IEEE Instrumentation and Measurement
Technology Conference
Anchorage, AK, USA, 21-23 May 2002

Sensor Fusion for Context Understanding

Huadong Wu[1*], Mel Siegel[2(contact author)], and Sevim Ablay[3]
[1,2]Robotics Institute, Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213, phone: (412)-268-8742;
[3]Applications Research Lab, Motorola Labs
1301 E. Algonquin Road, IL02/2230, Schaumburg, IL 60196, phone: (847)-576-6179
E-mail: [1]whd@cmu.edu, [2]mws@cmu.edu, [3]Sevim.Ablay@motorola.com

ABSTRACT

To answer the challenge of context-understanding for HCI, we propose and test experimentally a top-down sensor fusion approach. We seek to systematize the sensing process in two steps: first, decompose relevant context information in such a way that it can be described in a model of discrete facts and quantitative measurements; second, we build a generalizable sensor fusion architecture to deal with highly distributed sensors in a dynamic configuration to collect, fuse and populate our context information model. This paper describes our information model, system architecture, and preliminary experimental results.

Keywords: intelligent sensors, networking, sensor fusion, context-aware

1. INTRODUCTION

Humans can convey very complicated ideas easily because they have an implicit understanding of their environmental situation, or “context”. “Context-aware computing” research aims at intelligently connecting computers, users, and the environment; its sensing technology research — the research of intelligently integrating multiple sensors and multiple sensor modalities, or sensor fusion — however is not entirely up to the challenge yet.

A user’s situational context may include various levels of relevant information about his/her activities and intention as well as the current environment (with higher-level contexts being derived from multiple lower-level contexts and sensors’ outputs). A context-aware computing system can sense such a wide range of information possibly only via a network of sensors working in concert. However, different sensors have different resolutions and accuracies as well as data rates and formats, and the sensed information may have overlaps or even conflicts. To characterize “context” information hence means that some sensor fusion technologies are indispensable.

Classical sensor fusion technology has difficulty fulfilling context-sensing requirements because: (1) the computation is in a mobile environment and available sensors are often highly distributed, resulting in a highly dynamic configuration; (2) for the system to be commercially successful, the employed sensors cannot be expensive; (3) as the goal of context-sensing is to facilitate human-computer-interaction, its measurement resolution and accuracy requirements are usually commensurate to that of human perception capabilities, which are often too difficult for inexpensive sensors to achieve; (4) the sensed context information is more meaningful to humans in a semantic description than in numerical parameters.

The motivation of our research is to push sensor fusion towards context-sensing or context-understanding: to construct a contextual information model and build a generalize-able sensor fusion software architecture that can support mapping sensors’ raw output data into the contextual information hierarchy.

2. CONTEXT MODEL

Sensor fusion for context-aware HCI (Human-Computer-Interaction) faces a two-fold challenge: (1) how properly to represent context in a computer understandable way, and (2) how to map sensors’ output into the context representation. This section deals with the first question.

2.1. Context Classification

As described in the last section, context can be anything from low-level parameters, such as time and temperature, to highly abstract concepts such as intention and social relationship. To completely describe or represent our colorful real world in a computer-understandable way seems an insurmountable task; so we start by classifying a few common but interesting contexts and try to deal with only a subset of the knowledge about a user’s environment.

[*] Huadong Wu is the recipient of a Motorola Partnerships in Research Grant.

Using a human user-centered approach, the context information can be roughly expanded along three dimensions:

- **Space/physical**: the user's outside environment, the user's own activity in reacting to the environment, and the user's internal physical and mental status;
- **Time/history**: current time-of-day that is conventionally assumed appropriate for some activities, personal and related group's activity schedule, personal activity history and preferences;
- **Spiritual/social relationship**: regarding people the user will likely care about.

Our literature search has uncovered very few efforts to study contextual information classification and modeling thus far. Schmidt et al [4] proposed to structure context into two general categories, human factors and physical environment, with three subcategories each. Orthogonal to these categories, history provides an additional dimension of context information. Dey et al [3] suggested categorizing the contexts into a two-tier system. In their two-tiered system, the primary tie has the four primary pieces of information that characterize the situation of a particular entity: location, identity, time, and activity. All the other types of contexts are considered as second level – they can be indexed by the primary contexts because they are attributes of the entity with primary context.

Either of these prior methods is difficult to use because of its complexity and fuzziness in definition.

Instead, we propose a pragmatic user-centered approach to simplify the context information into three categories: (1) environment, (2) the user's own activity, and (3) the user's own physiological states. Each category has its own subcategories and their contents may overlap. The key point is that this classification can provide us with guidance and references in building contextual information architecture model.

As an example, Table 1 lists simple common environmental contexts. With this framework, it is easy to decide which context information should be included in system design based on objective analysis of effectiveness versus implementation costs.

2.2. Context Representation

Once it has been decided what context information should actually be included in our context-aware system, we can design our context information model to represent the context. We will not deal with very abstract concepts at our first stage; instead, we assume that the commonly used context information can be simplified into a collection of discrete facts and events. Though not excluding numeric parameters, our context model primarily uses discrete fact and event descriptors in an unambiguous and efficient representation.

This means that the context should be decomposed to the extent that it can be represented in a format of numerical values, string descriptions or indices. The point is that we build such a context model to describe a very simplified world, yet the information is useful for our targeted applications.

Table 1. Commonly desired environmental contextual information

Location	City, altitude, weather (cloudiness, rain/snow, temperature, humidity, barometer pressure, forecast), location and orientation (absolute GPS reading, relative to some objects)
	Change of location: traveling, speed, heading
Proximity	Close to: building (name, structure, facilities, etc. knowledge), room, car, devices (function, states, etc.), vicinity temperature, humidity, vibration, oxygen richness, smell
	Change of proximity: walking/running, speed, heading
Time	Day/date: meaning of day/time (office hour, lunch time, etc.), season of year, etc.
	History experience, schedule, expectation, etc.
People	Individuals or group (e.g. audience of a show, attendees in a cock-tail party); people activity and interaction (casual chatting, formal meeting, eye contact, attention arousing etc.); non face-to-face interaction
	Social relationship and interruption source: incoming calls, encountering, etc.
Audiovisual	Human talking (information collection), music, etc.; in-sight objects, surrounding scenery etc.
	Noise-level, brightness of environment
Computing & connectivity	Computing environment (processing, memory, I/O, etc., hardware/software resource & cost), network connectivity, communication bandwidth and cost etc.

Context information is preferably modeled hierarchically, the same way human beings deal with a large amount of information. For computer systems to "understand" — store, associate, update, and retrieve information — a relational database is preferable. Therefore, the realization of our context information model is a set of tables in a relational database; each table describes an aspect of context information. For system scalability, dynamic context information will have its own table with its number of records changing to reflect the dynamic content.

All the sensed context information will have a pair of numbers to indicate this sensed message's confidence interval; and for the items with dynamic content, there is a time stamp to indicate when this information was updated.

2.3. *A Context-Modeling Case Study*

The context-aware HCI requires a model of the real world, existing as a relational database in digital domain. However, “databases do not model the real world, although it is a common misconception that they do … rather, databases are models of users’ perspective of the world.”[9] Because there is no strict rule regarding whether a design is correct or not, design of a relationship model is to some extent like an artist’s impression rather than a scientist’s analysis. We will give a simple example to illustrate our approach to designing context information architecture.

Our application scenario is a small group of users who frequently use a small conference room. The users’ basic information and preferences are pre-registered. Properties of the conference room, such as its location, function, facilities, usage-policy etc. are also predefined. Reflecting our user-centered and application-oriented design philosophy, we use the Entity-Relationship Model [9] approach to model context information, where the “user” and “conference room” are the two key entities. These entities have a dynamic relationship, i.e., once a registered user is detected in the room, the relationship of presence is updated as shown in Figure 1.

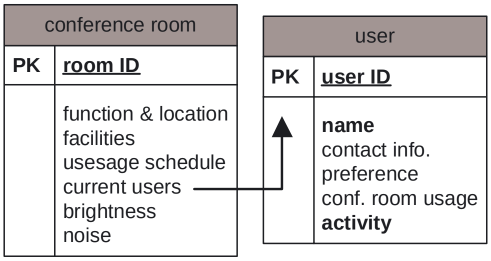

Figure 1. Entity Relationship between conference room and users

The context information should be easily scalable so that we can add more detailed or higher-level context information whenever we decide that is necessary or cost-effective. For example, in addition to, say, active-badge or fingerprint reader to detect user presence in the conference room, we can set an omni directional camera at the center of the conference table, and use multiple microphones, to detect meeting activity. This adds context information as shown in Figure 2:

In our preliminary implementation[1], we use recorded data from the previous Stiefelhagen et al experiments for user’s “focus of attention” analysis research via monitoring meeting participants’ head pose and speaking activities to simulate sensors’ behavior and to update our dynamic context information database.

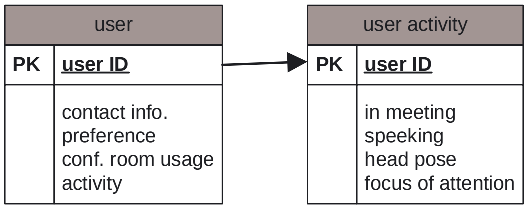

Figure 2. Context information architecture is scaled up to include user's meeting activity context

Our thesis is that the context information architecture can be pre-defined (though it is effected by practical implementation and cost-effectiveness considerations), and the usage of context information to some extent can be separated from the concerns of how the contexts are actually acquired. From applications’ perspective, the context information would be more desirably presented following our context classification with some practical adjustment so that the frequently used information does not go down deep into sub-branches in the information hierarchy tree.

3. SENSOR FUSION IMPLEMENTATION

With a hierarchical context information architecture iteratively designed as in Section 2.3, the next step is to design a sensor fusion system to implement the mapping mechanism from raw sensory data into this context database. To simplify context sensing and context usage, it is desirable that a context-aware system have a modular and layered architecture. Such a system ought be built so as to insulate applications from the context extraction. Correspondingly, the sensor-implementation should be transparent to context extraction and interpretation.

3.1. *System Architecture*

There are many system architectures proposed and implemented to support context-aware computing. From a context acquisition and consumption point of view, they can be roughly classified as “context component architectures” and “context blackboard architectures” [1]. A context component architecture views the world as a set of context components corresponding to real world objects such as

[1] Refer to our other IMTC’2002 paper (IM-1076) “Sensor Fusion Using Dempster-Shafer Theory” for further details.

users, facilities, and devices; these components interact with each other as the agents of real world objects. A context blackboard architecture treats the world as a blackboard, where different types of context can be filled in and taken off.

There are some practical tradeoffs between the two categories of system architecture design. Our system design is based upon the George Institute of Technology's Context Toolkit system [2], enhanced by us with "sensor fusion mediator" modules that manage the uncertainty of sensors' outputs. It is basically a context component architecture augmented by some context blackboard architecture features to manage sensor fusion. Our system architecture diagram is shown in Figure 3.

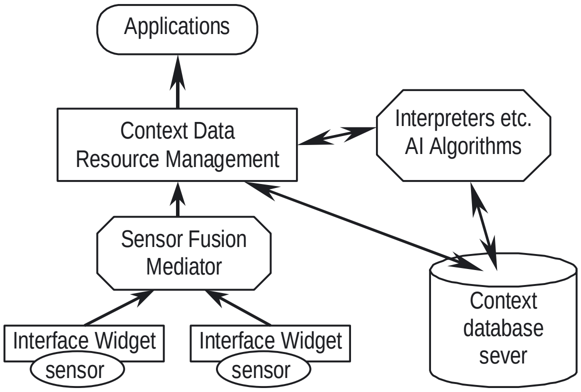

Figure 3. System architecture for sensor fusion of context-aware computing

In this system, each sensor's output includes both measurement data and measurement confidence. For each kind of context information, there is a sensor fusion mediator to collect and process the measurement confidence.

3.2. Implemetation and Experiments

The sensor fusion is implemented in the process of combining like kinds of context observation information: overlaps and cross-verifying information will increase the estimation confidence, whereas conflicting signals will decrease the associated estimation confidence.

The sensor fusion mediator modules also function as system updating coordinators. Each sensor fusion mediator manages the information about all sensors that generate a corresponding kind of context information, the normal time interval needed to update the context from the list of sensors' observation, and an updating flag to indicate whether the system is too busy to update its context information. Each newly available sensor will report its availability to the relevant sensor fusion mediator, and if possible will announce its imminent unavailability.

Any sensor in the list that first reports an observation may trigger the information updating process, but in general, the sensor fusion mediator will decide when and how to update the state by appointing some sensors to act as the active triggering source. An updating process begins with the sensor fusion mediator raising the update flag; it then starts to query all available sources in the sensor list. If the updating process detects a sensor configuration change, it updates the sensor list and adjusts the estimated time interval needed to update context information next time.

Given that the context information architecture has been predefined, and microphone and camera sensor have been set up as described in Section 2.3, the context sensing system automates system interactions between sensor Widgets, sensor fusion mediators and context as illustrated in Figure 4.

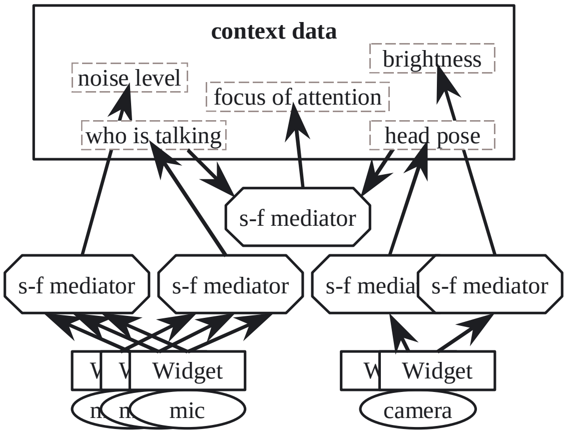

Figure 4. Interactions between sensors, fusion mediators, and context

We have finished the baseline research: programming and testing sensor fusion engine with Dempster-Shafer theory of evidence method as our key sensor fusion algorithm. Our first application case study relates to extracting higher-level context (i.e. analysis of users' focus of attention in small group meeting) from lower-level context information sources, where we use pre-recorded and pre-analyzed audiovisual "who is talking" and "the user's current head pose" data sets.

We are integrating the sensor fusion engine into our context-sensing system architecture to simulate sensors' real-time behavior and interactions between system components.

4. CONCLUSION AND FUTURE RESEARCH

Our experiments today are preliminary but promising. We anticipate more concrete results by conference presentation time. To achieve the goal of flexibility and scalability, besides following the universal guideline of "modular and layered architecture", we design the sensor fusion system with a crisp interface between the sensor fusion mediator module and the actual sensor fusion realization algorithm module. This enables us to flexibly design and test different sensor fusion algorithms with different sensor sets.

From system architecture analyses and preliminary experiments, we conclude that we can expect performance improvement over currently existing context-sensing systems in the following areas:

- **Simplification of sensor fusion**: in most cases the sensor fusion process becomes a task of recalculating confidence. For the case of Dempster-Shafer theory implementation, at various levels of abstraction, it allows us to freely combine all sensors' observations with any confidence level, — even the information such as ignorance about a proposition.
- **Availability of information**: context information usage is further separated from context acquisition. On the one hand it makes applications more resilient to the influence of system hardware configuration change and the system easier to scale up; on the other hand, the centralized context information aggregation makes it easier for artificial intelligent algorithms to access the context data to derive more abstract higher-level context information.

Our sensor fusion research aims at HCI context-sensing requirements in context-aware computing applications, where sensors are highly distributed and their configuration is highly dynamic. Based on the assumption that more complex context can be decomposed into simpler discrete facts and events, our proposed top-down systematic approach can provide a clear path towards letting computers understand context in human-like ways. The next step in our research agenda is to integrate additional low-level HCI sensing systems such as camera-based user-tracking, microphone-based speaker-recognition, and tactile-based fingerprint readers. We plan to test these in real situations, e.g., of conference-room and other user tracking applications. Our long-term goal is to make this sensor fusion platform flexible and scalable enough to be widely adopted in various context-aware applications.

5. REFERENCES

[1] Huadong Wu, thesis proposal: "Supporting Sensor Fusion for Context-aware Computing", July 2001, http://www.cs.cmu.edu/~whd/ContextAware/Proposal.pdf

[2] Anind K. Dey, "Providing Architecture Support for Building Context-Aware Applications", PhD thesis, November 2000, Georgia Institute of Technology, http://www.cc.gatech.edu/fce/ctk/pubs/dey-thesis.pdf.

[3] Anind K. Dey, and Gregory D. Abowd, "Towards a Better Understanding of Context and Context-Awareness", GVU Technical Report GIT-GVU-99-22. June 1999, ftp://ftp.cc.gatech.edu/pub/gvu/tr/1999/99-22.pdf

[4] Albrecht Schmidt, Michael Beigl, and Hans-W. Gellersen, "There is more to Context than Location", Proceedings of the International Workshop on Interactive Applications of Mobile Computing (IMC98), November 1998, Rostock, Germany, http://www.teco.uni-karlsruhe.de/~albrecht/publication/imc98.ps.

[5] Jason Pascoe, "Adding Generic Contextual Capabilities to Wearable Computers", in the Second International Symposium on Wearable Computers, pages 92-99, Pittsburgh, October 1998. IEEE Computer Society. http://www.cs.ukc.ac.uk/pubs/1998/676/index.html.

[6] Lawrence A. Klein, "Sensor and Data Fusion Concepts and Applications" (second edition), SPIE Optical Engineering Press, 1999, ISBN 0-8194-3231-8

[7] Schilit, B.N., Adams, N.L., Want, R., "ContextAware Computing Applications", Proceedings of the Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, December 1994. IEEE Computer Society. http://citeseer.nj.nec.com/schilit94contextaware.html

[8] Steve Shafer, "Ten Dimensions of Ubiquitous Computing", Keynote presentation at Conference on Managing Interactions in Smart Environments, December 1999. http://www.research.microsoft.com/easyliving/Documents/1999%2012%20Ten%20Dimensions.doc

[9] David M. Kroenke, "Database Processing: Fundamentals, Design & Implementation" 8th edition, Prentice Hall, 2002