Representing Context in Hypermedia Data Models

Frank Allan Hansen
Department of Computer Science
Centre for Interactive Spaces
University of Aarhus, Denmark
Åbogade 34, DK-8200 Århus N
fah@interactivespaces.net

ABSTRACT

As computers and software systems move beyond the desktop and into the physical environments we live and work in, the systems are required to adapt to these environments and the activities taking place within them. Making applications context-aware and representing context information along side application data can be a challenging task. This paper describes how digital context traditionally has been represented in hypermedia data models and how this representation can scale to also represent physical context. The HyCon framework and data model, designed for the development of mobile context-aware hypermedia systems, are presented as concrete examples of a model combining application data, hypermedia structures and context information in one model.

Keywords

Context, Context-awareness, Context Modeling, Hypermedia, HyCon

1. INTRODUCTION

The visions of Pervasive Computing [19, 20] present the idea of ubiquitous computing power available through numerous mobile and stationary computer devices distributed (some more or less invisible) throughout our environment and accessible through a ubiquitous network. As these new technologies change the way we work with computers and information there is an increasing need to aid users in finding and producing relevant, situated information.

Supporting users in these activities has been the ambition for hypertext and hypermedia researchers for a long time. Indeed, one may argue that the motivating reason for creating hypermedia systems has always been to contextualize information and to support users in accessing and understanding information in different contexts.

Over the years many hypermedia systems have been developed which support both substantive and annotative activities but almost all of these systems have only been concerned with links between pieces of information in the computer (i.e., in the digital context) and have not supported links to or between objects in the physical world. Furthermore, information is "put into" a context when linked to i.e., a context defined by the author, but few systems supports the context of the reader (where is the reader? What is he/she doing? Why is the information interesting for him/her and so forth). Thus, the physical use context has typically been ignored and considered something outside the systems. However, as hypermedia (and computer systems in general) moves beyond the desktop and into the physical environments we live and work in, adding context-aware capabilities to the systems is necessary to make them truly useful components in pervasive computing environments.

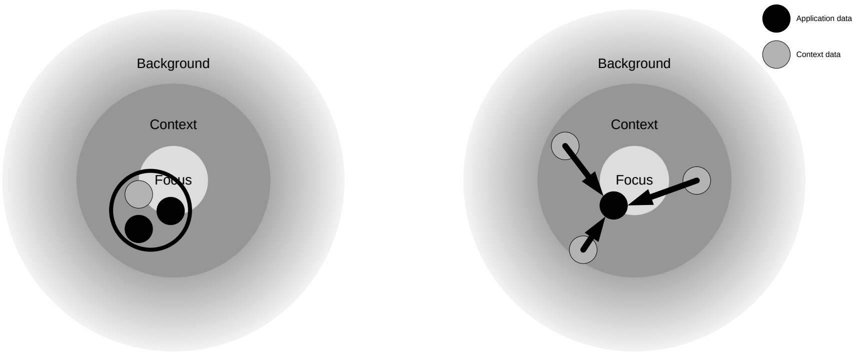
This paper discusses how sensed context data can be utilized by hypermedia applications and focuses especially on how Open Hypermedia data models can be extended to represent objects (both digital and physical), their context, and hypermedia structures on top of them. The paper is structured as follows: section 2 introduces a general definition of context and section 3 discusses how context can be represented in open hypermedia models. Section 4 introduces the HyCon framework and data model and section 5 briefly describes the experiences with the data model thus far.

2. DEFINING CONTEXT

Most people have an understanding of the meaning of the term *context*, but the usage of the word is often vague since the word has different meanings depending on, well, the context. For example, in computer science we talk about analyzing and designing for specific use contexts, about multitasking context switch, context-free languages and grammars, contextual links, context-sensitive help etc.

The idea of utilising context and creating context-aware systems that "adapts according to the location of use, the collection of nearby people, hosts, and accessible devices, as well as to changes to such things over time" was first introduced by Schilit*et al.* [15] and Schilit and Theimer [16]. Context was here described as "location, nearby people and objects, as well as changes to those objects over time". Schilit*et al.* stated that there were three important aspects of context: "where you are, who you are with, and what resources are nearby". Dey [9] argues that these definitions are too specific and hard to apply when designing context-aware systems; enumerating which context aspects are important is not possible since they change from situation to situation. Instead Dey proposes a more operational context definition [9, p. 5]:

International Workshop on Context in Mobile HCI, September 19, 2005, Salzburg, Austria.



(a) (b)

Figure 1: Two views on context: (a) Objects exist *in* a context: context can be modeled as a composite or collection of objects. (b) Objects *have* a context: context can be modeled as properties of the object.

> "*Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.*"

Using this definition, information can be categorized as context on an application level, as e.g., done by Schilit*et al.* in [15]. Even though this general definition puts it into the hand of the application developers to determine what information qualifies as context for a specific application, Dey [10] provide some help in listing four *essential* categories of context information: identity, location, status (activity), and time. These categories are denoted as essential because they can be used to derive other kinds of context. The identity of a person can e.g., be used to look up the persons contact information, location can be used to find other colocated entities and the four categories may be combined to deduce more complex patterns—if several people are in the same room, at the same time with a meeting agenda it may be deduced that the meeting is actually taking place. Similar operational definition of context are given by Schmidt *et al.* [17] and Chen and Cotz [5].

Applying such operational definitions of context in a general way e.g., to data models in framework implementations, requires flexible and extensible designs since they must support the modeling of "*any* information that can be used to characterise the situation of an entity…". For context-aware hypermedia frameworks a number of requirements follow directly from the general context definitions: The framework should support a data model combining both application data objects and hypermedia structures (i.e., link and composite structures, annotation objects, etc), and at the same time be extensible enough to support different types of context specified by application programmers. Having determined what constitute context for a specific application, the framework should also support developers in implementing behaviors and features to take advantage of the context information in their applications.

3. CONTEXT IN HYPERMEDIA MODELS

Hypermedia systems have typically modeled the concept of context in one of two ways: either as a collection or composite serving as a partitioning mechanism on the (global) hypertext or as attributes of key-value pairs, associated with links, nodes and other hypermedia objects, describing parameters of the context the hypermedia objects belong to. Context modeled as collections is often a purely structural partitioning concept constraining browsing and linking to some kind of context given by the user explicitly or implicitly. Key-value pairs associated with objects are typically used to describe in which context the objects are visible. When the object structures are retrieved only those structures whose context attributes match that of the user's context or the given query are returned; structures that do not are pruned away. Neptune's *Contexts* [7], Intermedia's *Webs* [21], and the Webvise *Context* composites (based on the the Open Hypermedia Systems Working Group[1] (OHSWG) navigational hypermedia model OHP-NAV [11]) belong to the first category while FOHM's *context objects* [13] and Storyspace's *Guards* [2] belong to the second.

Taking these two views as the starting point gives rise to different understandings of the context model as illustrated in Figure 1:

Context as collections: Context can be modeled as collections of first class objects in the model. Objects being part of the same collection are considered to belong to the same physical context. This model corresponds to a *belongs to* relationship where the objects exist *in* a context as illustrated in Figure 1(a).

Context as object attributes: Context can also be modeled by assigning attributes, describing context parameters, to data objects. This allows context to be modeled as simple key-value(s) pairs and corresponds to objects that *have a* context relationship.

Using key-value pairs has also been used to model context in a number of other context-aware systems e.g., Schilit's and Theimer's *located objects* [16] (physical entities with a location) which were modeled as tuples of key-value attributes and the context-aware Web browser Mobisaic [18] used environment variables to store and update context information which was used in dynamically generated URLs.

[1] `http://www.cs.aue.auc.dk/ohswg/`

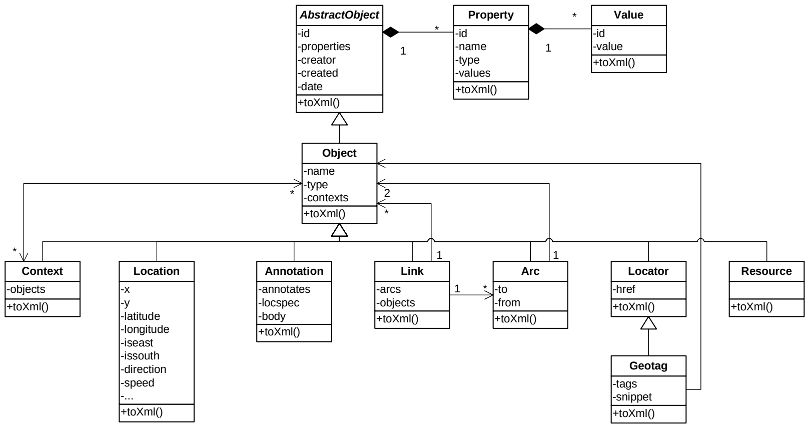


Figure 2: The HyCon hypermedia model.

4. THE HYCON FRAMEWORK

The *Hyper*per*Con*text platform (HyCon) is developed in the Centre for Interactive Spaces at the University of Aarhus to provide a general platform suitable for experiments with hypermedia mechanisms in a context-aware and mobile environment [4]. HyCon includes both an architecture and infrastructure for implementing context-aware services and applications and a framework which can be used by applications programmers to build such services and applications. The HyCon framework was designed to encompass annotations, links, and guided tours associating locations and RFID- or Bluetooth-tagged objects with maps, Web pages, and collections of resources. The HyCon architecture extends upon earlier location based hypermedia systems [6, 1, 14] by supporting authoring in the field and by providing access to browsing and searching information through a novel geo-based search (GBS) interface for the Web [12].

4.1 The HyCon Datamodel

The data model implemented in the HyCon framework describes the hypermedia structures and context objects used by services and applications built on the framework. The data model is illustrated in Figure 2. All objects in the model are described as subclasses of the abstract `AbstractObject` class. Instances of its subclasses all share common attributes: they all have globally unique identifiers (GUIDs), meta-data describing their creator and modification timestamps, and a set of property-value pairs.

Instances of the `Context` class are referential composites, which can hold other objects belonging to the same physical or digital context. Objects from the `Location` class represent physical locations and are used as the primary mechanism for identifying physical locations in the model. The `Annotation` class represent the data model's support for annotations. Annotations can include multimedia content such as text, images, sound, and video. Annotations can be associated with any type of object derived from the `Object` class. Hence, the model supports the general notion of annotations annotating other annotations. Links and link trails are modeled by the `Link`, `Arc`, and `Locator` classes describing XLink [8] based structures on top of the other objects in the system. The Locator objects are used to represent digital resources reachable by a URI in the model with Arc objects specifying the traversal behaviour of the link. The data model has been generalised to support linking instances of the generic `Object` class and thus support easy linking of `Locator`-type objects as well as classes derived from the `Object` class. The `GeoTag` class is used to tag elements directly with URIs (e.g., to tag a location with a descriptive Web page).

The HyCon data model supports both ways of modeling context as described above. Context can be modeled as first class objects in the model e.g., links or annotation structures can be associated with a `Location` object through a `Context` composite object to designate that the links are available on that physical location. While this technique gives the programmer the advantages of an object model, all services and storage structures have to be updated to handle the different context types, when new classes of context is added to the model.

The other way to model context is by using the `Property-Value` objects associated with the `AbstractObject` class. This allows context to be modeled as simple key-value(s) pairs and corresponds to objects that *have a* context relationship. In contrast to the object model this gives a type-less context model (i.e., the key-value pairs are sets of strings), but since all services and the storage structures already have support for handling these structures it is much easier to experiment with this model or create applications which support a more ad-hoc notion of context e.g., for prototyping purposes.

5. EVALUATION

In terms of completeness i.e., whether it is possible to use the HyCon data model in various applications to model context and application data, we have been quite satisfied. A number of different applications have been built atop the HyCon framework including the mobile, context-aware hypermedia system HyConExplorer designed for tablet PCs and Java enabled smart phones [3]. The HyConExplorer is designed to let users working in the field access and produce situated information related to their context. Furthermore, smart board applications for online user tracking and mobile location based games have been implemented with the framework and data model.

We found that for most applications, representing context as key-value(s) properties on the generic objects was the most convenient way to represent context information. The four essential context types (identity, location, status, and time) can easily be modeled by this mechanism and adding ad-hoc information is relatively easy for the application programmer since it requires no change to server side services.

6. CONCLUSION

Applying the general, operational definitions of context and context-awareness described herein to data models in pervasive software architectures, requires flexible and extensible designs since they must support the modeling of arbitrary context information.

This paper describes two ways of modeling context in data models for hypermedia applications and frameworks: the first is to model context as collections of objects corresponding to a *belongs to* relationship where the objects exists *in* a context and the second uses simple key-value(s) pairs corresponding to objects that *have a* context relationship.

The HyCon framework and data model is presented as a concrete example of these two ways of representing context.

7. REFERENCES

[1] G. D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton. Cyberguide: a mobile context-aware tour guide. *Wireless Networks*, 3(5):421–433, 1997.

[2] M. Bernstein. Storyspace 1. In K. M. Anderson, S. Moulthrop, and J. Blustein, editors, *HYPERTEXT '02: Proceedings of the thirteenth ACM conference on Hypertext and hypermedia*, pages 172–181, College Park, Maryland, USA, June 2002. ACM Press.

[3] N. O. Bouvin, C. Brodersen, F. A. Hansen, O. S. Iversen, and P. Nørregaard. Tools of contextualization: Extending the class room to the field. In *Proceedings of the 4^{th} International Conference for Interaction Design and Children*, Boulder, Colorado, USA, June 2005.

[4] N. O. Bouvin, B. G. Christensen, K. Grønbæk, and F. A. Hansen. HyCon: A framework for context-aware mobile hypermedia. *The New Review of Hypermedia and Multimedia*, 9:59–88, 2003.

[5] G. Chen and D. Kotz. A survey of context-aware mobile computing research. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, November 2000.

[6] K. Cheverst, K. Mitchell, and N. Davies. The role of adaptive hypermedia in a context-aware tourist GUIDE. *Communications of the ACM*, 45(5):47–51, 2002.

[7] N. M. Delisle and M. D. Schwartz. Contexts-a partitioning concept for hypertext. *ACM Transactions on Information Systems*, 5(2):168–186, 1987.

[8] S. DeRose, E. Maler, D. Orchard, and B. Trafford (editors). XML Linking Language (XLink). W3C Recommendation 27 June 2001, W3C, June 2001. `http://www.w3.org/TR/xlink/`.

[9] A. K. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5(1):4–7, 2001.

[10] A. K. Dey, D. Salber, and G. D. Abowd. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction (HCI) Journal*, 16(2-4):97–166, 2001.

[11] K. Grønbæk, L. Sloth, and N. O. Bouvin. Open hypermedia as user controlled meta data for the Web. In *Proceedings of the* 9^{th} *International World Wide Web Conference*, pages 553–566, Amsterdam, Holland, May 2000. W3C.

[12] F. A. Hansen, N. O. Bouvin, B. G. Christensen, K. Grønbæk, T. B. Pedersen, and J. Gagach. Integrating the Web and the World: Contextual trails on the move. In D. de Roure and H. Ashman, editors, *Proceedings of the* 15^{th} *ACM Hypertext Conference*, Santa Cruz, CA, USA, Aug. 2004. ACM Press.

[13] D. E. Millard, D. T. Michaelides, D. D. Roure, and M. J. Weal. Beyond the traditional domains of hypermedia. In D. Millard, J. Haake, and S. Reich, editors, *Proceedings of the International Workshop on Open Hypermedia Systems Core Concepts and Research Directions, Pre-Conference Workshop at the ACM* 13^{th} *International Conference on Hypertext and hypermedia (HT'02)*, pages 26–32, University of Maryland, College Park, MD 20742, USA, June 2002. Informatik Berichte, FernUniversität Hagen, Hagen.

[14] J. Pascoe. The stick-e note architecture: extending the interface beyond the user. In *Proceedings of the* 2^{nd} *international conference on Intelligent user interfaces*, pages 261–264. ACM Press, 1997.

[15] B. N. Schilit, N. I. Adams, and R. Want. Context-aware computing applications. In *Proceedings of the Workshop on Mobile Computing Systems and Applications*, pages 85–90, Santa Cruz, CA, USA, Dec. 1994. IEEE Computer Society.

[16] B. N. Schilit and M. Theimer. Disseminating active map information to mobile hosts. *IEEE Network*, 8:22–32, 1994.

[17] A. Schmidt, K. A. Aidoo, A. Takaluoma, U. Tuomela, K. V. Laerhoven, and W. V. de Velde. Advanced interaction in context. In *HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, pages 89–101, London, UK, 1999. Springer-Verlag.

[18] G. M. Voelker and B. N. Bershad. Mobisaic, an information system for a mobile wireless computing environment & engineering. In *Proceedings of the Workshop on Mobile Computing Systems and Applications*, pages 85–90, Santa Cruz, CA, USA, Dec. 1994. IEEE Computer Society.

[19] M. Weiser. The computer for the 21^{st} century. *Scientific American*, 265(3):66–75, Feb. 1991.

[20] M. Weiser. Some computer science issues in ubiquitous computing. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3):12, 1999.

[21] N. Yankelovich, B. J. Haan, N. K. Meyrowitz, and S. M. Drucker. Intermedia: the concept and the construction of a seamless information environment. *Computer*, pages 81–96, Jan. 1988.