

# Context-Aware Systems—The ‘Right’ Information, at the ‘Right’ Time, in the ‘Right’ Place, in the ‘Right’ Way, to the ‘Right’ Person

Gerhard Fischer

Center for LifeLong Learning & Design (L3D)  
Department of Computer Science and Institute of Cognitive Science  
University of Colorado, Boulder USA  
[gerhard@colorado.edu](mailto:gerhard@colorado.edu)

## ABSTRACT

Based on the assumption that the scarce resource for many people in the world of today is not information but human attention, the challenge for future human-centered computer systems is not to deliver more information “to anyone, at anytime, and from anywhere,” but to provide “*the ‘right’ information, at the ‘right’ time, in the ‘right’ place, in the ‘right’ way to the ‘right’ person*”.

This article develops a *multidimensional framework* for *context-aware systems* to address this challenge transcending existing frameworks that limited their concerns to particular aspects of context-awareness and paid little attention to potential pitfalls. The framework is based on insights derived from the development and assessment of a variety of different systems that we have developed over the last twenty years to explore different dimensions of context awareness.

Specific *challenges, guidelines, and design trade-offs (promises and pitfalls)* are derived from the framework for designing the next generation of context-aware systems. These systems will support advanced interactions for assisting humans (individuals and groups) to become more knowledgeable, more productive, and more creative by emphasizing context awareness as a fundamental design requirement.

## Keywords

context awareness, socio-technical systems, information access and delivery, adaptive and adaptable systems, task models, user models, design intent, filter bubbles, serendipity, promises, pitfalls

## 1. INTRODUCTION

The scarce resource for most people is not information but *human attention* [38]. The next generation of human-centered computational environments requires *context-aware socio-technical systems* [14] that have some awareness of the tasks users are performing and some understanding of the knowledge background of individual users and their location in the world. The basic challenge addressed by this paper was articulated by Suchman who argued convincingly that “*the interaction between people and computers requires essentially the same interpretive work that characterizes interaction between people, but with fundamentally different resources available to the participants*

[39]. Context awareness increases the resources on which systems can rely to become more *human-centered*.

This paper first defines the concept of context and discusses problems that *context awareness* can address. It describes systems that we have designed, built, and evaluated for exploring different dimensions of context. Derived from these system-building efforts, a multidimensional framework is articulated that serves to identify research challenges, guidelines, and design trade-offs for future *context-aware systems*.

## 2. CONTEXT-AWARE SYSTEMS

### 2.1 Defining Context

The interaction between humans and computers in *socio-technical systems* takes place in a certain context referring to the physical and social situation in which computational devices and environments are embedded. The context is determined by (1) the people involved (including their background knowledge and their intentions), (2) the objective of the interaction (including the tasks to be carried out), and (3) the time and place where the interactions occur. An important aspect of context-aware systems is how the information representing the context is *obtained* [10]:

- In today’s world of ubiquitous and pervasive computing many activities that people engage in take place *inside a computational environment* and can be tracked and analyzed by software components (providing information about activities, time, and location).
- In design activities, context is not a fixed entity sensed by devices, but is emerging and is unbounded. Some context parameters may be inferred from partial designs but the intent of the designer may need to be articulated explicitly via specification components.
- Some context parameters are determined by information outside of computational environments; for example: people talking to each other around a computationally enhanced table. In such environments either sensing mechanisms are required to map external events into computational objects or people have to provide the information explicitly.

The second important aspect of context-aware systems is how the context is *represented*. A particular challenge often consists to infer higher-level goals from low-level observed operations [11]. If the users have to create context information explicitly, the question is who of the stakeholders is willing to do so.

The third important aspect of context-aware systems is for what objectives and purposes the context information is *used for*. Rather than creating an information overload problem, the context should be used to say “*the ‘right’ information, at the ‘right’ time, in the ‘right’ place, in the ‘right’ way to the ‘right’ person*”.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AVI ‘12, May 21–25, 2012, Capri Island, Italy

Copyright © 2012 ACM 978-1-4503-1287-5/12/05... \$10.00

Specific research and developed efforts have explored *particular aspects* of context — for example:

- *Location-based information systems* [8] have focused on a narrow notion of context: how to capture location automatically by hardware and software sensors.
- *Recommender systems* [23] have explored techniques for recommending various products or services to individual users based on the knowledge of users' tastes and preferences as well as users' past activities (such as: previous purchases, previous articles read, previous search commands issued).
- *Ambient Intelligence research* [18] has analyzed environments with many embedded devices where these devices can recognize the situational context of users and exploit the additional information for personalization and customization.
- Most of our work (to be described in the following sections) has been focused on *design activities, high-functionality environments, and learning environments*. We have explored *unique aspects of context aware systems* such as: design intent, specification components, critiquing systems, information access and delivery, intrusiveness, and the synergy between adaptive and adaptable components.

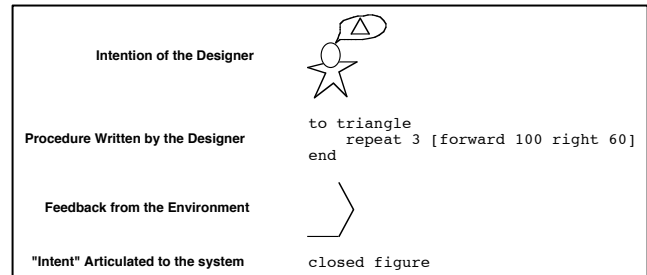
## 2.2 Problems Addressed by Context-Aware Systems

**Information Overload and Human Attention.** Design representations for a world suitable in which the scarce factor is information may be exactly the wrong ones for situations in which the scarce factor is *human attention* [38]. In a world in which people suffer from information overload, information needs to be presented that is relevant to the task at hand and tuned to the background knowledge of the user [11].

**Differentiating Contexts.** In specific situations, different aspects of context matter. Some context parameters are *easy to capture* (and can often be derived automatically as a side effect from activities); for example: (1) *who* (when a login is required); (2) *where* (when devices have embedded GPS chips); and (3) *when* (when a clock provides time stamps). Other parameters are *more difficult to capture* (and often need to be explicitly articulated); for example: (1) the *concepts* with which individual people are familiar with; (2) the *intentions* which guide their design activities; and (3) the *social relationships* and *local cultures* that determine people's interest and behavior.

**Unarticulated Design Intent.** In design, a large fraction of context-relevant information cannot be inferred from the environment because the context resides outside the environment, it is unarticulated, or it exists only in the head of a designer. Figure 1 illustrates this problem with a simple example. A

designer wants to write a simple LOGO procedure to draw an equilateral triangle [31]. The written procedure, however, draws a different figure when the instructions are executed. The resulting figure is not “wrong” per se; it is only wrong with respect to the unarticulated intent that exists only in the designer’s mind. Without access to this intent, a system is unable to detect that a problem exists. If the programming environment would allow users to articulate intentions explicitly (e.g., the intent articulation to draw a “*closed figure*” in Figure 1), and the designer was willing to do so, the additional context could be used to identify a breakdown situation (a mismatch between the intentions and the instructions given) and provide the designer with feedback and opportunities for reflection and learning (for example: understanding the difference between internal and external angles that may have caused the discrepancy between the intentions and the procedure in Figure 1).



**Figure 1: Creating Context by Articulating Intentions**

**Information Access and Delivery.** In high functionality environments, users are often unaware of a large percentage of the functionality. Explicit queries and search commands from the user do not allow exploring system functionality for which the user does not even know that it exists (i.e.: to ask a question, one must know what is not known). Systems that volunteer arbitrary chosen pieces of information (examples of such systems being: the “Did you know (DYK)” program [30] or several “Tip of the Day” components embedded in Microsoft Word and Google Earth) are not found useful by most people and most users turn them off despite their potential for interesting serendipitous encounters [25; 34]. Table 1 summarizes the major properties associated with information *access* and *delivery* systems.

## 3. SCENARIOS AND EXEMPLARY DEVELOPMENTS OF CONTEXT-AWARE SYSTEMS

Context awareness has been a desirable objective recognized for a long time and anticipated in early visionary developments including: (1) *Bush’s Memex* [6] included trails that would leave information behind that other users could exploit; and (2) the “*Read Wear and Edit Wear*” system [19] used a graphical user

**Table 1: Information Access (“Pull”) and / or Delivery (“Push”)**

	access (“pull”)	delivery (“push”)
<b>examples</b>	browsing, search engines, bookmarks, passive help systems	“Tip of the Day”, broadcast systems, critiquing, active help systems
<b>strengths</b>	non-intrusive, user controlled	serendipity, creating awareness for relevant information, rule-enforcement
<b>weaknesses</b>	task relevant knowledge may remain hidden because users can not specify a query	intrusiveness, too much decontextualized information
<b>major system design challenges</b>	supporting users in expressing queries, better indexing and search algorithms	context awareness (intent recognition, task models, user models, relevance to the task-at-hand)

interface to increase the back talk of the artifact by facilitating a reflective conversation with artifacts [36]. The challenge has been to translate these ideas and prototypes into practice. The following developments describe not only ideas but they are implemented systems that have been evaluated.

**Active Help Systems: Incremental Learning of High-Functionality Environments.** If people know something about a topic, they can use search to find it; but if they do not know something exists, they cannot search for it. *Information delivery systems* help people become aware of things that they do not know that they exist; but they suffer from the shortcoming that without context awareness they often throw a piece of decontextualized information at users that is in most cases irrelevant to the users' activities. We have explored active help systems (as examples of information delivery systems) for incrementally learning complex editing tools [15] and large reuse libraries [44].

Based on the analysis of users' understanding of numerous high functionality environments (including UNIX, Microsoft-Word, digital cameras, etc.), we provided empirical evidence for different knowledge levels as illustrated by Figure 2:

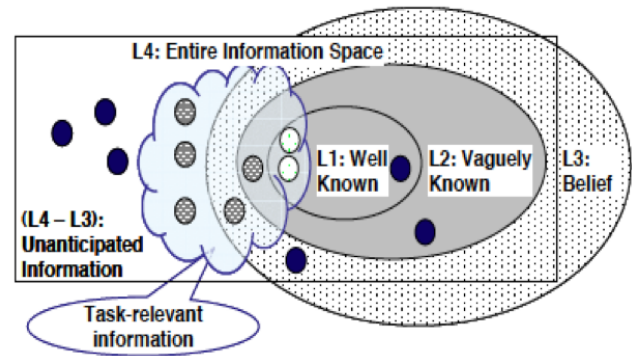
- the ovals L1, L2, and L3 represent the models held by the user: L1 is the domain of functionality with which a user is familiar and uses frequently; L2 is the domain of functionality which the users uses occasionally and passive help systems provide adequate support for this domain; and L3 is the mental model which the user has of the system;
- the square L4 represents the actual system; as the diagram indicates there are two areas of particular interest: (1) L3 $\wedge$ -L4 representing functionality which the user assumes it will exist but does not exist; and (2) L4 $\wedge$ -L3 representing functionality existing in the system but the user has no awareness about it;
- the cloud represents the information needed for the *inferred* task-at-hand (it is shown with fuzzy boundaries because the system may have only an incomplete understanding of it).

Based on this differentiation, context-aware information delivery systems can make the following decision: the information structures associated with (1) the black dots are not relevant for the task-at-hand and should therefore not be delivered; (2) the white dots inside the cloud should also not be delivered because they are already known; (3) the shaded dots are task-relevant and not known and therefore should be delivered.

**Domain-Oriented Design Environments (DODEs).** DODEs [9] support the following dimensions of context-awareness (an example for the domain of kitchen design is shown in Figure 3):

- the *application domain* in which the design activity takes place including extensive knowledge bases of components and existing examples;
- the *artifact* under construction (shown in the "Work Area" pane in Figure 3) as an indication of the task-at-hand;
- the users' intentions and goals as articulated with a *specification component* [29] (not shown in Figure 3) to allow them to enrich the description of their tasks for assistance in debugging a problematic situation (as shown in Figure 1).

**Critiquing Systems.** Human understanding in design evolves through a process of critiquing existing knowledge and consequently expanding the store of design knowledge. Critiquing is a dialog in which the interjection of a reasoned opinion about a product or action triggers further reflection on or changes to the artifact being designed. Critical analyses of our early stand-alone



**Figure 2: Computing User- and Task-relevant Information Delivery in a High-Functionality Environment**

critiquing systems [16] combined with empirical evaluations, led us to realize that the challenge in building critiquing systems is not simply to provide feedback but to exploit context awareness to say "*the 'right' information, at the 'right' time, in the 'right' place, in the 'right' way, to the 'right' person*". This finding provided the rationale to embed computer-based critiquing systems in DODEs (see the pane "Messages" in Figure 3 which displays two critiquing messages). Embedded critics can support context-awareness in the following dimensions: (1) they increase the designer's understanding of design situations by pointing out problematic situations in the design process; (2) they support the integration of problem framing and problem solving by providing a linkage between the design specification and the design construction; (3) they help designers access relevant information in the large information spaces provided by the design environment; and (4) with an explicit model of the designer's intentions for a *particular* design, critics can be selectively enabled based on these intentions and provide less intrusive and more relevant advice.

**Cultures of Participation.** Many of the most pressing problems facing the world today are systemic problems transcending the individual human mind. Solving these problems requires social structures that enable groups of people to share knowledge and resources in support of collaborative design, working, and learning [12]. We need to invent, design, and assess computationally enabled and enhanced social organizations by bringing together people who each know something but do not know other things. Exploiting this "symmetry of ignorance" will transcend support for individuals and foster a social environment that exploits the synergy of many by taking advantage of the "wisdom of crowds" [40]. In such environments, mutual awareness is important for collaborative efforts by supporting and integrating distributed and interdependent activities [35].

Cultures of participation require the *co-design* of social and technical systems. They need awareness models and concepts that not only focus on the artifact but also exploit the social context in which the systems will be used, they must allow participants to act as information seekers and helpers, and must foster relationships and give rewards in order to support all participants. Peer-support communities (representing one example of a culture of participation [17]) often suffer from too much information in the abstract and not enough information to address specific problems (as characterized by the slogan "*If only HP knew what HP knows*" [37]). Elements to support context-awareness in peer-support communities include

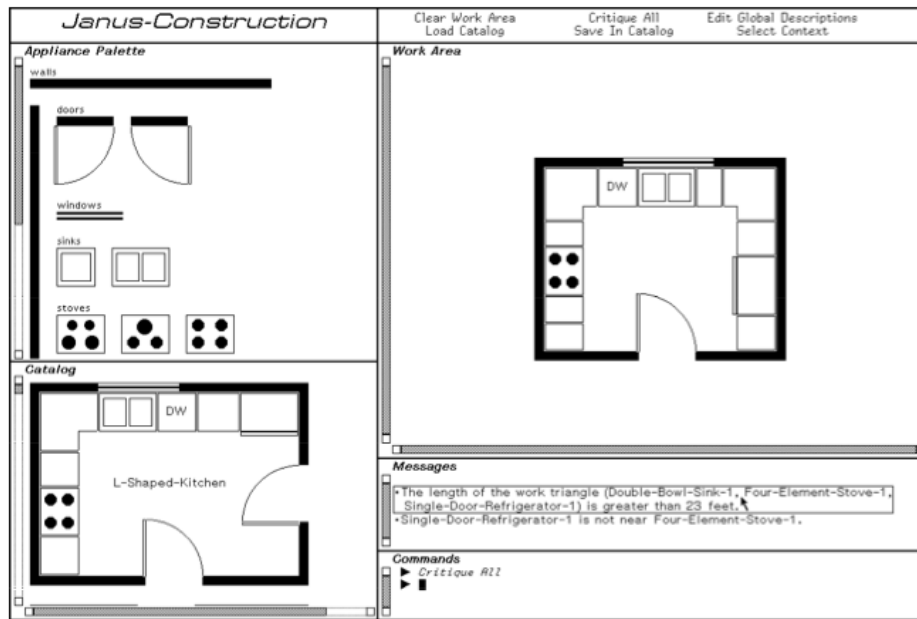


Figure 3: A Domain-Oriented Design Environment Supporting Kitchen Designers

- for the *information seeker*: (1) Who has the knowledge I need? (2) Who is willing to help me at this moment? (3) Is the information reliable?
- for the *information provider*: (1) Who might be interested in this information? (2) How should I represent the information that others will understand it? (3) How do I select the right level of intrusiveness?

#### 4. A MULTIDIMENSIONAL FRAMEWORK

Figure 4 provides an overview of our multidimensional framework for context-aware systems and indicates how the different parts of the papers are related to each other.

**The 5 Rs.** The system developments described in the previous section addressed how to provide “*the ‘Right’ Information, at the ‘Right’ Time, in the ‘Right’ Place, in the ‘Right’ Way to the ‘Right’ Person*”. The word ‘right’ is put into quotes because in most cases there is no simple ‘right’ or ‘wrong’ (see Section 5). The individual dimensions can be characterized briefly as follows:

- the ‘right’ information* requires task modeling and can be inferred from partial constructions in design, from interests derived from previous actions (e.g.: books bought, movies watched) or described via specification components [10];
- the ‘right’ time* addresses intrusiveness of information delivery (e.g.: when to notify a user about the arrival of a e-mail message, when to critique a user about a problematic

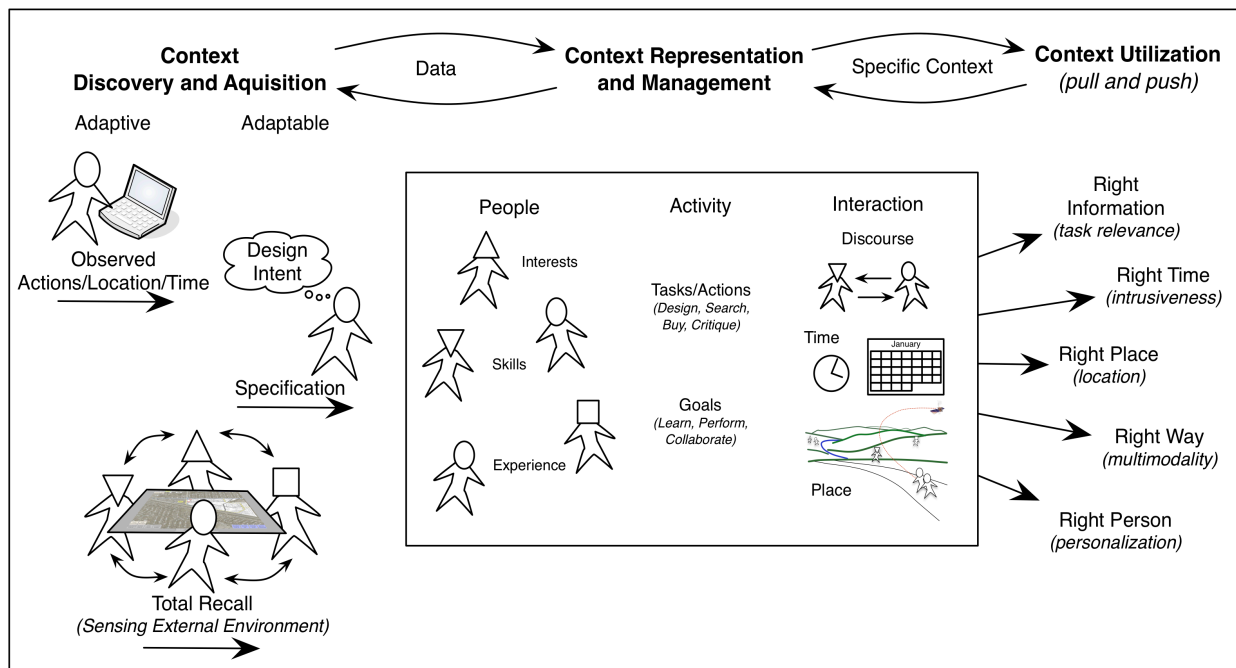


Figure 4: Multidimensional Framework for Context-Aware Systems



**Table 2: A Comparison between Adaptive and Adaptable Systems**

	Adaptive	Adaptable
<b>definition</b>	dynamic adaptation by the system itself to current task and current user	users change the functionality of the system
<b>knowledge</b>	contained in the system; projected in different ways	knowledge is extended by users
<b>strengths</b>	little (or no) effort by users; no special knowledge of users is required	users are in control; users know their tasks best
<b>weaknesses</b>	users often have difficulties developing a coherent model of the system; loss of control	users must do substantial work; complexity is increased (users need to learn adaptation components); systems may become incompatible
<b>mechanisms required</b>	models of users, tasks, and dialogs; incremental update of models	support for end-user modifiability and development
<b>application domains</b>	active help systems, critiquing systems, recommender systems	end-user modifiability, tailorability, design in use, meta-design

design decision); it requires balancing the costs of intrusive interruptions against the loss of context-sensitivity of deferred alerts [Horvitz et al., 1999];

- *the ‘right’ place* takes location-based information into account [8];
- *the ‘right’ way* differentiates between multi-model representations; e.g.: by using multimedia channels to exploit different sensory channels is especially critical for users who may suffer from some disability [7];
- *the ‘right’ person* requires user modeling; e.g.: as it is exploited in recommender systems and in intelligent tutoring systems [11; 27].

**Capturing Context.** Context can sometimes be sensed (requiring little work done by the user; e.g.: buying a book online) or it has to be articulated and described by the users (e.g.: specification components of design intent or rationale for design decisions). This raises the interesting challenges: (1) How can we capture the larger (often unarticulated) context of what users are doing (especially beyond the direct interaction with the computer system)? (2) How can we increase the “richness of resources” available for computer programs to understand what they are told about their users and to infer from what they are observing their users doing (inside the computational environment and outside) [20]?

Ubiquitous computing [42], embedded communication [33], and usage data [19] make an attempt to reduce the unnecessary separation of computational artifacts from the physical objects they represent and from the discussions surrounding them (this separation created computational environments that are “deaf, blind, and quadriplegic agents” [5]). History and interaction patterns are needed to document how artifacts were developed and which actions and contributions individual users have made.

**Adaptive and Adaptable Systems: Support different modeling techniques.** Many user modeling approaches failed because they relied too much on *one* specific technique. There is evidence [41] that substantial leverage can be gained by *integrating implicit modeling* (with adaptive components) and *explicit modeling* (with adaptable components) (see Table 2).

A simple example to illustrate the need to integrate adaptive and adaptable components is the “AUTO-CORRECT” feature in Microsoft-Word. This feature automatically detects and corrects misspelled words and incorrect capitalization (e.g.: (1) “hte” is transformed into “the” and (2) “EHR” into “HER”). While the first transformation is wanted, the second one is not what the users wants in a context where “EHR” is used as an abbreviation for an organizational unit called “Education and Human Resources”. The designers of “AUTO-CORRECT” recognized the limitations of determining the context in the abstract and they provided tools for users to adapt the feature to specific situated needs including: the deletion and addition of individual entries and/or to turn off the feature altogether.

**Meta-Design.** Context-awareness requires that users have usable and useful tools (such as adaptable components (see Table 2) and specification components for design intent (see Figure 1)) to inspect, modify and create the information structure that determine the context. Recognizing this need for end-user development and control (as illustrated by the examples of our work described in earlier sections) has led us to the development of *meta-design* [13]. Meta-design supports the co-creation of artifacts: designers at design time create contexts, seeds, and tools that empower users at use time to create additional content, modify tools, and improve and co-create the design.

**Enhancing Learning Environments with Context Aware Systems.** The differentiation between design and use time is an important distinction for different classes of context-aware systems supporting learning [26]. For example: in *intelligent tutoring systems* much more is determined and known at design time about the interactions taking place at use time compared to interactive learning environments or domain-oriented design environments in which users have much more flexibility and control what they would like to do. Interactive learning environments therefore create very different demands in exploiting context to identify and support opportunities for learners than intelligent tutoring systems. These demands are addressed by active help systems, critiquing systems (see Figure 3), and by identifying *zones of learnability* [22] that address “the ‘right’ way” dimension by identifying tasks and information representation that are both comprehensible and challenging for an individual learner. One way to achieve this objective is using

Latent Semantic Analysis to determine the appropriate learning materials and their representations [43].

## 5. IMPLICATIONS FOR FUTURE HUMAN-CENTERED COMPUTING

**Research and Design Challenges.** The arguments and examples described in earlier sections support the basic assumption that context is more than a fixed entity that can be inferred from sensors in a physical environments (e.g.: to identify “the ‘right’ place” [8] or that can be restricted to user modeling techniques in environments in which users are limited to activities envisioned in detail at design time.

The examples (in Section 3) and the multidimensional framework (in Section 4) provide requirements for addressing research and design challenges for future context aware systems including:

- how to *identify and infer user goals* from low-level interactions?;
- how to *integrate different modeling techniques* (e.g.: adaptive and adaptable components; involve users in explicit activities in order to derive context relevant information from them)?;
- how to capture the *larger and unarticulated context* for understanding what users are doing? This challenge has become simultaneously (1) *more tractable* as human beings do more tasks *inside* a computational environments and (2) *more demanding* as cyber-physical systems, augmented reality, pervasive computing transcend desktop environments and create context dimensions grounded in the real world where we live, work, learn, and collaborate;
- how to assess the *costs* (not only in monetary dimensions, but in intellectual efforts and time) of creating context(e.g.: creating machines interpretable representations in the Semantic Web, providing design rationale, tagging and rating of an artifact, curation of large information repositories);
- how these activities can be distributed among communities, and how the contributors will be rewarded for their additional efforts [45].

**Promises and Pitfalls.** Context awareness socio-technical environments represent a “Faustian Bargain” (as every powerful, far-reaching, and transformational development does). Some of the important *design trade-offs* to be understood and carefully evaluated (whether they represent promises or pitfalls) are:

- *Filter Bubbles and Group Think:* As web companies exploit context awareness to tailor their services (including news and search results) to people’s personal tastes, there is a unintended consequence: recipients get trapped in “*filter bubbles*” [32], a unique universe of information computed by algorithms exploiting context awareness based on users previous actions and behaviors. The promises of this approach is that it reduces the information overload, the drawback is that users do not get exposed to information that could challenge or broaden their worldview and that unexpected encounters with different, topics and opinions are eliminated. Filters bubbles may lead to *groupthink* [21], a psychological phenomenon occurring within groups of people. The negative consequences of groupthink are: the loss of (1) individual creativity, (2) uniqueness, and (3) independent thinking and a tendency to minimize conflict and reach a consensus decision without critical evaluation of alternative ideas or viewpoints. Unexpected encounters with different themes and opinions are of central importance for democracy, freedom, and tolerance.

- *Making information relevant to the task at hand versus serendipity:* As argued in the earlier sections of the paper, context aware systems should make information relevant to the task at hand and should expose and tailor tool functionality to users’ needs and background knowledge (see Figure 2). But everyone has experienced that serendipitous encounters (1) can acquaint users with pieces of functionality they find truly useful and (2) have a fundamental impact on people lives and decision making.

These two objectives represent a design-tradeoff that cannot be resolved (at least not by distant designers or algorithms). Users should have interaction mechanisms at their disposal (e.g.: buttons to turn a feature on or off, sliders for setting preferences, links pointing to more extensive information for follow-ups) allowing them to select their own personal, situation- and time-dependent *best mix* of these design trade-offs.

- *Intrusiveness* (“the ‘right’ time” dimension) represents another feature for which no ‘right’ solution exists in the abstract. Users (and not the designers) should control how they want to be notified of the arrival of an email message or when a computational critic should fire.
- *Remembering and Forgetting:* Another design trade-off exists between remembering and forgetting [1]. One aspect of context aware systems is to support a new synergy between human memories and digital memories. The promise is that we are able to capture and record an arbitrary amount of information about our lives. Context aware mechanisms will support users to access and find relevant information at the right time for remembering things. Information delivery will confront users with relevant information even if they do not know what they are looking for (see the area  $L4 \wedge \neg L3$  in Figure 2). The pitfall of “*total recall*” [3] is to miss the virtues associated with *forgetting* [1; 2; 28] which allows us to avoid that by being immersed in the past we have no resources left to explore the present. As the digital world has entered an age of immense spaces for storing information, users are not forced anymore to clean up because their hard disk is full. This activity has become a choice and we need criteria to influence and guide these choices.
- *Privacy:* We live in a world where more and more events take place and are tracked in some computational environment and recorded for context awareness. To name a few examples: telephone calling cards, shopping cards at supermarkets, book ordering at electronic book stores, websites visited, active badges worn. It will be a major challenge to find ways to avoid misuses, either by not allowing companies to collect this information at all or by finding ways that the individual users have control over these user models [24].

**Control.** All of the design trade-offs offer promises and pitfalls and the question is who has the control how a context aware system will be configured. Systems will guess incorrectly and perform hidden changes that users do not like. In most cases, they often lack the possibility or at least the transparency for users to turn off “smart” features, which can get more in the way than help (as illustrated briefly with the simple example of “Auto Correct” earlier). As argued at the beginning, context aware systems will only be partially aware of the total problem-solving process their human partners engage in. Whereas these drawbacks of context aware systems may be only annoying in word processors or recommendation systems for books and movies, they are

unacceptable in other socio-technical environments (such as computational agents in the cockpit supporting pilots) [4]. To avoid misunderstanding and miscommunication, context-aware systems should provide tools and represent information in ways that empower rather than diminish users, giving users the control over their tasks and information needs.

## 6. CONCLUSIONS

Context awareness has been an interesting theoretical objective and construct for the design of socio-technical system for a long time. Recently some dimensions of context awareness (e.g.: recommendations buying books and renting movies, taking location into account) have become prominent features of some widely used environments. This paper (based on selected system developments and assessments over the last two decades) has tried to show that context awareness is a multidimensional objective and that additional mechanisms (e.g.: specification components, adaptable components to put owners of problems in charge rather than the original designers or algorithms) are necessary to exploit the full potential of context awareness.

The ultimate objective of human-centered computing is that it will serve the benefit of users (acting as individuals or in teams): by empowering them, by improving their experience, by making them more productive and creative, and by integrating social and technical dimensions.

To understand the true impact of context awareness represents not only a research theme of the past, but it remains an important challenge for the future. Many interesting challenges are ahead of us: how to understand the *design trade-offs*, the *promises*, and the *pitfalls* (1) between adaptive and adaptable systems, (2) between information delivery (“push”) and information access (“pull”) technologies, (3) between contextualized information representations and serendipity, (4) how to avoid that people get cocooned in their filter bubbles; and (5) how to exploit context-awareness in cultures of participation.

## 7. ACKNOWLEDGEMENTS

I thank the members of the Center for LifeLong Learning & Design at the University of Colorado, who have made major contributions to the ideas and developments described in this paper. The research was supported in part by grants from the National Science Foundation including: (1) IIS-0613638 “A Meta-Design Framework for Participative Software Systems”; (2) IIS-0968588 “Energy Sustainability and Smart Grids: Fostering and Supporting Cultures of Participation in the Energy Landscape of the Future”; (3) OCI-1028017 “Transformative Models of Learning and Discovery in Cultures of Participation; and (4) IIS-1111025 “Theoretical Frameworks and Socio-Technical Systems for Fostering Smart Communities in Smart Grid Environments”.

## 8. REFERENCES

- [1] Bannon, L. J. (2006) "Forgetting as a Feature, Not a Bug: The Duality of Memory and Implications for Ubiquitous Computing," *CoDesign*, 2(1), pp. 3-15.
- [2] Barua, D., Kay, J., Kummerfeld, B., & Paris, C. (2011) "Theoretical Foundations for User-Controlled Forgetting in Scrutable Long Term User Models" In *Proceedings on Ozchi 2011*, ACM, Canberra, pp. 40-49.
- [3] Bell, G. & Gemmell, J. (2009) *Total Recall*, Dutton, New York, NY.
- [4] Billings, C. E. (1991) *Human-Centered Aircraft Automation: A Concept and Guidelines*, NASA Technical Memorandum, Report No. 103885, NASA Ames Research Center.
- [5] Bobrow, D. G. (1991) "Dimensions of Interaction," *AI Magazine*, 12(3), pp. 64-80.
- [6] Bush, V. (1945) "As We May Think," *Atlantic Monthly*, 176(1), pp. 101-108.
- [7] Carmien, S., Dawe, M., Fischer, G., Gorman, A., Kintsch, A., & Sullivan, J. F. (2005) "Socio-Technical Environments Supporting People with Cognitive Disabilities Using Public Transportation," *Transactions on Human-Computer Interaction (ToCHI)*, 12(2), pp. 233-262.
- [8] Dey, A. K., Abowd, G. D., & Salber, D. (2001) "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications," *Human-Computer Interaction*, 16(2-4), pp. 97-166.
- [9] Fischer, G. (1994) "Domain-Oriented Design Environments," *Automated Software Engineering*, 1(2), pp. 177-203.
- [10] Fischer, G. (2001) "Articulating the Task at Hand and Making Information Relevant to It," *Human-Computer Interaction Journal, Special Issue on "Context-Aware Computing"*, 16, pp. 243-256.
- [11] Fischer, G. (2001) "User Modeling in Human-Computer Interaction," *User Modeling and User-Adapted Interaction (UMUAI)*, 11(1), pp. 65-86.
- [12] Fischer, G. (2011) "Understanding, Fostering, and Supporting Cultures of Participation," *ACM Interactions* XVIII.3 (May + June 2011), pp. 42-53.
- [13] Fischer, G. & Giaccardi, E. (2006) "Meta-Design: A Framework for the Future of End User Development" In H. Lieberman, F. Paternò, & V. Wulf (Eds.), *End User Development*, Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 427-457.
- [14] Fischer, G. & Herrmann, T. (2011) "Socio-Technical Systems: A Meta-Design Perspective," *International Journal of Sociotechnical and Knowledge Development*, 3(1), pp. 1-33.
- [15] Fischer, G., Lemke, A. C., & Schwab, T. (1985) "Knowledge-Based Help Systems" In L. Borman, & B. Curtis (Eds.), *Proceedings of Chi'85 Conference on Human Factors in Computing Systems*, ACM, New York, pp. 161-167.
- [16] Fischer, G., Nakakoji, K., Ostwald, J., Stahl, G., & Sumner, T. (1998) "Embedding Critics in Design Environments" In M. T. Maybury, & W. Wahlster (Eds.), *Readings in Intelligent User Interfaces*, Morgan Kaufmann, San Francisco, pp. 537-559.
- [17] Gorman, A. & Fischer, G. (2009) "Toward an Analytic Framework for Understanding and Fostering Peer-Support Communities in Using and Evolving Software Products" In *Proceedings of the International Conference Communities and Technologies (C&T'2009) Penn State University, June* pp. 1-9.
- [18] Gross, T. (2010) "Towards a New Human-Centred Computing Methodology for Cooperative Ambient Intelligence," *Journal of Ambient Intelligence and Humanized Computing*, Springer, Berlin, 1, pp. 31-42.
- [19] Hill, W. C., Hollan, J. D., Wroblewski, D., & McCandless, T. (1992) "Edit Wear and Read Wear" In P. Bauersfeld, J. Bennett, & G. Lynch (Eds.), *Proceedings of Chi'92 Conference on Human Factors in Computing Systems*, ACM, New York, pp. 3-9. Available at: <http://www.acm.org/pubs/articles/proceedings/chi/142750/p3-hill/p3-hill.pdf>.
- [20] Horvitz, E., Kadie, C., Paek, T., & Hovel, D. (2003) "Models of Attention in Computing and Communication: From Principles to Applications," *Communications of the ACM*, 46(3), pp. 52-59.

- [21] Janis, I. (1972) *Victims of Groupthink*, Houghton Mifflin, Boston.
- [22] Kintsch, W. (2002) "The Potential of Latent Semantic Analysis for Machine Grading of Clinical Case Summaries," *Journal of Biomedical Informatics*, 35(1), pp. 3-7.
- [23] Knijnenburg, B. P. & Kobsa, A. (2012) *Making Decisions About Privacy: Information Disclosure in Context-Aware Recommender Systems*, Report UCI-ISR-12-1, Institute of Software Research, University of California, Irvine, CA.
- [24] Kobsa, A. (2007) "Privacy-Enhanced Personalization," *Communications of the ACM*, 50(8), pp. 24-33.
- [25] Krotoski, A. (2011) *Serendipity, Aka "Accidental Discovery": Does the Web Help or Hurt?*, available at <http://dmlcentral.net/blog/aleks-krotoski/serendipity-aka-accidental-discovery-does-web-help-or-hurt>.
- [26] Luckin, R. (2010) *Re-Designing Learning Contexts* Routledge, London.
- [27] Martin, B., Mitrovic, A., Koedinger, K. R., & Mathan, S. (2011) "Evaluating and Improving Adaptive Educational Systems with Learning Curves," *UMUAI*, 21(3), pp. 249-283.
- [28] Mayer-Schönberger, V. (2009) *Delete — the Virtue of Forgetting in the Digital Age*, Princeton University Press, Princeton, NJ.
- [29] Nakakoji, K. (1993) *Increasing Shared Understanding of a Design Task between Designers and Design Environments: The Role of a Specification Component*, Ph.D. Dissertation, University of Colorado at Boulder.
- [30] Owen, D. (1986) "Answers First, Then Questions" In D. A. Norman, & S. W. Draper (Eds.), *User-Centered System Design, New Perspectives on Human-Computer Interaction*, Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, pp. 361-375.
- [31] Papert, S. (1980) *Mindstorms: Children, Computers and Powerful Ideas*, Basic Books, New York.
- [32] Pariser, E. (2011) *Beware Online "Filter Bubbles" (Ted Talk)*, available at [http://www.ted.com/talks/eli\\_pariser\\_beware\\_online\\_filter\\_bubbles.html](http://www.ted.com/talks/eli_pariser_beware_online_filter_bubbles.html).
- [33] Reeves, B. N. (1993) *Supporting Collaborative Design by Embedding Communication and History in Design Artifacts*, Ph.D. Dissertation, University of Colorado at Boulder.
- [34] Roberts, R. M. (1989) *Serendipity: Accidental Discoveries in Science*, John Wiley & Sons, Inc., New York.
- [35] Schmidt, K. (2002) "The Problem with 'Awareness'." *Computer Supported Cooperative Work: The Journal of Collaborative Computing*, 11(3-4), pp. 285-298.
- [36] Schön, D. A. (1983) *The Reflective Practitioner: How Professionals Think in Action*, Basic Books, New York.
- [37] Sietloff, C. G. (1999) "If Only Hp Knew What Hp Knows: The Roots of Knowledge Management at Hewlett-Packard," *Knowledge Management*, 3(1), pp. 47-53.
- [38] Simon, H. A. (1996) *The Sciences of the Artificial*, third ed., The MIT Press, Cambridge, MA.
- [39] Suchman, L. A. (1987) *Plans and Situated Actions*, Cambridge University Press, Cambridge, UK.
- [40] Surowiecki, J. (2005) *The Wisdom of Crowds*, Anchor Books, New York.
- [41] Thomas, C. G. (1996) *To Assist the User: On the Embedding of Adaptive and Agent-Based Mechanisms*, R. Oldenburg Verlag, München/Wien.
- [42] Weiser, M. (1993) "Some Computer Science Issues in Ubiquitous Computing," *Communications of the ACM*, 37(7), pp. 75-83.
- [43] Wolfe, M. B., Schreiner, M. E., Rehder, R., Laham, D., Foltz, P. W., & Landauer, T. K. (1998) "Learning from Text: Matching Reader and Text by Latent Semantic Analysis," *Discourse Processes*, 25, pp. 309-336.
- [44] Ye, Y. & Fischer, G. (2002) "Supporting Reuse by Delivering Task-Relevant and Personalized Information," *Proceedings of 2002 International Conference on Software Engineering (ICSE'02)*, Orlando, FL, pp. 513-523.
- [45] Ye, Y., Nakakoji, K., & Yamamoto, Y. (2008) "Understanding and Improving Collective Attention Economy for Expertise Sharing" In Z. Bellahsene, & M. Leonard (Eds.), *Proceedings of Caise (International Conference on Advanced Information Systems Engineering)*, Springer, Berlin, pp. 167-181.