

Interactionist AI and the promise of ubicomp, or, how to put your box in the world without putting the world in your box

Lucian Leahu, Phoebe Sengers
Cornell University
301 College Avenue
Ithaca, NY 14850
{lleahu, sengers}@cs.cornell.edu

Michael Mateas
University of California, Santa Cruz
1156 High Street M/S SOE3
Santa Cruz, CA 95064
michaelm@cs.ucsc.edu

ABSTRACT

In many ways, the central problem of ubiquitous computing – how computational systems can make sense of and respond sensibly to a complex, dynamic environment laden with human meaning – is identical to that of Artificial Intelligence (AI). Indeed, some of the central challenges that ubicomp currently faces in moving from prototypes that work in restricted environments to the complexity of real-world environments – e.g. difficulties in scalability, integration, and fully formalizing context – echo some of the major issues that have challenged AI researchers over the history of their field. In this paper, we explore a key moment in AI’s history where researchers grappled directly with these issues, resulting in a variety of novel technical solutions within AI. We critically reflect on six strategies from this history to suggest technical solutions for how to approach the challenge of building real-world, usable solutions in ubicomp today.

Author Keywords

AI-completeness, interactionist AI, reflective HCI

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

In the relatively short history of our field, ubiquitous computing has frequently engaged techniques, methods and approaches from Artificial Intelligence (AI) [e.g. 29]. This is hardly a surprise, since the fields share an overlapping interest in building technologies that make sense and respond in a sensible way to the complex dynamics of human environments. This common interest supports the useful adaptation of AI methods such as machine learning,

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, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

UbiComp’08, September 21-24, 2008, Seoul, Korea.
Copyright 2008 ACM 978-1-60558-136-1/08/09...\$5.00.

sensing, and decision algorithms to ubicomp systems. In addition, the special class of real-time, responsive, embedded AI systems share a particular affinity with ubicomp systems, since they are intended to be deployed in similar environments, share many overlapping requirements, and therefore face similar challenges in coming to grips with situations that are fast-changing, unpredictable, and laden with complex human meaning.

In particular, ubicomp is currently facing a series of challenges in scaling up from prototypes that work in restricted environments to solutions that reliably, robustly work in the full complexity of human environments [2,26]. These challenges echo problems AI researchers tackled as the field sought to move beyond ‘blocks-world’ solutions to build real-time systems that could work in dynamic, complex environments. In this paper, we reflect on 6 episodes from this history where researchers found novel ways to reframe core AI problems in order to develop new technical solutions. We use these episodes as a starting point for thinking through how ubicomp may be able to draw on similar forms of reframing in order to spur the development of robust, reliable, real-time ubiquitous systems today.

BACKGROUND

In order to better understand the problems AI researchers were addressing and their relevance to ubicomp, we begin by describing the recurrent difficulties AI was running into in the 1980s, and the ways in which what we will here term ‘interactionist AI’ reframed AI problems in order to move beyond these recurrent difficulties. We will then explain how these challenges are mirrored in contemporary ubicomp’s push to move out of the laboratory and into the real world.

Making AI work in the world

The history of AI, “the most colorful and controversial branch of computer science” [24, p. 54], has been marked by numerous highs followed by crashing lows in terms of success, recognition and funding. One such low was triggered by the undelivered promises of early AI research, resulting in the AI funding winter of the 1970s. As the goal

of computational intelligence proved to be harder to achieve than initially expected, in the early 1980s, a number of focal areas of interest for AI began to crystallize. These areas targeted individual intelligent capabilities: knowledge representation, planning, computer vision, natural language understanding, machine learning, etc. In what later became known as classical AI, these problems were generally tackled by identifying real-world entities relevant to the problem, formally representing them using symbolic representations, and using rules and inference to manipulate said representations to derive answers [48]. Based on the principle of divide-and-conquer, progress was to be made through independent work on each of the competencies revealed through functional decomposition; these functional modules would eventually be combined into a complete intelligent system.

Although it had appeared that this divide-and-conquer strategy would allow for measurable, incremental improvements in AI algorithms that could eventually be merged into a single system, a number of problems emerged. First, attempts to provide technical solutions to each of these problems were often overwhelmed by the amount of contextual information and intelligence each seemed to require. Incorporating more contextual information in systems by way of symbolic representation did not seem to really solve the problem, since each addition to the model would trigger a chain of additional related entities that had to be accounted for. This proliferation of considerations the algorithms needed to address led to problems of scalability in terms of computational complexity [e.g. 16]. At the same time, AI researchers discovered the complexities of maintaining consistent and accurate world models even in simplified environments [17]. Finally, although research development in AI was organized around modular functionalities, when attempts were made to combine these functionalities it was found that integration was less a matter of plug-and-play than a substantial technical achievement on its own [46].

Given the fast successes claimed by early AI in demonstrating advanced problem solving capabilities, sometimes beyond human level (e.g. playing chess), these difficulties came as a surprise. Particularly so, as some of the subproblems researchers hoped to address – such as vision or natural language understanding – are easy, even effortless for humans, yet proved to be very hard for computers. Realization gradually dawned that these functional modules were intertwined, each requiring others in order to work. As a consequence, such problems were sometimes termed *AI-complete* problems to suggest that a solution to these problems would presuppose a solution to the general problem of intelligence [33]. The notion of AI-completeness suggests that it will be very difficult, if not impossible, to develop complete, robust, and general solutions to AI-complete problems – i.e., that the challenges AI had taken on were hopeless by conventional methods.

The problems of AI-completeness became particularly obvious and challenging in the area of autonomous agents, i.e. software and robotic systems that are intended to correspond to a complete (though perhaps simplistic) animal, person, or character. By necessity, agents require functional integration. In light of the growing complexities faced by tackling AI-complete problems head on, a number of alternative approaches arose in this area to work around issues pertaining to AI-completeness, such as behavior-based AI, situated action, and believable agents. These researchers developed specific technical strategies to circumvent AI-completeness by focusing on solutions that worked in specific, situated contexts rather than solving the general problem of intelligence (this work was informed in part by Lucy Suchman's critiques of the limitations of planning algorithms, which are also well-known in HCI [44]). Because, as we will describe below, these researchers saw themselves as providing concrete, technically feasible approaches for supporting real-time, intelligent interaction with a changing environment, we will refer to these approaches as *interactionist AI*. Interactionist AI questions the assumptions of generality, rationality, and complete and correct world models underlying classical AI, focusing instead on embedding autonomous systems within specific, real-world contexts. Our goal in this paper is to reflect on the technical strategies developed by interactionist AI researchers in order to identify tactics that may be useful for ubicomp today.

For the purposes of this paper, interactionist AI is understood as historically appearing in two waves. The first wave was concerned with constructing autonomous agents, primarily robots that operate robustly in real-world environments. These first-wave systems were generally not concerned with human interpretation of the agent, but rather with robust and fault-tolerant performance in a complex environment. The second wave was, and is, concerned explicitly with human interpretation of and interaction with autonomous agents, with a focus on effective communication between agent and human. This was a natural evolution of interactionist AI as humans became part of the environment in which the autonomous agent should operate.

Both waves of interactionist AI make use of context-specific regularities in the environment to avoid AI-completeness. For example, first-wave systems avoid the need for a complete and correct theory of the physical world by leveraging assumptions and dynamical regularities of the environment to directly select actions based on sensor values. Similarly, second-wave systems avoid the need for a complete and correct theory of human communication (semantics, pragmatics, mental state), by leveraging extra-system knowledge in the minds of human participants in a specific context of interaction.

In the next section, we will describe why these lessons of interactionist AI might be particularly pertinent for the challenges ubicomp currently faces.

From AI to Ubicomp

The central insight that drove technical solutions in interactionist AI was that in moving from small-scale, partial prototypes to complete systems that could work in the real world, it can be helpful to circumvent the problem of developing full-scope, human-scale intelligence by taking advantage of the situated properties of the context of use in the technical design of the system. Because ubicomp is now facing a similar challenge to scale up from experimental components tested in controlled conditions to robust, reliable, integrated, real-world systems, there is a real potential for the solutions that proved useful for autonomous agents to map to ubicomp.

First-wave interactionist AI focused on the following three challenges: responding to dynamically changing environments, graceful degradation of behavior in the face of unexpected environmental conditions, and balancing several, often conflicting goals. This requires that agents be robust to noisy, faulty and inconsistent sensor values, provide sensible functionality even when the environment changes unexpectedly or components fail, and solve the action-selection problem in such a way as to accomplish internal goals while remaining reactive to the environment. Ubicomp systems today face similar challenges with respect to sensing [e.g. 2,26], robust and reliable operation in unexpected real-world conditions [e.g. 50,21], and real-time, flexible juggling of competing goals [e.g. 2,43,19].

Second-wave interactionist AI focuses on the following three challenges: incorporating explicit interaction with humans as part of the environment (human-in-the loop), explicitly communicating agent (system) state in a manner actionable by humans, and participating in rich social and cultural contexts. This requires that agents leverage dynamical regularities in individual and group behavior, match internal state complexity and external representation complexity, often taking advantage of representation strategies from the expressive arts, and manipulate and reference cultural codes whose full meaning is strictly greater than that represented within the system. Ubicomp systems today face similar challenges with respect to human behavioral regularities [e.g.12], expressive representation [e.g. 14], and social and cultural embedding [e.g. 45, 51]. From this description, it should be clear that the challenges and goals of interactionist AI are strikingly similar to those facing ubicomp today; and, therefore, technical solutions developed by interactionist AI researchers may prove to have relevance to these contemporary challenges. Yet if this is the case, given the existing interchanges between AI and ubicomp, why haven't these solutions already been explored? The simple answer is because interactionist AI is not currently a central theme in AI, and thus is not very visible or accessible to the ubicomp researcher.

Though first-wave interactionist AI had some clear victories in developing new forms of real-time, responsive systems, the approaches did not scale to more complex

agents with higher-level cognitive competencies. The ideological and technological battle between traditional and first-wave interactionist AI resulted essentially in a draw; neither side really attempted to scale outside their comfort zone, with symbolic partisans staying safely away from real-time responsiveness and noisy sensors, and interactionist partisans staying safely away from longer-term planning, complex inference, and natural language interaction. Robotics settled into a standard three-tier architecture, which combines interactionist and symbolic methods; interactionist methods handle real-time execution (the bottom two tiers) and symbolic methods handle long-term planning (the top tier). Though this state of détente has been productive for robotics (though integration issues remain, often limiting the useful generativity possible in the symbolic reasoning layer), by effectively ending the conversation, it has made interactionist AI's insights into technical and design strategies less visible.

Second-wave interactionist AI is still in its youth, with a small but growing number of active researchers contributing to the field. But the influence of this body of work tends to be restricted to smaller, specialized conferences, such as AI and Interactive Digital Entertainment, Intelligent Virtual Agents, and Human-Robot Interaction, rather than being visible within AI as a whole. Thus the focus on AI as a method of communication and means for structuring audience experience, while commonplace within, for example, the game AI community, is not very accessible to the ubicomp researcher.

Reflecting on the insights of interactionist AI holds value for ubicomp for the following reasons. First, although interactionist AI may not be the route to full, autonomous intelligence, its ability to generate systems of many, distributed, autonomous, robust and responsive components that communicate through the environment is directly relevant to Weiser's vision of ubiquitous computing [49]. Second, while interactionist AI's critiques of classical approaches are no longer as relevant in AI (having been absorbed into the background), they remain relevant in ubicomp because ubicomp's existing approaches towards system design and representation often resemble those of classical AI. For example, emphasis in our field, as in classical AI, is placed on incorporating systematic representations of external entities that are pertinent to the system's functionality into the system. In ubicomp, such representations revolve around users' identity, activity, location, and indeed harder to define notions, such as context [2,18,20]. As we will demonstrate below, the similarity between these representational approaches may support relatively straightforward adaptation of interactionist AI solutions to the problems of ubicomp.

Before moving on, we would like to comment briefly on the statistical revolution in AI. Machine learning (ML) has become a major approach within AI; by enabling statistical reasoning over large datasets, ML has circumvented

knowledge engineering bottlenecks that plagued classical AI. However, ML is not an approach for building complete agents, but rather provides statistical solutions for specific competencies (ML's application to activity recognition is particularly relevant in a ubicomp context). Interactionist AI, in contrast, provides strategies for building complete, integrated agents, which is why we focus on interactionist AI in this paper.

SIX STRATEGIES

As we began to explore in the previous section, interactionist AI approaches arose in reaction to recurring difficulties in extending formal, symbolic approaches to real-world, embedded systems. It sought to create systems that appeared to be intelligent, without running into the problems associated with AI-completeness. In this section, we reflect on six technical strategies developed in interactionist AI. In each case, we will describe why and how the strategy was developed, then reflect on how that strategy could be usefully adapted for the development of ubiquitous computing technologies. Strategies one through three are drawn from first-wave interactionism, strategies four through six from second-wave interactionism.

Strategy #1: Tightly integrate sensing and action in complete working systems

The first strategy we will explore was developed in the field of robotics and addressed the issues that emerge in integrating the functional components of intelligence in real-world, embodied systems. Traditionally [e.g. 37], a mobile robot's control system is decomposed into function modules such as perception (sensing), modeling (mapping sensor input to a representation of the world), planning, task execution and motor control. This *functional decomposition* involves a chain of modules, each of which solves a subproblem and passes the result along to the next module, eventually performing a motor action.

In this setting, even a very minimal working system requires that an instance of each module must be realized and that all modules must fit together. It becomes clear that in the event of faulty functionality of one of the components, the robustness of the entire system is compromised. Furthermore, alterations within a single module are limited to either 1) keeping the interface to the neighboring modules unchanged or 2) modifying the interface, which translates into propagating the changes along the module chain. As such, common maintenance operations, such as debugging and optimization, as well as ordinary upgrades like adding new sensors or responding to conflicting goals by changing the control unit (usually located in the planning module) may result in costly and time consuming changes in the entire system architecture.

In light of these difficulties, roboticist Rodney Brooks proposed an alternative called *behavioral decomposition* [9]. Instead of breaking up the problem of robotic action into functional, modular subproblems, behavioral decomposition relies on an incremental method that stacks

increasingly complex layers of behavior on top of each other. Each layer links perception to action and is responsible for a certain behavior: low level layers implement basic behaviors (e.g. avoid obstacles), while higher levels provide more complex ones (e.g. build maps of the environment and compute paths). Higher levels subsume the behavior of lower levels, while the lower levels do not know about the existence of the higher layers. Once a level is completed it is never altered: introducing new behaviors is achieved by simply adding another layer of functionality on top.

Genghis, a six-legged walking robot, exemplifies the potential of this approach [9]. Genghis integrates a large number of sensors and actuators, achieving a rich array of behaviors from basic ones such as standing-up and walking to complex actions such as following people. Genghis has no perception module and therefore does not engage in sensor fusion; instead, programmers manage conflicts between sensors only when the inconsistencies are relevant to the current behavioral situation. The robot (along with others built by Brooks' group) represents a technical proof that complex behavior can be achieved by way of many independent simple behaviors, with little or no central coordination.

Consequences for Ubicomp

Turning our attention back to ubiquitous computing, we see a similar challenge to build systems that integrate components reliably in complex real-world contexts, and a similar opportunity to build up system functionality incrementally in order to assure that experimental components actually function usefully in integrated systems.

The functional decomposition, which turned out to be problematic for integration, continues to structure much ubicomp research. For example, in context-aware systems, a common development approach explicitly separates sensing from the core of the application and aims to develop it as a stand-alone application regardless of context of use [18,27]. Instead, Brooks' work suggests explicitly structuring system architecture to reflect the intricate relationship between sensing and acting. Approaches that fundamentally rely on a deep coupling between sensing and acting represent viable alternatives for ubicomp systems in which 1) sensor fusion cannot be decoupled from the situation of use, i.e. the relevance of the sensor readings cannot be established a priori and/or may become obvious only in light of the specific, current situation of use and 2) conflicting sensor readings do not necessarily lead to different actions. By tying sensing and action together as Brooks suggest, we may be able to address technically Dourish's critique that context and action in ubicomp should be more tightly intertwined [20].

Perhaps the most significant lesson from Brooks' work is that **early component integration**, irrespective of the approach taken, can uncover opportunities as well

constraints that would otherwise go unnoticed. Specifically, it could help focus the system development by highlighting the sine qua non technical requirements, as opposed to getting slowed down by what we might ideally want, but is secondary to the bare functionality of the system. Interactionist strategies allow applications to grow organically (a frequent requirement for ubicomp system development [1]) by incrementally upgrading with new, more complex functionality without altering existing functionality.

One example of an ubicomp system that implicitly embodies such an orientation towards system design is ButterflyNet, a capture and access mobile system designed for biology fieldwork [52]. Rather than focusing on developing individual cutting-edge functionality, the designers created a fully functioning application by combining off-the-shelf, heterogeneous devices - a digital camera, and a digital handheld pen system. They designed its architecture to be easily extendable e.g. to support adding new sensors. It is quite robust and offers graceful degradation; paper notes are automatically digitized by the pen system, but in case of errors the user still has the paper version.

Strategy #2: Sense rather than represent

Both classical AI and ubicomp rely heavily on explicit, formal models of the world as well as on manipulating symbols corresponding to entities in the world in order to understand the environment and derive results that are meaningful in the context of use. In AI, this approach has proven to be quite cumbersome as it may require large amounts of storage, internal models must be continuously kept up to date (the frame problem is a classic problem in this regard [e.g. 22]) and inference can be computationally costly, all of which are problematic even in simplified environments [17].

In response to the challenges inherent in building and maintaining accurate world models, Brooks suggested a radical solution: abandon the use of sophisticated world models and instead use sensing to extract the minimum information required to drive behavior. This is in some sense a corollary of the first strategy, in that tight coupling of sensing and action is at odds with maintaining a complete world model. However, as a strategy, it is useful to pull out separately as it refocuses system designers to consider what world state can be directly exploited without modeling.

Herbert is an example of a fairly sophisticated robot built by Brooks' group that was able to engage in complex behavior such as walking around busy offices, identifying and picking up empty soda cans. It accomplished this with no internal model of the office space and no internal communication between behavior layers. Instead, Herbert, following the dictum "the world is its own best model" [10, p.5], relied solely on its sensors to continuously sense and react to the environment by following simple, opportunistic

rules (e.g. if a soda can is detected, move towards it). By relying on regularities in the world, Herbert demonstrates that real-world systems can avoid using explicit internal representations, and thus avoid the difficulties involved in maintaining such representations so that they remain true [11].

Consequences for Ubicomp

The challenges posed by representation are of relevance for ubicomp, in light of significant efforts to model context [e.g. 2,18] and critiques of their intrinsic limitations [e.g. 40,20]. World models are also becoming popular in ubicomp in an attempt to build more powerful ubicomp systems [e.g. 25]. The issues explored by interactionist AI researchers suggest there are serious technical limitations to the goal of building complex, reliable, and dynamically relevant world models. The Brooksonian rule-of-thumb – limit representation, sense instead – suggests that it might in fact be easier to achieve complex, useful behavior without building and reasoning about internal representations of context. Systems such as Herbert demonstrate that apparently intelligent, and useful, behavior can be achieved without the system having a formal or explicit understanding of what it is doing.

Strategy #3: Develop ad-hoc, situated representations

There are limitations to the complexity of systems that can be built if designers literally eliminate *all* internal representation. As interactionists reached these limitations, they began developing strategies for re-introducing representation without re-introducing all the problems of maintaining complete, correct and coherent world models.

One such alternative representation strategy is David Chapman and Phil Agre's *deictic representation* developed in their work on reactive planning [4]. Agre and Chapman developed a computer program controlling Pengi, a penguin that plays the computer game Pengo. The game takes place in an ice blocks maze inhabited by killer bees. As is typical for interactionist AI applications, this environment is highly dynamic: the bees and some of the ice blocks are constantly moving, ice blocks can be pushed by either Pengi or the bees, where pushing an ice block causes it to slide in the direction it was pushed until it hits another ice block. In order to win, Pengi must avoid being killed by bees or ice blocks, and kill all the bees by crushing them using ice blocks. Although the setting is fairly simple, the bees behavior is randomized, therefore Pengi's environment is constantly changing, often in unpredictable ways.

Pengi managed to play "a pretty decent game of Pengo. In its present state it is a little better than I am, which is to say that it wins from time to time and usually puts up a good fight" [3, p.265]. What is remarkable about Pengi is that the complex behavior it exhibits is accomplished without maintaining a complete, coherent world model. Such a world model would typically represent the entities that appear in Pengi's environment (the bees, the ice blocks and

Pengi itself) using explicit, objective representations, e.g. *bee#1* to keep track of the location of a particular bee. In contrast, deictic representations are defined in terms of the agent's situation (location, scope, etc.) and the roles entities might play with respect to the agent's activity. For example, *the-bee-I-am-attacking*, as the name suggests, represents the bee Pengi is currently attacking. At different points in time, any of the bees in the game may 'become' this bee. In other words, Pengi only keeps track of the entities relevant to his current actions, relying on the fact that the individual identity of these entities does not matter. This deictic approach saves Pengi a lot of unnecessary computation and potential sources of error: as the world changes, only representations of parts of the world *currently* relevant to Pengi's actions are tracked and updated.

Deictic representation is informed by the realization that much of our interaction with the environment is guided by regularities in the world: e.g. we use cups to drink from, forks to eat with, etc. As such, the identities of cups, forks, etc. become irrelevant. Therefore, deictic representation does not designate unique entities in the world, but rather a role that a certain kind of entity can play. Different entities can play this role at different times, but the system will treat them the same way. As a consequence, the representation is not objective, in the sense that it is defined relative to the system's needs and interests rather than the identities of real-world entities. With deictic representation, a system only represents different entities to the extent that they fill roles that have behavioral consequences for the system.

Another example of an ad-hoc approach to representation was realized in Loyall and Bates's Hap agent architecture [31], which supports real-time, reactive behavior. A Hap agent is divided into a collection of behaviors, each of which integrates sensing and action in a similar manner to that suggested by Brooks, but which can also optionally query and update symbolic representations. The Hap architecture is designed to allow programmer-scripted behavior, similar to Pengi's control structure, to be flexibly interleaved with optional stretches of reasoned behavior which somewhat resemble plans. Like Pengi, the Hap agent architecture is explicitly designed to only require its programmers to specify, store, maintain and reason with symbolic representations to the extent that they are explicitly required by any individual behavior. Unlike Pengi, these representations can be stored and maintained over time as needed; so Hap agents can know where they are over the course of an interaction, or, in later architectures developed from Hap [41,35], can explicitly reflect on their own behavior. Unlike classical AI architectures, they do not need to do so consistently and coherently across their entire structure; they only need to do so to the extent specifically useful for the situation and behavior they find themselves in.

Consequences for Ubicomp

Alternative approaches to non-systematic representation support more complex functionality than non-representational approaches (like strategies 1 and 2), without introducing the complexities of complete, objective world models. For example, deictic representations could be used in a smart home to represent roles that specific household objects such as forks and chairs play in specific behavioral contexts, rather than attempting to represent the individual identities of objects, and the "total context" of the home. Even when using more traditional forms of symbolic representation, Hap demonstrates that designers can reduce the complexity of building real-world working systems by dropping the requirement that representations must be complete, coherent and consistent.

Strategy #4: Design for human-in-the-loop dynamics

With strategy 4, we start mining insights from second-wave interactionist AI. Where first-wave interactionists focused on robust behavior in complex environments, second-wave interactionists focus on human interpretation of and interaction with autonomous systems. A recurring theme of first-wave interactionist AI was to design systems to take advantage of structural regularities of the physical environments into which they were deployed. In a move reminiscent of Simon's ant [42], this bootstrapping on the dynamics of the world allowed the internal code of the system to be simplified while maintaining complexity of observable behavior. Second-wave interactionists built on this work by bootstrapping on the behavioral regularities of human beings.

One example of such work is Kismet, a sociable, humanoid robot, developed by Cynthia Breazeal and her collaborators at MIT [8]. Kismet engages people in face-to-face interaction and gives, reads and learns social cues such as gaze, eye movement, posture, gesture, etc. much as human babies do. Serendipitous interactions with Kismet highlight the role people's interactions play in bootstrapping system behavior. One such interaction was caused by limitations of the cameras that were part of Kismet's vision system, which required that the person interacting with the robot be located within a certain range from the cameras. If the person came too close, the robot would suddenly move back so that the distance would fall within the range again. This sudden movement was often interpreted as an invasion of Kismet's 'private space' and would cause the person to retreat as well. Similarly, if an object was too far, Kismet would crane his neck towards the object to get a better view. A human observer would read interest in Kismet's action and would usually bring the object closer.

Consequences for Ubicomp

These episodes show how complex interactions with responsive systems may develop not only from the internal code of the system but also from the dynamics that occur when that code interacts with users who interpret, respond to, and trigger new behavior in the system. Given that

ubicomp systems are always directed towards human users, they could capitalize on the fact that simple mechanisms, when coupled with user behavior, can achieve complex design goals.

For example, Kristina Höök's group has developed emotional sensing interfaces for mobile devices based on the notion of the affective loop [28]. The idea behind the affective loop is that the problem of sensing users' emotions can be reframed if the gestures the user must use to express an emotion tend to reinforce that emotion (e.g. vigorous shaking to express excitement). By having the sensed modality reinforce the expressed emotion, users are more likely to actually feel the emotion that they are expressing. Such systems can support explicit reflection on emotion and richer experiences of emotional communication – even in cases where the system itself has limited-to-no understanding of the emotions involved.

Strategy #5: Leverage socio-cultural knowledge

Historically, the approach of complete world modeling grew out of AI research performed in micro-worlds, such as the classic blocks world, in which it was possible to explicitly represent the entire world, given that the micro-world had been purposefully simplified and circumscribed just so as to make this possible. Attempts to apply micro-world approaches to real-world systems lead to the difficulties and interactionist responses described above. The strategies above are all approaches for eliminating or limiting the use of internal representations of the world. Another response is to engage in more complex and complete micro-world modeling, but to carefully align the internal micro-world maintained by the system with rich social and cultural knowledge possessed by human participants, in such a way that the system can participate in contexts much richer than it is literally representing and reasoning about. In other words, we can separate the use of complex, extensive representation from the idea that representations must exist in a one-to-one correspondence to the outside world. Instead, representations can participate in external, human meaning systems whose complexity is *not* fully captured internally by the system. This is the strategy of building micro-worlds with cultural significance.

First-wave interactionism served as a useful corrective within AI, and pragmatically, opened up new technical strategies for building robust, real-world systems. However, in its ideological formulation, the first-wave has sometimes argued the un-nuanced position that “symbols are bad, behavior is good.” It is perfectly possible, however, to use complex, formal representational methods from classical AI in richly interactive, real-world systems. As Mateas has argued, interactionism does not, of necessity, have to commit to specific technical approaches; the difference, rather, is in how those representations are incorporated into system design [34].

For example, Steffi Domike, Michael Mateas, and Paul Vanouse's *Terminal Time* [36] is an AI system that generates ideologically-biased historical documentaries of the last millennium in real-time based on audience feedback. The system asks questions of the audience to determine their apparent ideological bias, and then generates a half-hour historical documentary which reflects this bias. At first, the system's spin on history is gentle, creating a story which feels comfortable to an audience's preconceptions; but gradually the system ramps up the spin, making the audience's ideological orientation uncomfortably apparent.

Terminal Time uses complex, formal models to represent and reason about historical events, using planning algorithms to construct historical documentaries derived from Carbonell's classic work on ideologically-biased reasoning [15]. As such, it is subject to the critiques of planning and formal models promoted by researchers like Brooks – notably, that it imposes an a priori framework on the world rather than responding flexibly and intelligently to actual events. Yet these properties are precisely what make the system work, since the system is not intended to tell the full truth of the last millennium but present extreme views onto it. The design of *Terminal Time* avoids the trap of AI-completeness by explicitly acknowledging that the system is not intended to give one, true perspective on the world but instead an idiosyncratic, yet useful one. The full rhetorical import of the histories generated by *Terminal Time* are not completely represented in an internal model; rather, the internal model provides just enough structural scaffolding to usefully manipulate external symbols such as visual imagery, music, and generated narrative track, the full human meaning of which the system is unaware of. Thus, symbolically represented micro-worlds can serve as systems of authorship, a mechanism for structuring human experiences, where the richness of the experience lies in the heads of people, not the system.

Consequences for Ubicomp

A similar strategy can be applied in ubiquitous computing if we shift focus from building systems that behave in absolutely correct ways to constructing ones that explicitly maintain an alien, but nevertheless useful perspective; Mateas terms this strategy *alien presence* [39]. When using alien presence as a design strategy, *it is no longer necessary for a system to fully understand the domain of human activity which it is addressing in the way humans would*, since it is intended to be apparent to users that the system has its own forms of understanding. For example, Mateas and Böhlen's interactive sculpture Office Plant #1 [13] is a robotic office plant that changes shape based on its current reading of the social and emotional content of its users' email. Its output is enigmatic and ambiguous; its changing form suggests that the social and emotional tone of email matters, but is not a directly readable meter of social and emotional tone; because of this, the system does not need to

be able to understand this tone in a way which corresponds directly to human understanding).

Strategy #6: Design for an engaged audience

As examples such as Kismet and Pengi demonstrate, the behavior of the systems that interactionist AI researchers built often resonated much more deeply with human observers than a superficial reading of their code would suggest. Joseph Bates and his Oz group at CMU built on this understanding to suggest a further route around the problem of AI-completeness [5,6]. Specifically, Bates argued that classical AI tended to frame the problem of intelligence as creating a system that was so robust that even a hostile interlocutor (such as in the Turing Test) would be unable to distinguish its behavior from that of an intelligent being. Yet in Bates's area – interactive computer characters and interactive fiction – this notion of an inherently hostile user did not apply. Bates proposed that the problem of creating full-level human intelligence could be simplified by taking advantage of the fact that interactive computer characters could count on human audiences actively seeking to engage with and believe in them. As a consequence, the problem of intelligence for agents was less to be able to behave in an optimally intelligent way as to avoid making any serious mistakes that would break what Bates termed 'believability,' i.e. the willing suspension of disbelief by the audience.

As this work was extended [e.g. 30,38], 'believability' came to refer to the notion that the technical design of computer characters could be simplified by driving it largely *by the way in which agents were interpreted by friendly human audiences*, rather than by whether or not the agent would 'count' as intelligent if its behavior was analyzed either in a vacuum or by a hostile observer intent on identifying the limits of its functionality. In other words, system design could be simplified and made more effective by making the eventual experience and interpretation of the system by human users central to *every* technical decision made in the design of the system, rather than by considering this only when it came time to design the final user interface.

Consequences for Ubicomp

If we extend this idea to ubiquitous computing, it suggests technical advantages of simplification in shifting from a conception of ubicomp as proactive computing – i.e. setting the goal that systems should be able to engage in intelligent anticipation of human behavior, irrespective of human awareness or interaction – to designing systems which provide human-understandable reactions which can be further interpreted and built on by human users. One example of such simplification is in the use of sensing in the Home Health Horoscope system [23]. In this case, sensors track changing patterns of usage of objects that may be relevant for emotional reflection in the household. Informed by these patterns, the system's output is presented to the family in the form of daily horoscopes. The members

of the family use their rich background knowledge and understanding of the household to make sense of the horoscopes. While the system's relatively simple sensing will inevitably be unable to truly understand the full emotional tone of the household, the audience's interpretations fill in the gaps in the system's understanding of the situation. Because of this, there is no need to tackle the technical challenge of full, complete, and accurate sensing.

This idea extends to any setting in which the manner in which the system is experienced as performing is more important than the internal optimal functionality of the system – what we might in modern HCI parlance term experience design. Indeed, focusing on human-understandability of system behavior may help to ameliorate problems of user confidence in ubicomp systems; this has been identified as a major challenge for smart homes [21]. Although the idea of taking the human experience of the system as the point of functionality is not new in HCI, the use of believability as a reorientation of agent architecture can remind us of the fact that this focus on understandable user interpretation is not simply an interface requirement which should be largely the concern of product designers but a stance from which *every* level of technical work involved in building systems – including sensing, decision algorithms, machine learning techniques, etc. – could be rethought and simplified.

CONCLUSION

Our goal in this paper is not to propose interactionist AI as the magic solution to the problems of ubicomp. Indeed, the tactics of interactionist AI were often extreme, which limited the eventual efficacy of the approach. But the tactics were extreme for a reason: to provide a lens for rethinking the assumptions behind classical AI, and in particular to suggest that it was possible to make concrete technical progress without requiring general solutions, complex or consistent world representations, or full capabilities to reason about the world. This ability to spur reflection on how we might be able to do more with less are precisely what makes interactionist AI a valuable moment for reflection for ubicomp. In particular, interactionist AI arose in response to the realization that the problems AI was tackling presupposed a solution to the general problem of machine intelligence, i.e. what AI researchers termed the problem of AI-completeness. Interactionist AI researchers made the AI problem easier by avoiding tackling AI-completeness head-on, developing alternative strategies for AI systems that achieved their design goal without having to solve the human level intelligence problem.

When seen from this lens, we can recognize that much of ubicomp is implicitly tackling a similar problem to classical AI – trying to approach a human-level understanding of the world. This is particularly noticeable in the framing of proactive computing, or "proactive environments that anticipate our needs and act on our belief" [47] – such a

vision is difficult to achieve without systems that reliably and robustly can understand the world of human action. A similar implicit need for human-level understanding of the world underlies the framing of context-aware computing – as Dey argues, “[o]ne of the holy grails of context-aware computing is to have applications that do the right thing at the right time” [20, p. 7].

In solving this problem, ubicomp is using approaches that often mirror the methods of classical AI: breaking up the problem into functional modules, relying on formal representations and direct mappings between the outside world and world model, and seeking general solutions to the problems of sensing and action. This orientation is clearly reflected, for example, in Abowd and Mynatt’s influential paper laying out the ubiquitous computing research agenda: “Without good representations for context, applications developers are left to develop ad hoc and limited schemes for storing and manipulating [information about context]. The evolution of more sophisticated representations will enable a wider range of capabilities and a true separation of sensing context from the programmable reaction to that context” [2, p. 37]. While this framing has clearly been useful for the field, we need to be aware that we might be able to achieve the same goals without heading down the difficult path of achieving human-level understanding, a problem which we might in our field term *ubicomp-complete*.

Indeed, this problem of ubicomp-completeness resonates with recent critiques within ubicomp that question the ability and desirability to attain human-level understanding of the context of use. Looking at these issues through the lens of interactionist AI provides a useful starting point for turning these conceptual critiques into concrete technical proposals.

For example, both Dourish and Salvador & Anderson argue that the separation of context and action is problematic in ubicomp because the aspects of context that will be pertinent cannot be known until the action takes place [20,40]. The tight integration of sensing and action in interactionist AI agent architectures may support a more ad hoc, situated approach to tying context and action together than approaches which aim to separate perception of context as an individual and general functional unit.

As another example, Bell and Dourish argue that current ubicomp research is oriented around a conception of the world as orderly, homogeneous, and accessible to machine sensing and inference [7]. A major focus of ubicomp research has been on designing for such a world, in which complex representations and full world models make sense. However, Bell and Dourish argue, the world in which ubicomp systems are currently deployed is messy and heterogeneous and is likely to stay that way. If we take the messiness, heterogeneity, and unknowability of the world as a starting point for technical design, then the strategies of interactionist AI become particularly appropriate, since they

do not rely on the orderliness of the world and were designed with the explicit understanding that the world is inherently contingent. These strategies suggest shifting from the goal of designing complete, integrated ubicomp systems to building many small, ad-hoc devices that leverage regularities in the environment, and in people’s behaviors and perceptions, to participate usefully and engagingly in the human world.

ACKNOWLEDGMENTS

We thank Alex Taylor for stimulating discussions and insightful comments on earlier drafts of the paper. This work was supported in part by the following NSF grants: IIS-0752684, IIS-0238132 and IIS-0534445.

REFERENCES

1. Abowd, G. D. Software engineering issues for ubiquitous computing. In *Proc. Intl Conf. on Software Engineering*. IEEE Computer Society Press, 75-84.
2. Abowd, G. D. and Mynatt, E. D. Charting past, present, and future research in ubiquitous computing. *ACM Trans. Comput.-Hum. Interact.* 7, 1, 29-58.
3. Agre, P. E. *Computation and Human Experience*. Cambridge: Cambridge UP.
4. Agre, P. E. and Chapman, D. Pengi: an implementation of a theory of activity. In *Proc. AAAI’87*, 268-272.
5. Bates, J. The role of emotion in believable agents. *Commun. ACM* 37, 7 (Jul. 1994), 122-125.
6. Bates, J. Loyall, A. and Reilly, W. Integrating Reactivity, Goals, and Emotion in a Broad Agent. In *Proc. Conf. of the Cognitive Science Society*, 696-70.
7. Bell, G. and Dourish, P. Yesterday’s tomorrows: notes on ubiquitous computing’s dominant vision. *Personal Ubiquitous Comput.* 11, 2, 133-143.
8. Breazeal, C.L. *Designing Sociable Robots*. MIT Press.
9. Brooks, R. A. A Robust Layered Control System for a Mobile Robot. Technical Report. UMI Order Number: AIM-864., MIT.
10. Brooks, R. A. Elephants don’t play chess. *Robotics and Autonomous Systems* 6, 3-15.
11. Brooks, R. A.. Intelligence without representation. *Artif. Intell.* 47, 1-3 (Feb. 1991), 139-159.
12. Brown, B., Taylor, A, Izadi, S., Sellen, A., Kaye, J. and Eardley, R. Locating Family Values: A Field Trial of the Whereabouts Clock. In *Proc. UbiComp’07*, 354-371.
13. Böhlen, M. and Mateas, M. Office Plant #1. *Leonardo* 31 (5), 345-348.
14. Burke, J., Mendelowitz, E., Kim, J. and Lorenzo, R. Networking with knobs and knats? In *Ubiquitous Computing, Concepts and Models Workshop*, Gothenburg, Sweden, 2002.
15. Carbonell, J. *Subjective understanding*. Ph.D. Thesis, Yale Computer Science, Research Report #150.

16. Chapman, D. Planning for conjunctive goals. *Artificial Intelligence*. 32, 3 (Jul. 1987), 333-377.
17. Chatila, R. and Laumond, J. Position referencing and consistent world modeling for mobile robots. *Robotics and Automation*. 2 (Mar 1985), 138-145.
18. Dey, A. Understanding and Using Context. *Personal Ubiquitous Comput.* 5, 1 (Jan. 2001), 4-7.
19. Dey, A. K., Ljungstrand, P., and Schmidt, A. Distributed and disappearing user interfaces in ubiquitous computing. In *Proc. CHI '01*, 487-488.
20. Dourish, P. What we talk about when we talk about context. *Personal Ubiquitous Comput.* 8, 1 (Feb. 2004), 19-30.
21. Edwards, W. K. and Grinter, R. At Home with Ubiquitous Computing: Seven Challenges. In *Proc. UbiComp '01*, 256-272.
22. Ford, K. M. and Hayes, P.J. *Reasoning Agents in a Dynamic World: The Frame Problem*. Greenwich, CT: JAI.
23. Gaver, W., Sengers, P., Kerridge, T., Kaye, J., and Bowers, J. Enhancing ubiquitous computing with user interpretation: field testing the home health horoscope. In *Proc. CHI '07*, 537-546.
24. Grudin, J. Turing maturing: the separation of artificial intelligence and human-computer interaction. *interactions* 13, 5 (Sep. 2006), 54-57.
25. Harle, R. K. and Hopper, A. Towards autonomous updating of world models in location-aware spaces. *Personal Ubiquitous Comput.* 12, 4, 317-330.
26. Hightower, J. and Borriello, G. Location systems for ubiquitous computing. *Computer*, 34, 8, 57-66.
27. Hightower, J., Brumitt, B. and Borriello, G. The location stack: a layered model for location in ubiquitous computing. *Mobile Computing Systems and Applications*, 22-28.
28. Höök, K., Ståhl, A., Sundström, P., and Laaksolahti, J. Interactional Empowerment. In *Proc. CHI '08*, 647-656.
29. Kray, C. and Baus, J. AIMS 2005: artificial intelligence in mobile systems. In *Proc. MobileHCI'05*, vol. 111, 353-354.
30. Loyall, B. A. *Believable Agents: Building Interactive Personalities*. PhD thesis, CMU.
31. Loyall, B. A. and Bates, J. Real-time control of animated broad agents. In *Proc. of Conf. of the Cog. Sci. Society*.
32. Maes, P. How to do the Right Thing. *AI Memo 1180*, MIT.
33. Mallery, J. C. Thinking About Foreign Policy: Finding an Appropriate Role for Artificially Intelligent Computers. *The 1988 Annual Meeting of the International Studies Association.*, St. Louis, MO.
34. Mateas, M. Expressive AI: A hybrid art and science practice, in *Leonardo*, 34 (2), 147-153.
35. Mateas, M. and Stern, A. A Behavior Language: Joint Action and Behavioral Idioms. In H. Prendinger and M. Ishizuka (Eds), *Life-like Characters*, Springer.
36. Mateas, M., Vanouse, P., and Domike, S. Generation of Ideologically-Biased Historical Documentaries. In *Proc. AAAI'00*, 236-242.
37. Nilsson, N. J. Shakey the robot. Tech. Note 323, SRI AI Center.
38. Reilly, S. N. *Believable Social and Emotional Agents*. PhD thesis, CMU.
39. Romero, M., Pousman, Z., and Mateas, M. Tableau machine: an alien presence in the home. In *Extended abstracts of CHI '06*, 1265-1270.
40. Salvador, T. and Anderson, K. Practical Considerations of Context for Context Based Systems. In *Proc. UbiComp '03*, 243-255.
41. Sengers, P. Designing Comprehensible Agents. In *Proc. IJCAI-99*.
42. Simon, H. A. *The Sciences of the Artificial*. MIT Press, 1996.
43. Starner, T. The challenges of wearable computing: Part 2. *Micro*, vol.21, no.4, pp.54-67, Jul/Aug 2001.
44. Suchman, L. *Plans and situated actions*. Cambridge : Cambridge University Press, 1987.
45. Taylor, A., Harper, R., Swan, L., Izadi, S., Sellen, A. and Perry, M. Homes that make us smart. *Personal and Ubiquitous Computing* 11(5), 383-393.
46. Vere, S. and Bickmore, T. A basic agent. *Comput. Intell.* 6, 1 (Jan. 1990), 41-60.
47. Want, R., Pering, T. and Tennenhouse, D. Comparing autonomic and proactive computing, *IBM Systems Journal*, 42(1), 129-135.
48. Winograd T. & F. Flores. *Understanding Computers and Cognition*. Ablex Publishing Corp, 1985.
49. Weiser, M. The Computer for the 21st Century. *Scientific American*, 265 (3), 94-104.
50. Weiser, M. Some computer science issues in ubiquitous computing. *SIGMOBILE Mob. Comput. Commun. Rev.* 3, 3 (Jul. 1999), 12.
51. Woodruff, A., Augustin, S. and Foucault, B. Sabbath day home automation: "it's like mixing technology and religion". In *Proc. CHI'07*, 527-536.
52. Yeh, R., Liao, C., Klemmer, S., Guimbretière, F., Lee, B., Kakaradov, B., Stamberger, J., and Paepcke, A. 2006. ButterflyNet. In *Proc. CHI '06*, 571-580.