer Interaction in the New Millennium,* New York: Addison-Wesley, pp. 191-209.

Mayhew, D.J. (1999) *The Usability Engineering Lifecycle - A Practitioner's Handbook for User Interface Design*, San Francisco, CA: Morgan Kaufmann Publishers, Inc.

Mumford, E. (1983) *Designing Human Systems for New Technology - The ETHICS Method*, Manchester, UK: Manchester Business School.

Nielsen, J. (1993) *Usability Engineering*, New York: AP Professional.

Norman, D.A. (2004) *Emotional Design: Why We Love (Or Hate) Everyday Things*, Cambridge, MA: Basic Books.

Patrick, J.R. (2003) "The Future of the Internet, Keynote Speech," Americas Conference on Information Systems, Tampa, FL.

Preece, J., Y. Rogers, and H. Sharp (2002) *Interaction Design: Beyond Human-Computer Interaction*, New York: John Wiley & Sons.

Preece, J., Y. Rogers, H. Sharp, D. Benyon, S. Holland, and T. Carey (1994) *Human-Computer Interaction* Addison-Wesley.

Reeve, J. (2005) *Understanding Motivation and Emotion*, (4th ed.), New york: John Wiley & Sons, Inc.

Rising, L. and N.S. Janoff (2000) "The Scrum Software Development Process for Small Teams," *IEEE Software*July/August, pp 26-32.

Russell, J.A. (2003) "Core Affect and the Psychological Construction of Emotion," *Psychological Review* (110)1, pp 145-172.

Satzinger, J.W., R.B. Jackson, and S.D. Burd (2004) *Systems Analysis & Design in a Changing World*, (3rd ed.), Boston, MA: Course Technology.

Shneiderman, B. and C. Plaisant (2005) *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, New York: Addison-Wesley.

Sun, H. and P. Zhang (2005) "The Role of Affect in IS Research: A Critical Survey and a Research Model," in Galletta, D. (ed.), *Human Computer Interaction and Management Information Systems: Foundations* M.E. Sharpe.

Sutcliffe, A. (1997) "Task-Related Information Analysis," *International Journal of Human-Computer Studies* (47), pp 223-257.

Sutcliffe, A. (2000) "On the Effective Use and Reuse of HCI Knowledge," *ACM Transactions on Computer-Human Interaction* (7)2.

Tractinsky, N., A.S. Katz, and D. Ikar (2000) "What is Beautiful is Usable," *Interacting with Computers* (13), pp 127-145.

Truex, D., R. Baskerville, and J. Travis (2000) "Amethodical Systems Development: The Deferred Meaning of Systems Development Methods," *Accounting, Management and Information Technology* (10), pp 53-79.

Valacich, J.S., J.M. George, and J.A. Hoffer (2004) *Essentials of Systems Analysis and Design*, (2nd edition ed.), Upper Saddle River, NJ: Prentice Hall.

Venkatesh, V. and F. Davis (2000) "A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies," *Management Science* (46)2, pp 186-204.

Venkatesh, V., M.G. Morris, G.B. Davis, and F.D. Davis (2003) "User Acceptance of Information Technology: Toward a Unified View," *MIS Quarterly* (27)3, pp 425-478.

Webster, J. and J.J. Martocchio (1993) "Turning Work into Play - Implications for Microcomputer Software Training," *Journal of Management* (19)1, pp 127-146.

Whitten, J., L. Bentley, and K. Dittman (2004) *Systems Analysis and Design Methods*, (6th ed.), Boston: McGraw-Hill Irwin.

Winograd, T. and F. Flores (1986) *Understanding Computers and Cognition*, Norwood, NJ: Ablex Publishers.
Zhang, P., I. Benbasat, J. Carey, F. Davis, D. Galletta, and D. Strong (2002) "Human-Computer Interaction Research in the MIS Discipline," *Communications of the AIS* (9)20, pp 334-355.
Zhang, P. and N. Li (Forthcoming) "The importance of affective quality," *Communication of the ACM*.
Zhang, P., G.M. von Dran, R.V. Small, and S. Barcellos (2000) "A Two-Factor Theory for Website Design," The Thirty-Third Annual Hawaii International Conference on System Sciences (HICSS33).

APPENDIX I. APPLYING THE HCI METHODOLOGY

This appendix illustrates the application of the HCSDLC methodology to a fictional case. We focus primarily on the HCI analysis and design stages of the HCI side and mention other activities when necessary.

BACKGROUND

"Teaching Tools" is a small company owned by two retired elementary school teachers, Janet and Chris. They have been creating teaching materials and tools for about ten years. During this time, they sold their products at school bazaars, through flyers, and by direct mail to existing customers. The owners wish to develop an e-commerce web site. They contracted with HCD (Human-Centered Development Inc.) to build the site.

HCI DEVELOPMENT LIFE CYCLE REPORT

Table A-1 summarizes the HCI development activities and deliverables for the Teaching Tools example. Detailed explanations are embedded in the table.

Table A-1. HCI Development Activities and Deliverables for the Teaching Tools

ID	HCI Development Activity	Teaching Tools Example
1.1	Project Selection and Planning	Owners and consultant have completed an in-depth feasibility study and determined the cost/benefits of the site and created a tentative budget and schedule for the project. They have made a decision to go ahead and build the web site.
2.1	Requirements Determination	The potential systems requirements are: (1) recommending a teaching tool that will meet customers' needs, (2) taking online orders by using credit cards, (3) providing sample lesson plans that go along with each tool, (4) providing learning objectives for each tool, and (5) providing a forum for teachers and learners to exchange ideas and experiences. The last one has a lower priority than the other four requirements.
2.2	User Needs Test	A ten-person focus group was selected to help determine whether the proposed requirements will meet customer needs. The focus group consists of 5 long-term customers and 5 new customers. Through an iterative process, the user needs test affirmed the 4 requirements and agreement that the discussion forum would be nice but not necessary to support sales.
2.3	Context Analysis	(1) Physical context: users may order or browse primarily from home or school. These two physical environments do not generally pose

ID	HCI Development Activity	Teaching Tools Example
		distracting physical aspects. The school environment can be a bit chaotic when students are present, but the teachers in the focus group said that they would not try to order Teaching Tools products with students present. However, one teacher said that she likes to get student input and often asks a small group of students to help her choose the tools at the beginning of the school year. The home environment can be a bit distracting also, but again the focus-group teachers said that they would wait for a quiet time at home to place orders. Overall, there were no special requirements due to the physical context and therefore they could choose a fairly standard e-commerce design. (2) Technical context: It is unlikely that users will browse the website using Palm PDAs, or mobile phones. It is more likely that the users will be browsing from a desktop or laptop PC with a cabled Internet connection. The screen can be assumed therefore to be full sized for a regular computer monitor. (3) Organizational context: The website reflects the business strategies of the organization and thus is subject to business decision changes made at the strategic level. The site is primarily commercial and has the goal of making money and budget constraints that must be respected. However, the owners are emphatic that the site should reflect their student-centered philosophy. In addition, Teaching Tools does not have other computer based information systems for the website to link to. (4) Social and cultural context: The site is not really considered a global site, however one of the owners is a teacher of English as a second language (ESL) and wants a Spanish version of the site and plans to add additional language versions as money allows.
2.4	User Analysis	(1) Demographic data: users are primarily female elementary school teachers in the US who speak mainly English and some Spanish. (2) Traits and intelligence: users have a college degree and are fairly experienced computer users and often purchase items through the Internet, (3) Job or task related factors: users may purchase items from the Teaching Tools website two times per year.
2.5	Task Analysis	The overall goal is to select the appropriate teaching tool. Sample tasks: Task 1 is to specify the teaching requirements and selection criteria for the tool. The criteria include; cost, author reputation, level of difficulty, level of study (grade level), and supplement; Task 2 is to evaluate the criteria and provide alternative options. Task 3 is to choose the tool.
2.6	Evaluation Metrics	After some research on existing web sites and metrics published in trade journals, the following evaluation metrics are established (note: use needs test has verified the usefulness of the website): (A) Ergonomic metrics: 1. 85% of the potential customers can read the text and image

ID	HCI Development Activity	Teaching Tools Example
		with ease. 2. 85% of the potential customers feel that the sound produced by the system is audible and not hurting 3. 85% of the potential customers think that using the system does not impose health concerns <u>(B) Usability metrics</u>: 1. New users are able to navigate and use the main functions within 10 minutes 2. Users are able to get to the main tasks with one or two clicks 3. Ordering teaching tools should be done within one minute from the time of clicking the submit button until a confirmation screen is returned. 4. Error rate should be less than 1 in every twenty users for each purchase task. 5. The number of complaints should be less than 1 in 100 uses. 6. Secure connection should be alerted when the consumer is entering his or her credit card and personal information. <u>(C) Affective and emotional metrics:</u> 1. 70% of the tested shoppers should have (a) aesthetic, (b) enjoyable, (c) engaging and (d) satisfactory rating of at least 4 out of 5 2.At least 10% of the shoppers would participate in the forum at least once every three months (read or send postings) 3.At least 70% of the potential target users would trust the website for credit card use. (D) Usefulness metrics: 1. Customers can order the types of foods that they normally cannot get from a local store. 2. Customers can order small amount of foods with an affordable price and shipping. Customers can learn new ways of cooking international gourmet meals.
2.7	Alternative Selection	Three prototype designs were developed to reflect three alternative design strategies. They differed in systems requirements and HCI evaluation expectations. The low range one has the very basic system functions and minimum evaluation expectations. The high range has the most powerful set of functions and the highest level of evaluation expectations. The middle range is a trade-off between the low and high ones. The main constraints for choosing the final design strategy were financial and level of user sophistication. The chosen alternative is the one that meets the level of user sophistication and the most affordable. Additional features may be added over time as money is available.
2.8	Formative Evaluation	Ongoing testing to see if evaluation metrics are being met or should be adjusted.
3.1	Interface Specification	<u>(1) Metaphors and visualizations</u>: 1. Classroom metaphor or Storefront metaphor for organizing products and tools 2. Shopping cart metaphor for holding potential purchases <u>(2) Media design</u>: Text, drawings or photos, and animations may be used.

ID	HCI Development Activity	Teaching Tools Example
		(3) Dialogue design: Existing interaction styles, such as menus, form-fill-ins, natural languages, dialog boxes, and direct manipulation, will be used. (4) Presentation design: Concerns the overall organization of the whole site and the layout of each page. Include navigational buttons or bars, and/or floating text. See next section and Figures 5-10 for design examples.
3.2	Formative Evaluation	Prototypes reiterated and ongoing testing occurred via focus group and owner feedback to see if evaluation metrics are being met.
4.1	Coding	Visual Basic.Net is used as a coding tool.
4.2	Formative Evaluation	Prototypes and working systems reiterated and tested.
4.3	Summative Evaluation	A test web site was created as a beta test site. Key customers were invited to visit the site and provide feedback. Then, a production version of the system was developed. The site has been modified and improved over time. The current site is stable but improvements are made according to customer and owner feedback. The current site has met the evaluation metrics set up in the early phases of the project. It took several months and countless revisions to meet these criteria. The site is beginning to pay for itself and has added 50% to overall revenues compared to the quarter prior to the installation of the web site.

Next we discuss the interaction design phase in detail. In this phase, HCD designs the user interface with the goal of supporting the issues identified during context, task and user analyses and also meeting the HCI evaluation metrics requirements. The design team for Teaching Tools created mock up designs to demonstrate their design ideas and to gain feedback from the owners and potential users. One design incorporates stacked buttons and pop-up menus (Figure A-1). The second design uses navigational buttons and floating text (Figure A-2).

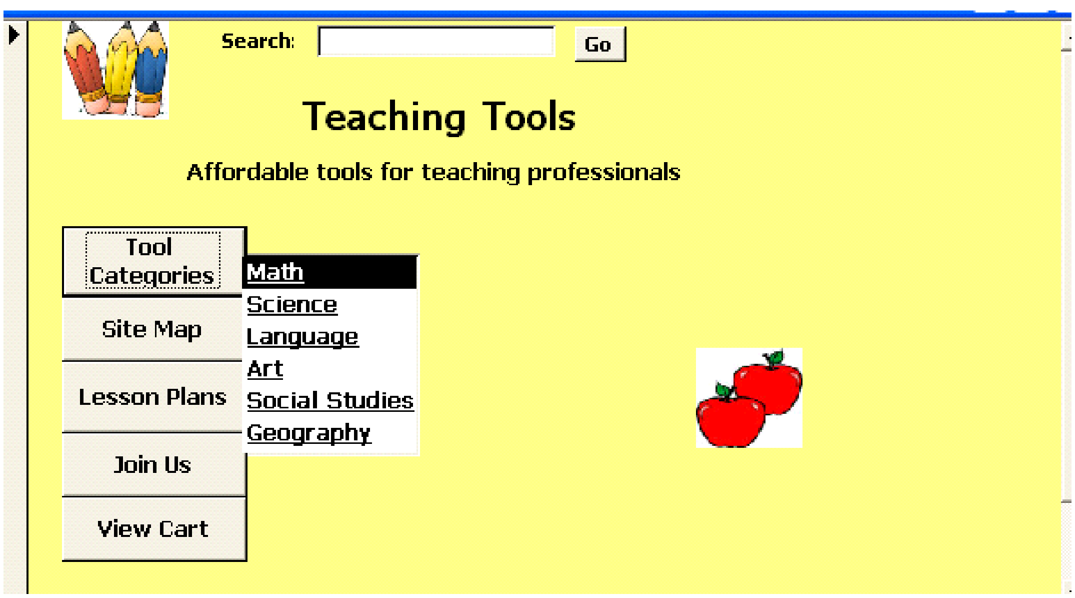


Figure A-1. Design 1 for the Teaching Tools Home Page

Figure A-2. Design option 2 for Teaching Tools Home Page

The search option was added at the focus groups' suggestion. They wanted to be able to search from anywhere in the site for a specific tool for which they already knew the name.

After trying out both options, the user focus group, together with Janet and Chris selected design option 2. The next screen (Figure A-3) follows from the selected design. It includes navigation buttons..

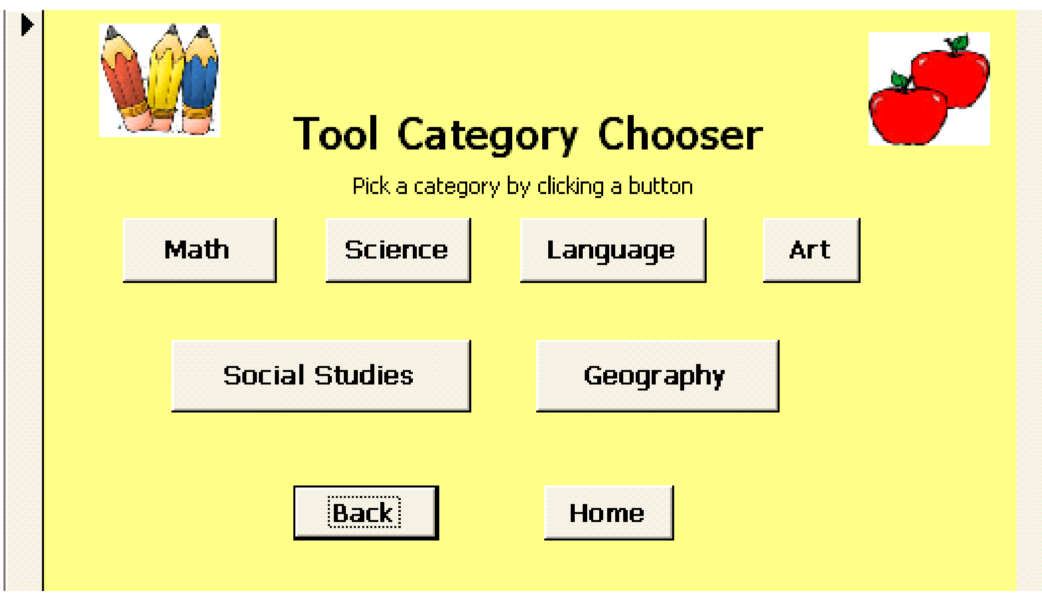


Figure A-3 Tool Category Chooser Screen

Once the user chooses the category of tool, the next task is to specify the values of the list of parameters of the tools that were identified during the analysis stage.

The selection criteria for the tool chooser screen includes: 1) cost, 2) author reputation, 3) level of difficulty, 4) level of study, and 5) whether a supplement is included. Figure A-4 shows the design of the Tool Chooser screen.

In addition to the metaphor and dialog design, interaction design includes also media design and presentation design. As noted in Section V, presentation design follows established design principles. For example, the fields that are related logically are also placed physically in close proximity. For a particular choice of tool, the tool category, level of study, and level of difficulty are relatively inflexible. The user is offered more flexibility and room for compromise if necessary in determining the cost, author reputation and whether a supplement is necessary. Hence, the first three fields are located within one physical group and the remaining three fields in another. Notice also, that the level of difficulty of the tool is in relation to the level of study so that these two fields are positioned one near the other. Finally, the order of the fields attempts to follow a logical or preferred order of input. These screens are all examples of design guidelines that we apply in presentation design.

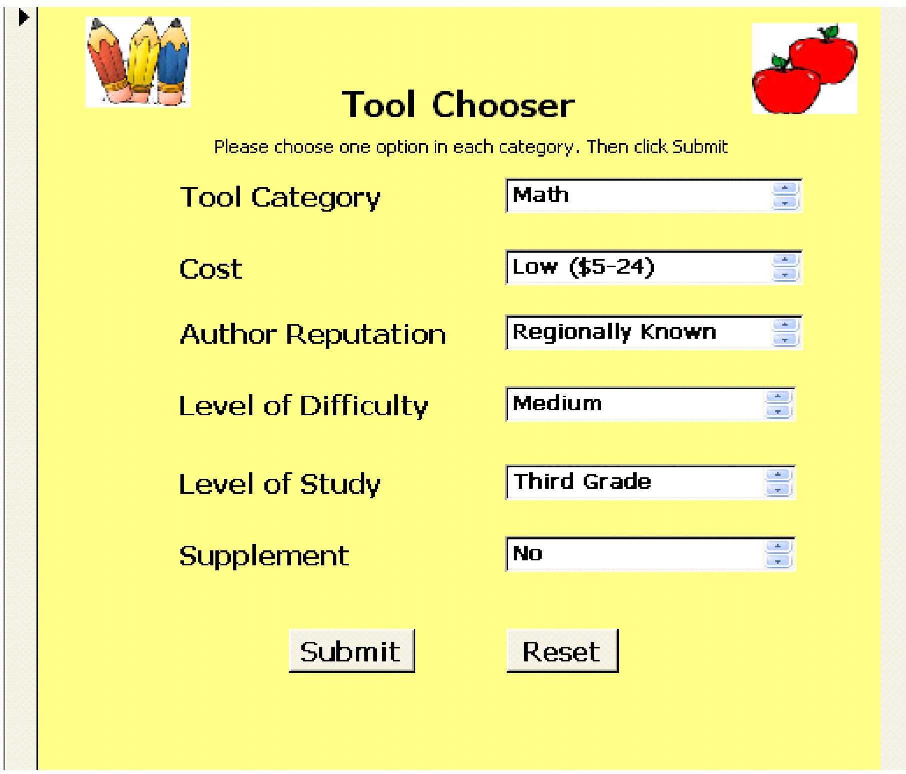


Figure A-4. Tool Chooser Screen Design

The Tool Chooser Screen allows the user to select from various options by scrolling through the options so that the desired option appears in the text box. Once all the options are satisfactory, the user then clicks the submit button. A confirmation message box for both the submit button and the reset button allows the user to confirm the choices made. Figure A-5 shows the confirmation message box for the submit button.

Once the user confirms the selection criteria in the Tool Chooser screen, the system searches the database of existing tools and finds all the tools that match the selection criteria and presents them to the user one tool at a time. The user can explore the tool through many different avenues including Figure A-6, which shows a tool that matches the selection criteria in Figure A-5.

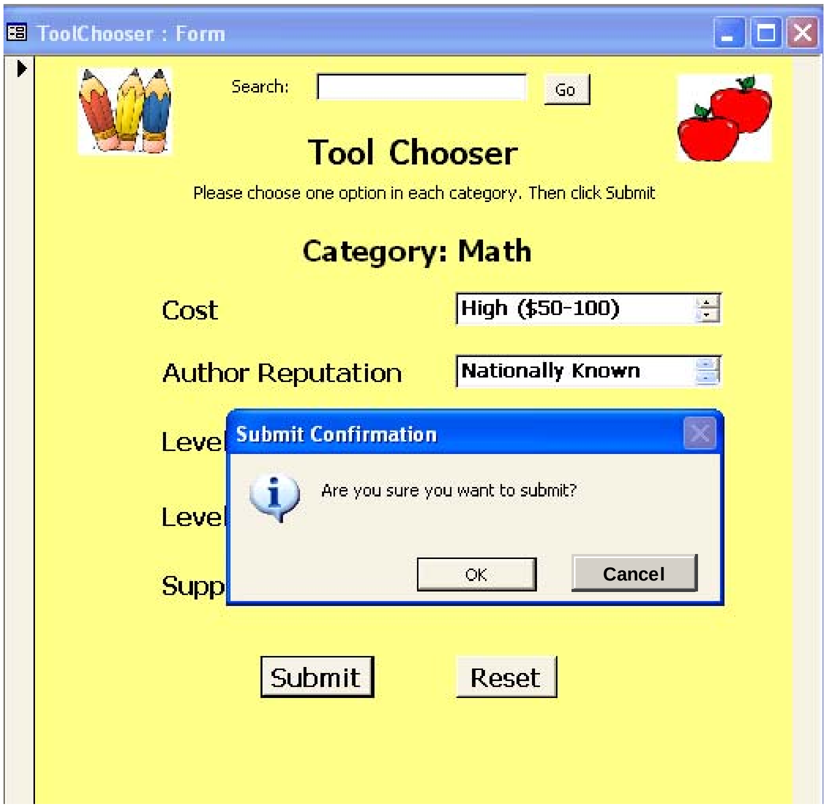


Figure A-5 Confirmation of Submit Button Click Event

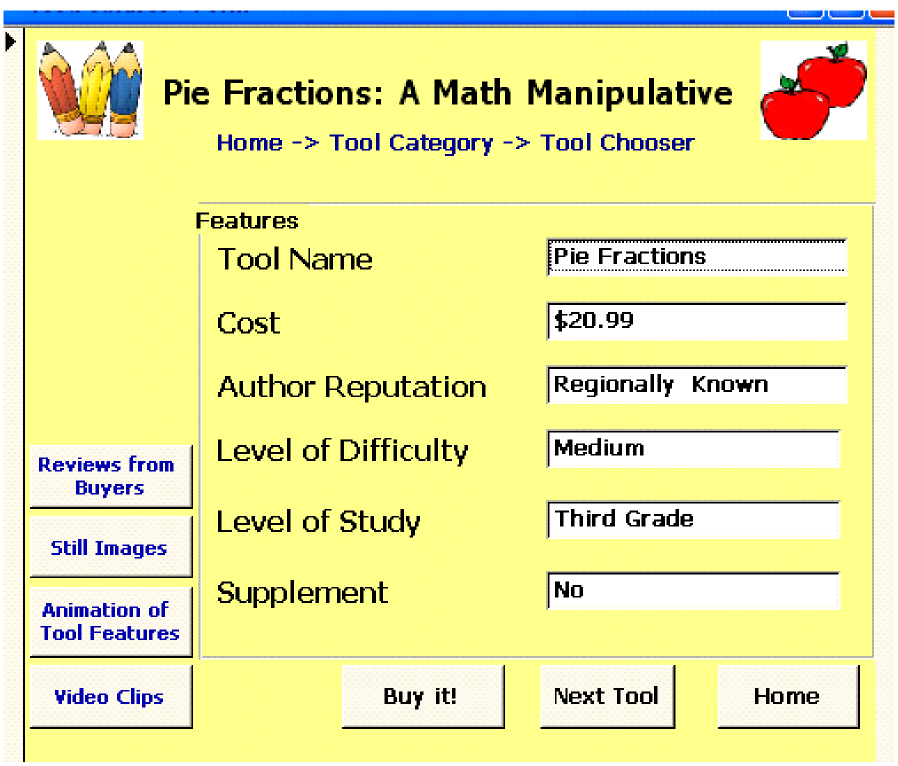


Figure A-6 Tool Features Screen

The screen in Figure A-5 is meant to help the user evaluate a particular teaching tool. What is the best way of presenting the tool? Media design requires that we consider the alternative media and select those that are most helpful in evaluating the tools. In this case, it was important to provide not only textual descriptions but also pictures (still images) of the tool from revealing directions. Furthermore, to show how the tool's features are used dynamically, animations can be used to represent the sequential operation efficiently. Moreover, teachers are always concerned about how the tool is actually used in class. Therefore, a video clip of using the tool in a realistic session is also available. The result of the media design is reflected in the buttons at left side of the screen.

If the user wants this tool, he or she can click the "Buy it!"-button that goes to the shopping cart and subsequently to the purchase screen. If the user is not sure whether he or she wants to buy the tool, the user can place the tool in the shopping cart and view the cart at any time. From the cart page, the user can delete any of the tools not wanted. If the user knows they do not want the tool, he or she can click the Next Tool button, which brings up the next tool that matches the selection criteria. If there are no more tools, a message box pops up that indicates that there are no more tools and asks the user if they want to try to find another tool (sends them back to the tool chooser category page) or finish shopping by going to the shopping cart. The example ends here. The shopping cart screen and the payment screen would be similar to and consistent with existing e-commerce screens.

LIST OF ABBREVIATIONS

DFD	Data Flow Diagrams
HCSD	Human-Centered Systems Development
HCSDLC	Human-Centered Systems Development Life Cycle
ISDM	Information Systems Development Methodology
OLT	Organizational Level Tasks
SA&D	Systems Analysis and Design
SDLC	Systems Development Life Cycle
TAM	Technology Acceptance Model
TLT	Tool Level Tasks
UCD	User-Centered Design

ABOUT THE AUTHORS

Ping Zhang is Associate Professor at the School of Information Studies, Syracuse University. She earned her Ph.D. in Information Systems from The University of Texas at Austin. She teaches Human-Computer Interaction, Systems Analysis and Design, Database Management Systems, and other IT related topics. Her research appears or will appear in journals such as *Behaviour & Information Technology (BIT)*, *Communications of ACM (CACM), Communications of AIS (CAIS), Computers in Human Behavior (CHB), Decision Support Systems (DSS), International Journal of Human-Computer Studies (IJHCS), International Journal of Electronic Commerce (IJEC), Journal of American Society for Information Science and Technology (JASIST), Journal of AIS (JAIS).* Prof. Zhang is the author or editor of 3 books. She is Associate Editor for IJHCS and CAIS, and a guest co-editor for special issues of JAIS, JMIS, IJHCS, IJHCI, and BIT. Dr. Zhang is the founding chair (2001-2004) of the AIS Special Interest Group on HCI (SIGHCI).

Jane M. Carey is associate professor of information systems in the School of Global Management and Leadership at Arizona State University. Dr. Carey's research interests are in the areas of human computer interaction (HCI), the impact of Information Technology on Supply

Chain Management in China, and software globalization. She published seven books as well as many journal articles and book chapters.

Dov Te'eni is Professor of Information Systems in the faculty of management at Tel-Aviv University, Israel. He is also the chairman of Meital - Israel's Higher Education E-learning center. Dov studies several related areas of information systems: human-computer interaction, computer support for communication, knowledge management, systems design and non-profit organizations. His research usually combines model building, laboratory experiments, and development of prototypes like Spider and kMail. Dov serves as Senior Editor for *MIS Quarterly* and is on the editorial boards of *Journal of AIS, Information and Organizations,* and *Internet Research*. His publications appear in journals such as *Management Science, MISQ, Organization Science, Communications of the ACM*, and in more specific journals of HCI such as *IJHCS, Behaviour and Information Technology, Computers in Human Behavior* and *IEEE Transactions*.

Marilyn Tremaine is professor in the Information Systems department at NJIT. Prior to NJIT, she was professor of Computer and Information Systems at Drexel University and Professor of Computer Science at the University of Toronto. She also worked as Vice President of Research and Development for a microcomputer software company and as Senior Research Scientist at the EDS Center for Applied Research. Her current research focuses on the design and development of multimodal interfaces to be used in collaborative settings. Dr. Tremaine has been active in the ACM SIGCHI community serving as VC of Communications, Chair of the Advisory Council, VC of Finance and SIGCHI Chair. She also chaired CHI'86, CSCW'92, and ASSETS'2000. She serves on six technical journal editorial boards and four to five conference program committees annually.