Evaluation of Contextual Models

Kurt Partridge, James "Bo" Begole, Victoria Bellotti

Abstract—**As context-aware systems have grown in sophistication, they have adopted more complex contextual models. This paper steps back from the details of model contents and composition and investigates how personal and group models can be evaluated. A specific method is proposed that leverages the transition in the user's cognitive state between "ready-at-hand" and "present-at-hand."**

Index Terms— **Context-aware systems, context, user modeling, model evaluation, ubiquitous computing**

I. Introduction

Context-aware systems have ballooned in sophistication over the past decade. The earliest applications did little more than associate data with a particular location [13, 12]. Today, systems represent much more abstract concepts, reason about these concepts using rigorous mathematical logics, and span a variety of domains from inferring transportation mode [8], to controlling home heating, ventilation, and air conditioning (HVAC) [5], to detecting interruptibility [2].

The more advanced an application is, the more detailed a model it requires. Early applications performed no modeling. They simply read sensor data and acted on it. New applications combine sensor data to build models of individuals or of groups of people.

But building accurate models of users and groups is not easy. Users and groups are too complex to model completely, so any system must choose an appropriate simplification. And finding such a simplification is today more art than science.

In this paper, we argue that for models to improve, there must be a clear way to measure them. We propose a metric that evaluates a model according to how often an application incorporating it makes a mistake that interferes with the user's task.

II. Contextual Models

Before considering how to evaluate contextual models, it is important to discuss what contextual models are. We describe three types of contextual models: environmental contextual models, personal contextual models, and group contextual models.

A. Environmental Contextual Models

Environmental contextual models describe data directly captured by sensors, such as accelerometers, microphones, compasses, GPS, and others. Simple processing of the data streams converts the data from a low-level description of physical state (e.g., electrical resistance in a thermistor) into human-level information (e.g., degrees Celsius).

Although many applications can use these low-level descriptions directly, context-aware systems typically build richer models by incorporating semantic assumptions or aggregating data from multiple sensors. For example, a cellphone that hears a particular GSM tower signal can combine this information with assumptions about the locations of the beacons to determine the position of the cellphone. Using multiple beacons and their signal strength measurements makes much finer-grain location resolution possible.

B. Personal Contextual Models

Personal contextual models represent characteristics of individuals, such as physiological conditions, emotional states, and tasks and activities. Some parts of personal context models are built the same way environmental models are. For example, a person's heart rate might be measured by using a heart rate monitor. A personal model containing heart rate would only need to combine this sensed data with the assumption that the monitor was attached to the correct person.

More sophisticated personal contextual models propose to model information that is subjective and hidden inside users' heads. These models are also built by combining environmental models with semantic assumptions, but the assumptions are less reliable, so the conclusions are less reliable. As Tolmie, et al. [11] have pointed out, environmental context provides a clue about individuals' inner states, but does not provide a guarantee.

For example, to do its job properly, a smart thermostat would need to know whether a user regarded the ambient temperature as too cold or too warm. Using cameras, the thermostat might infer that a user was cold because he held his arms tightly or wore a coat indoors. But several other situations could also cause these conditions. And the absence of these conditions would not indicate that the user was not cold.

As a second example, location can be used in context-aware systems to infer broad categories of mental states [9,10]. The inference is performed indirectly. First, sensed environmental data is converted to global coordinates. Then, the coordinates are converted to a categorical label, such as "home," "work," "street," or "store." Then, the label is associated with various activities that the user might perform at each kind of place. Combined with other sensed data, the user's current and future activities may be partially inferred, but not with complete accuracy.

Systems that infer mental states must treat this unreliable information carefully. For example, a system that automatically takes action based on occasionally inaccurate information would often irritate users and be considered

unusable. But a different system could use the same information to anticipate the user’s actions and to simplify user interface. It might prompt the user appropriately or suggest the most likely possible steps that a user might take. In cases where the information is inaccurate, the user could ignore the suggestions. Section III describes how the unreliability of information affects not just the design of systems, but also their evaluation.

C. Group Contextual Models

Contextual models can be extended beyond environmental models and individual models to include groups. Group contextual models affect personal contextual models by representing information about the group as a whole (such as tasks, activities, and members), but, more interestingly, by providing additional semantic interpretations of contextual data.

Cultural norms serve as a source of these semantic interpretations. A cultural norm establishes the meaning of behaviors for any member of the group. Many context-aware systems assume that sensed events indicate particular behaviors, which are then interpreted in terms of semantic meanings derived from cultural norms. For example, Lilsys [2] detects whether an office door is shut, and uses this information along with other inputs to model a person’s interruptibility.

As with personal context models, group context models may arrive at incorrect conclusions because the world is much more complicated than the model. In the experiment with the office door, we found that some people closed their doors primarily to block out noise but were willing to be interrupted even though their door was closed. However, they were aware of the secondary effect that a shut door would be interpreted as a signal of uninterruptability. To be effective, a context-aware system that models interruptibility would need to be aware of both possible reasons, and accurately model the reason behind the user’s actions.

A second source of semantic interpretations is a role within a group. Each role defines a different semantic interpretation of events. Roles may change over time. In a study of family member availability, Nagel, et al. [7] found that availability varied not only along expected lines of social engagement, but also along these dynamically changing roles, such as which parent was on duty. A context-aware system with some knowledge about roles and their meanings would work better than one without such knowledge.

Group contexts are difficult to use because groups are dynamically formed, and the norms and roles are often negotiated using natural language. Interruptibility, for example, is negotiated using shared context [1]. The user who normally uses a shut door to indicate unavailability might inform a colleague to disregard the shut door and interrupt her when necessary. This negotiation implicitly forms a new group composed of two people. Further contextual information might become associated with this new group even as new members are added to it. Work has begun on detecting the formation and modification of groups through analyzing communication patterns [4], but in general this is a difficult problem because it requires a deep understanding of natural language.

III. Evaluation of Contextual Models

For a context-aware system to be effective, it must be accurate. That is, the model must reflect as closely as possible the real-world situation it is modeling. If a model is inaccurate, the system will take inappropriate actions or present incorrect information.

To improve the accuracy of a context model, system evaluators must be able to reliably measure how well a model performs. Poorly-performing models can be adjusted and improved through repeated measurements and adjustments. Measurements are made by comparing the model with another model that is either the modeled entity itself or a trusted model of the modeled entity. Measurements can compare either a model’s structure or a model’s performance. Both have their strengths and weaknesses. Structural comparisons are more comprehensive because they show all possible ways that a model could respond to its inputs. Yet they are limited because the structure of both the model and what it is compared with may not be available. In some cases an evaluator may compare a structural model with the evaluator’s mental model of the entity, but this approach may miss some crucial differences. Performance comparisons are better at finding inconsistencies because they directly compare a model’s specific values against the entity or trusted model. However, they are limited to the performance traces studies. In the software testing field, these two approaches are referred to as whitebox testing and blackbox testing.

Environmental models are usually easy to evaluate. Temperature, noise level, light level, position, and other environmental characteristics are easily validated by comparing them with equivalent sensors. But user and group models are much more complicated. The entities being models are human thoughts, which are not easily quantified. Thoughts are dynamic, and subject to interpretation from the person’s previous experience.

One way to evaluate models of people and groups is to directly ask the user whether a model is correct. Direct questioning is very flexible. Provided that the user understands the question, direct questioning can cover any aspect of context. Techniques for direct questioning—such as questionnaires and interviews—are also well-known and well-studied.

However, direct questioning has disadvantages. If users’ statements are taken at face value, then the system evaluation may not be accurate. What people express may not accurately reflect what they really think. In some cases, a user may not be consciously aware of the aspect of context that is being asked about. Or a user might have internally conflicting thoughts, and only express one. Or the act of even asking the question may change the user’s state enough to make it difficult for them to accurately answer. Such is the case with interruptibility; interrupting the user to ask whether they are interruptible seems slightly paradoxical.

One way around some of these problems is postponed questioning. By asking the user whether a contextual model matches their past experience, the user's task flow is not interrupted, so errors of this sort are minimized. Sometimes, users may not have good memories of their past situation, but video or audio recordings can improve recall. Presenting aggregated visualizations, such as rhythm model diagrams [3] can also aid user recollection, and sometimes point out trends that the user was not aware of. However postponed questioning does not address issues of conflicting thoughts, and still requires users to consciously mediate their experience.

Another useful approach is ethnography. Ethnography is not as flexible as direct questioning, because it is limited to what the ethnographer observes. And although ethnographers themselves are biased, a trained ethnographer will notice facts related to the contextual model that the user may not. But ethnography does not scale well and is expensive because it requires a trained observer.

Here we are proposing a new mechanism in which a system may itself evaluate the effectiveness of its own contextual model while it is used. The idea is to design a system to take advantage of transitions between the two cognitive states of "ready-at-hand" and "present-at-hand" [14]. These states refer to situations when a person is using a tool to accomplish some task. Ready-at-hand means that the user is thinking not about the tool, but the task they are performing. The tool is "invisible" because the user's conscious thoughts are not directed toward it. Present-at-hand refers to a situation where the user is consciously aware of a tool because they are learning to use it or because the tool is broken. When a system incorporating a contextual model forces more transitions than an equivalent system with a different model, then the first system's contextual model is likely to be less accurate than the second's.

The transition can be detected in a variety of ways. In some applications, users may switch from the task they are working on to examine the application's settings. Or the application might be structured in such a way that it is clear when the user must correct an incorrect action taken by the application. Or, affective computing techniques leveraging galvanic skin response or vision might be applied to detect a rise in the user's stress level. Or, an unusual pause may be sufficient to identify a transition.

This metric is not perfect. If the user is engaged in an activity unrelated to the context that the system acts on, then the user might continue with the current activity if the action is inconsequential. For example, a system that incorrectly shut off the lights might not cause a transition in the mind of a user who was focused on Web browsing, because the darkened room lighting does not block the Web browsing activity. Even if the action is incorrect and related to the user's activity, it may not be important if the user is not particularly goal directed. For example, a user browsing for information might not care if the context-aware system makes suboptimal suggestions. And finally, a user primarily interested in experiencing fun might enjoy exploring a system's behavior based on mistaken contextual inferences.

This approach is a performance-based evaluation approach, so it is limited to the application and the situations that the application exercises. However, this may not be a significant weakness. Because the entities being modeled are very complex, models could cover a very wide range of situations, and it is helpful to limit the evaluation to the situations that an application actually involves. But an evaluator must be careful to exercise unusual but important situations if they do not arise naturally.

An advantage of this approach is that it is passive. That is, it can be used without burdening the user. Although one could argue that because users have already moved away from the ready-at-hand state that they are in a position to be able to consciously express to the system the failure of a contextual model, the more work that they have to do to express the failure the further from the ready-at-hand state they move. Ultimately, users care about their task and not the tool, so the system that provides the least burden is best.

Evaluation of group contextual models could especially benefit from a transition-based evaluation. Because users are themselves often not consciously aware of social dynamics (which in some cases require trained ethnographers to discover), it seems less likely that users would be able to identify flaws from visualizations of group contexts, and more likely that they would run into them while using such systems.

IV. Conclusion

This paper has explored the issues surrounding context-aware systems, emphasizing the problems of evaluation. We have described three levels of contextual models: environmental, personal and group. As described by Dourish [6] we recognize that when applications make use of context, they must combine sensed information with semantic assumptions. The most useful applications act as though they have intelligence, which means that they must incorporate facts normally only available in people's mental models.

Additional contextual features of cultural norms and roles become important when multiple people interact in groups. Because of these cognitive and social issues, evaluation of contextual models is challenging. We have proposed a mechanism that evaluates models according to how well they allow users to remain in the ready-at-hand state. This mechanism may be applied in different ways, depending on the sophistication of the user. With time, this mechanism and other techniques for evaluating personal and group context will help contextual models become more useful and more accurate than they are today.

References

[1] P. M. Aoki and A. Woodruff, "User Interfaces and the Social Negotiation of Availability," *Workshop on Forecasting Presence and Availability, CHI 04,*Vienna, Austria, Apr 2004.

[2] J. B. Begole, N. E. Matsakis, and J. C. Tang, "Lilsys: Sensing unavailability," *CSCW '04: Proceedings of the 2004 ACM conference on Computer Supported Cooperative Work,* pp. 511–514.

[3] J. B. Begole, J. C. Tang, and R. Hill, “Rhythm Modeling, Visualizations and Applications,” *UIST ‘03: Proceedings of the 16th annual ACM symposium on User Interface Software and Technology*, pp. 11–20.
[4] T. Choudhury and A. Pentland, “Sensing and Modeling Human Networks using the Sociometer,” *International Conference on Wearable Computing*, White Plains, NY, October 2003.
[5] D. J. Cook, M. Youngblood, E. Heierman, K. Gopalratnam, S. Rao, A. Litvin, and F. Khawaja, “MavHome: An Agent-Based Smart Home,” *Proceedings of the IEEE International Conference on Pervasive Computing and Communications*, 2003, pp. 521–524.
[6] P. Dourish, “What We Talk about When We Talk about Context,” *Personal and Ubiquitous Computing*, 2004.
[7] K. S. Nagel, J. M. Hudson, and G. D. Abowd, “Predictors of Availability in Home Life Context-Mediated Communication,” *CSCW ‘04: Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*, 2004, pp. 497–506.
[8] D. J. Patterson, L. Liao, K. Gajos, M. Collier, N. Livic, K. Olson, S. Wang, D. Fox, and H. Kautz, “Opportunity Knocks: a System to Provide Cognitive Assistance with Transportation Services.” *Proceedings of UBICOMP 2004: The Sixth International Conference on Ubiquitous Computing*, volume LNCS 3205, pp. 433–450.
[9] B. Schilit, N. Adams, and R. Want, “Context-Aware Computing Applications,” *IEEE Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, US, 1994.
[10] J. C. Tang, N. Yankelovich, J. Begole, M. Van Kleek, F. Li, and J. Bhalodia. “Connexus to Awarenex: Extending Awareness to Mobile Users,” *CHI ‘01: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2001, pp. 221–228.
[11] P. Tolmie, J. Pycock, T. Diggins, A. MacLean, and A. Karsenty, “Unremarkable Computing,” *CHI ‘02: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2002, pp. 399–406.
[12] G. M. Voelker and B. N. Bershad, “Mobisaic, An Information System for a Mobile Wireless Computing Environment,” *IEEE Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, US, 1994.
[13] R.Want, B. N. Schilit, N. I. Adams, R. Gold, K. Petersen, D. Goldberg, J. R. Ellis, and M. Weiser, “An Overview of the PARCTAB Ubiquitous Computing Experiment,” *IEEE Personal Communications*, 2(6):28–33, Dec 1995.
[14] T. Winograd and F. Flores, editors. *Understanding Computers and Cognition*, Ablex Publishing Corp., Norwood, NJ, USA, 1985.

Dr. Kurt Partridge is a member of research staff in the Ubiquitous Computing Area. His research interests are in simplifying the user experience in ubiquitous computing environments, by combining mobile, wearable, and infrastructure technologies. He received his Ph.D. in Computer Science and Engineering from the University of Washington in August 2005. His dissertation investigates technologies for automatically customizing and coordinating electronic devices without explicit user intervention. He holds a Masters Degree in Computer Science and Engineering from the University of Washington, and a Bachelors Degree from the University of California, Berkeley.

Dr. James “Bo” Begole is the manager of the Ubiquitous Computing area in the Computing Science Laboratory of the Palo Alto Research Center. He holds a B.S. in Mathematics from Virginia Commonwealth University, and M.S. and Ph.D. degrees in Computer Science from Virginia Tech. Prior to his studies, he served in the U.S. Army as an Arabic language interpreter (MOS 98G), stationed in the US, Panama, Egypt, and Greece. Dr. Begole was technical lead and Principal Investigator of several collaborative computing projects at Sun Microsystems Laboratories, investigating real-time collaboration and awareness technologies including the development of temporally predictive models of online presence. Recently he investigated the use of sensors in the workplace to determine when a person is less available for interruption.

Dr. Victoria Bellotti is a Principal Scientist and Manager of the Socio-Technical and Interaction Research area in the Computer Science Lab at PARC. She studies current and prospective technology users to understand their work-practice, their problems and their requirements for future technology. She also designs novel systems, having a number of pending patent applications, and works on analyzing existing or proposed technology design for utility and usability and on finding ways to improve it. She is active in the HCI community, having chaired, edited and reviewed numerous events and publications in the field. Her current research interests include Task and Information Management, Computer Supported Cooperative Work, Computer Mediated Communication and Ubiquitous Computing. Dr. Bellotti has a B.S. in psychology, an M.S. in ergonomics and a Ph.D. in Human-Computer Interaction, from London University UK.