ContextMap: Modeling Scenes of the Real World for Context-Aware Computing

Yang Li, Jason I. Hong, James A. Landay
Group for User Interface Research, Computer Science Division
University of California, Berkeley
Berkeley, CA 94720-1776 USA
{yangli, jasonh, landay}@cs.berkeley.edu

ABSTRACT

We present a scenegraph-based schema, the ContextMap, to model context information. Locations with hierarchical relations are the skeleton of the ContextMap where nodes of people, objects and activities can be attached. Context information can be collected by traversing the ContextMap. The ContextMap provides a uniform method to represent physical and social semantics for context-aware computing. In addition, context ambiguity can be modeled as well.

Keywords

Context-aware computing, scenegraph, context ambiguity

INTRODUCTION

Context is the glue to link the real world with the virtual world. Context is "any information that can be used to characterize a situation" [4]. We call the situation a scene of the real world. The information can be the temperature of a region. It also can be the activity of a person, e.g., reading a book, or the activity of a group, e.g., having a meeting.

Both the physical and the social semantics of a situation are required by context-aware computing. Social semantics are embodied through physical activities, and physical activities can be fully understood only under certain social circumstances. For example, we can see "running" as a status of a person at a physical level. It can mean "catching a bus" at a social level. Activity theory [1] sees an activity as functionally subordinated hierarchical levels, i.e., activities, actions, and operations. Each action performed by a human being has not only intentional aspects but also operational aspects. This reveals how social activities can be performed through physical actions and objects.

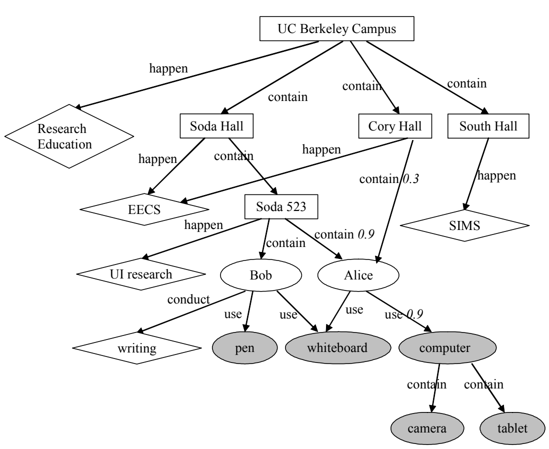

Figure 1: An example ContextMap. Rectangles indicate *Place* nodes. Diamonds stand for *Activity* nodes. *People* nodes are represented as ellipses and *Object* nodes are ellipses in gray.

Context information itself is recursively related. For example, linguistically, the context of a word is the sentence, which in turn gets its context from the paragraph. The Berkeley campus has the climate context of the City of Berkeley, which inherits it from the San Francisco Bay Area of California based on location containment.

To leverage the abundant interaction semantics of context, it is necessary to have an efficient way to model the context. We devised the ContextMap (see Figure 1) to model the situation of the real world for context-aware computing as a scenegraph-like structure. The ContextMap provides a consistent way to model context information and addresses the correlation and ambiguity of context data.

RELATED WORK

The Active Map [5] provides a basic organization of context that consists of a hierarchy of locations with a containment relation. We employed the location hierarchy as the skeleton of the ContextMap, but we include relations in addition to location containment.

Crowley et. al. [2] described context as a network of situations concerning a set of roles and relations. Roles may be "played" by one or more entities. Dey formulated three kinds of entities for context-aware computing: people, places and things (or objects) [4]. We model these roles and entities as nodes and edges of a ContextMap.

The scenegraph [6] has been widely used in computer graphics. Its dynamic propagation of graphical attributes greatly simplifies the representation of a scene and it proves an efficient way to model complicated scenes. To model scenes of the real world, we extended the scenegraph to deal with the context semantics of the real world.

INTRINSIC AND RELATIONAL CONTEXT ATTRIBUTES

The context information of an entity can be classified into intrinsic and relational attributes. Intrinsic attributes of an entity can be described without referring to others, e.g., the identity of a person can be his name. A person's status can be his age or health condition. However, relational attributes of an entity can only be specified by its relations with other entities. For example, the position of an entity can usually be described as a relative spatial relation with other entities, e.g., near or far and in or out.

NODES AND EDGES OF A CONTEXTMAP

Like a traditional scenegraph, a ContextMap is a directional acyclic graph (see Figure 1) and the context attributes are collected by a depth-first traversal. An entity, i.e., a place, a person or an object, is represented as a node of the graph. Each node maintains the intrinsic attributes of an entity that it represents. Relational attributes of an entity are represented by edges directly or indirectly linked to its node. So the context of an entity is represented not only by the attributes in its node but also by the node's position in the entire ContextMap. A ContextMap is a view of the real world that can be shared by multiple applications.

Another kind of node in a ContextMap is the *Activity* node, which represents the social semantics of an entity or a group of entities, e.g., reading a book or having a seminar. It can be applied to a sub-graph of a ContextMap like the dynamic propagation of graphical attributes in a scenegraph. It means that the activity is conducted by people with certain tools (physical objects) at a certain location. For example, in Figure 1, "UI Research" happens in Soda 523 and it indirectly indicates the activity of Bob, Alice and the tools they are using to achieve this activity.

Place nodes stand for entities that are places or sites. They can refer to a large region ("California") or a small area ("close to whiteboard"). The containment relation between *Place* nodes is stable and hierarchically structured, e.g., the UC Berkeley campus contains Soda Hall and will always do so. *Place* nodes and their containment relations constitute the skeleton of a ContextMap, which can be enriched by nodes describing people, physical objects, and activities. A ContextMap can be built by establishing a static *Place* hierarchy first. Directional edges from *Place* nodes can indicate *contain* relations for physical containment and *happen* relations for locations where some events, i.e., social activities or roles, happen. For example, "education & research" happens on the "UC Berkeley Campus". An *Object* node is for a physical object, e.g., a pen, which can have directional *contain* edges to its sub-components. *Contain* relations are transitive.

A *Person* node represents a person entity. Directional edges from a *Person* node can indicate *conduct* or *use* relations, specifying the person is conducting an action or using a physical object (tool), respectively. A *use* relation can transfer the semantics of a *contain* relation. For example, the fact that "Bob" is in "Soda 523" and he is using the "pen" indicates that the "pen" is also in "Soda 523".

A node can be referenced by multiple nodes. For example in Figure 1, both "Bob" and "Alice" are using the "whiteboard". The multi-reference to a node can also be used to model context ambiguity. For example, "Alice" could be either in "Soda Hall" or "Cory Hall" in Figure 1.

Intrinsic attributes of a node can be tagged by a timestamp to indicate when they are updated or a time span to indicate their validity. Moreover, a directional edge can be tagged to indicate the valid period of a relation.

MODELING CONTEXT AMBIGUITY

In reality, both sensed and interpreted context is often ambiguous [3]. The ContextMap models context ambiguity by tagging edges and the intrinsic attributes of nodes with confidence values. For example, the intrinsic attribute "health condition" of "Alice" could be 0.8. In Figure 1, the confidence of "Alice" in "Soda 523" is 0.9 and in "Cory Hall" it is 0.3. Edges without labelled values have the default confidence value "1.0".

Here we describe a simple method to calculate the confidence of transitive relations.

Given $x \xrightarrow{\alpha} y$ and $y \xrightarrow{\beta} z$, $x \xrightarrow{\alpha\beta} z$.

For example, the confidence of "Alice" using "computer" is 0.9. Since the confidence of Alice in Soda 523 is also .9, the confidence of "computer" in "Soda 523" is 0.81.

However, the confidence of "whiteboard" in "Soda 523" is the average of the confidences of all paths from "Soda 523" to "whiteboard". It is 0.95 based on [Soda 523, Bob, whiteboard] = 1 and [Soda 523, Alice, whiteboard] = 0.9.

CONCLUSION AND FUTURE WORK

ContextMap enables an efficient representation of complicated situations, particularly for relational context, by using dynamic attribute propagations and transitive relations. Both social and physical semantics of context can be represented in a consistent manner. Attributes and relations of nodes can be updated based on sensed information, e.g., a person's location and its confidence, or manually, e.g., an *Activity* node can be manually added in or manipulated beforehand or in runtime. ContextMaps will be provided as an infrastructure service to applications. We are continuing to refine the representation and evolution mechanisms of the ContextMap, and to enable easy construction of and access to ContextMaps.

REFERENCES

1. Bertelsen, O.W. and Bodker, S. Activity Theory. HCI Models, Theories, and Frameworks Ed. by Carroll, J.M. Morgan Kaufmann Publishers. 2003, pp. 291-324.
2. Crowley, J.L., Coutaz, J., Rey, G. and Reignier, P. Perceptual Components for Context Aware Computing, Proceedings of UBICOMP2002, Sweden.
3. Dey, A.K., Mankoff, J., Abowd, G.D. and Carter, S. Distributed mediation of ambiguous context in aware environments, Proceedings of UIST 2002, pp. 121-130.
4. Dey, A.K., Salber, D. and Abowd, G.D. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications, Human-Computer Interaction, 2001, 16(2-4), pp. 97-166.
5. Schilit, B. and Theimer, M. Disseminating Active Map Information to Mobile Hosts, IEEE Network, Vol. 8, pp. 22-32, 1994.
6. Strauss, P.S. and Carey, R. An Object-Oriented 3D Graphics Toolkit, ACM Computer Graphics, 1992, 26(2).