New Review of Hypermedia and Multimedia
Vol. 9, 2003, pp. 59–88

HyCon: a framework for context-aware mobile hypermedia

NIELS OLOF BOUVIN, BENT G. CHRISTENSEN,
KAJ GRØNBÆK & FRANK ALLAN HANSEN

Department of Computer Science, University of Aarhus, Åbogade 34, DK8200 Århus N, Denmark
e-mail: {n.o.bouvin, bentor, kgronbak, fah}@daimi.au.dk

This paper introduces the notion of context-aware mobile hypermedia. Context awareness means to take the users' context such as location, time, objective, community relations, etc., into account when browsing, searching, annotating, and linking. Attributes constituting the context of the user may be sensed automatically and/or be provided by the user directly. When mobile, the user may obtain context-aware hypermedia support on a variety of small and medium sized computing platforms such as mobile phones, PDAs, tablet PCs, and laptops. This paper introduces the HyCon (HyperContext) framework with an architecture for context-aware hypermedia. The architecture includes interfaces for a sensor tier encapsulating relevant sensors and represents the hypermedia objects in structures based on the XLink and RDF standards. A prototype called the HyConExplorer created with the framework is presented, and it is illustrated how the classical hypermedia features such as browsing, searching, annotating, linking, and collaboration are supported in context-aware hypermedia. Among the features of the HyConExplorer are real-time location-based searches via Google collecting hits within a specified nimbus around the user's GPS position. Finally, the use of scenarios for and evaluation of the use of the HyConExplorer in public school projects are discussed.

Keywords: Context aware computing; Context aware hypermedia; Geo-spatial hypermedia; Open hypermedia; XLink; SVG

1. Introduction

The idea of utilising location in computing systems was advocated by Mark Weiser, in his legendary paper on ubiquitous computing (Weiser 1991). The notion of context awareness was introduced by Schilit and Theimer (1994) and Schilit *et al.* (1994) at a conference on mobile systems. Since then, a large amount of research within the ubicomp communities has addressed issues of developing context-aware applications

ISSN 1361-4568 (print)/ISSN 1740-7842 (online) © 2003 Taylor & Francis Ltd
DOI: 10.1080/13614560410001725310

and physical environments. Despite the high level of attention, there are still many different definitions and notions of context being used in the literature. Dey (2001) has introduced a concise definition of context:

> 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.

and of context awareness:

> A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task.

According to Dey, there are three types of context-aware application support, namely presentation of information and services, automatic execution of services, and tagging of context to information to support later retrieval.

We see both needs and prospects in applying such a notion of context awareness to hypermedia, and we will argue that this is a new research area that requires more attention now as hypermedia moves beyond desktop computers and traditional Web browsers and into smaller mobile devices. We will briefly discuss previous hypermedia research that has touched upon issues regarding context.

1.1. Context and hypermedia

Relating Dey's definitions to hypermedia, we may think of adaptive hypermedia (De Bra *et al.* 1999; Brusilovsky and Rizzo 2002) applications aiming at adapting hypermedia navigation to the user based on some recorded profile of the user or their context. But the main focus of adaptive hypermedia has so far been on modelling the user (Beaumont 1994) rather than analysing the context in which the user is situated when browsing. Thus, we see a need for taking the user's real-time context into account when working with hypermedia.

In the hypermedia literature, the term context has played another role when discussing navigation context with the hypermedia structures per se. Delisle and Schwartz (1987) introduced the notion of context in their HAM architecture to constrain the substructure in which hypermedia navigation and linking were undertaken. This means that context in this case was considered a purely structural partitioning concept, and the only thing it has in common with the notion of context awareness in this paper is the fact that browsing and linking are constrained to some kind of context given by the user explicitly or implicitly. The FOHM hypermedia model (Millard *et al.* 20002002) supports the notion of context as a filtering mechanism on the hypermedia structures. FOHM allows first-class hypermedia objects to be augmented with context objects. These are sets of key/value pairs that 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.

Another notion of context in hypermedia is represented by the works of Hardman *et al.* (1993), where the notion of context is the surroundings of the link marker in a hypermedia document, and the issues are what to preserve in the user's view when following the link out of the context. Finally, El-Beltagy *et al.* (2001) introduced a notion of context, where the user's profile and a current Web page in focus are used to infer and generate new multi-headed links between the current page and other contextually related Web pages. This notion of context is also quite different from what we are introducing in this paper, since it is only the users' more or less static profile and the current Web page which are considered to be the context.

1.2. Context-aware hypermedia

By context-aware hypermedia we take Dey's definitions as the starting point, but when talking about hypermedia, we see the need to go beyond just providing information and services depending on the user's tasks. We see a need to let the users create annotations, links, collections, guided tours, and the like, automatically tagged with the captured context information. Thus, we focus on creating more than just a context-aware browsing and navigation system, namely full context-aware hypermedia systems, where users can become producers and provide hypermedia structured materials to co-workers, school mates, or friends.

Based on Dey's definitions, we have in our design discussions divided the concept of context into three basic perspectives: *physical context*, *digital context*, and *conceptual context*. The physical context is the physical surroundings of an entity; object, system or person. The physical context includes physical location, physical objects, physical interaction, and absolute time and space. Computer systems may be aware of the physical context by capturing information about it. The digital context includes computer models, infrastructure, protocols, available (digital accessible) devices and services, recorded history, and relative time and space. The conceptual context describes a user's activity, intention, focus, and understanding of the surroundings. Most entities may have both a physical and a digital representation, i.e. the physical phenomenon is modelled in the computer system. The relationship between the physical entity and the digital model is the user's understanding (the conceptual model or user model). The representation of physical entities in the digital model should thus reflect the conceptual context.

Since physical objects are characterised by being positioned in absolute time and space, we can store and retrieve information about them by capturing or computing related context information. We use this as the basis for browsing, searching, and producing hypermedia structures in the user's context. This is described in Sections 3–6. However, in the digital context, time and space are relative. Temporal events may be rearranged, and virtual physical context may be explored by changing parameters in the digital model (e.g. the user may explore structures related to a location which is both temporally and physically remote from their current location). We use this to support browsing, searching, and manipulation of hypermedia structures while being outside the physical context to which these structures are related. However, the digital context may also represent relationships not found in any physical context, e.g. the result of searching for annotations created by a certain user may be annotations associated with physical objects belonging to different points in time and space. Thus, the digital context may include relationships computed from, or on top of, the representations of associations in the digital and physical context(s).

1.3. Use scenarios and cases

Some projects working with context-aware systems are based on use scenenarios from indoor working environments like the PARCTab (Want *et al.* 1995); others focus on outdoor tourist-information systems (Abowd *et al.* 1997). We have found our inspiration and use scenarios in school environments focusing on project oriented learning activities outside the class room (Nørregaard *et al.* 2003). School projects have inspired the design, and initial prototypes have been developed in a participatory design process together with kids and teachers from a local school. Evaluation of the recently developed prototypes also took place within the school environment.

2. The HyCon framework for context-aware mobile hypermedia

We have over the years developed a number of hypermedia frameworks and systems, such as Devise Hypermedia (Grønbæk and Trigg 1994), the Arakne framework (Bouvin 2002), and Xspect (Christensen *et al.* 2003). None of these have explored mobility or context awareness, and we have therefore designed a new framework architecture serving as a platform for the development of context-aware services. This section discusses the framework architecture, its implementation, data model, and a prototype application built on the framework.

2.1. Service framework architecture

The HyCon service framework architecture is divided into four layers, as shown in Figure 1. The architectural approach is similar to earlier work, such as the Open Hypermedia System Working Group's Open Hypermedia Model (Reich *et al.* 1999) and Construct (Wiil *et al.* 2001), both heavily inspired by the seminal Dexter architecture (Halasz and Schwartz 1994). Key to the HyCon service framework architecture, compared to earlier work, is an extra layer dedicated to handling sensors and sensor information.

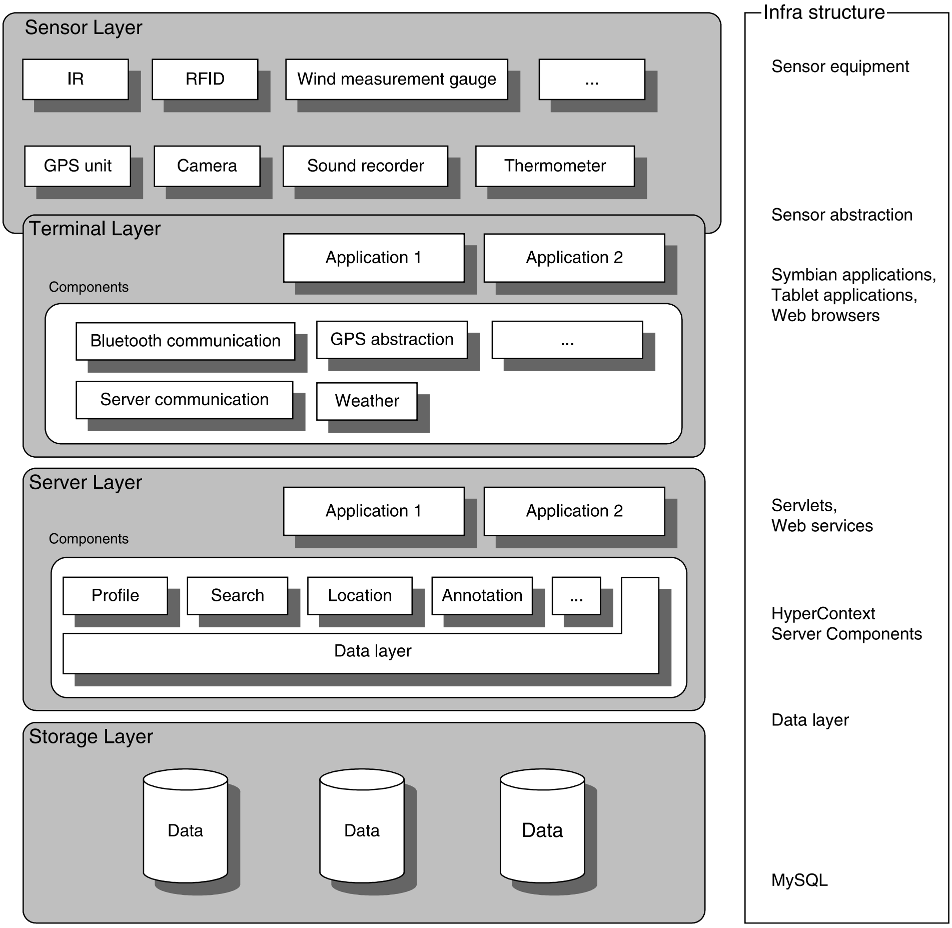


Figure 1 Hycon service framework architecture.

At the bottom of the figure is the storage layer, which handles persistent storage. The storage-layer interface is available to components and applications in the next layer: the server layer.

The components in the server layer form the basic functionality of the framework and include a small set of reusable building blocks such as the data-layer component, the location component, the subscription component, and the annotation component. The components are used to create service applications which implement interfaces to the component's functionality. Applications may use one or several components and add specialised functionality not available through the mixture of components, e.g. computation on the output from different components. This design provides a mechanism for decoupling the responsibilities of the building blocks and not creating mutual dependencies between individual components in the framework.

The purpose of the server layer is to provide data and functionality to the terminal layer. The division between reusable components and specialised applications is found again in the terminal layer: the components are implementing the communication to the server layer, and the applications are implementing functionality and user interfaces appropriate for the terminal device in use. The hardware platform of the terminal layer may include all sorts of mobile equipment: laptops, tablet PCs, PDAs, and cell phones. These devices may have access to sensor information. Sensors can be integrated within the terminal device (e.g. a camera in a phone) or be external (e.g. a Bluetooth-enabled GPS unit). Sensor information is accessed through tailored interfaces to components handling the parsing and computation of the raw sensor data. This sensor abstraction is very similar to the mechanism implemented in the Context Toolkit (described in Section 8). The sensor components in our framework correspond to widgets and interpreters in the Context Toolkit.

2.2. Interfaces and service infrastructure

The infrastructure and interfaces of the service framework can be realised in various ways—one way is the traditional client–server model, which was chosen for our implementation. The functionality of the storage layer is handled by a MySQL[1] server with a database having tables and relations corresponding to object and context relationships. The server layer offers a data-layer component that maps between language objects and the database format. The data layer supports a basic set of methods to store, retrieve, query, and delete objects from the database. This set of methods provides the interface to the storage layer.

The runtime environment for components and applications in the server layer is an application server. The applications are implemented as Java servlets and Java Web services, and thus provide their interfaces via CGI and SOAP to the terminal layer. The CGI version is internally represented as a filter chain with application servlets producing XML data output at the beginning of the chain and an XSLT servlet transforming the data to specific output formats [e.g. HTML or Scalable Vector Graphics (SVG) (Jackson and Ferraiolo 2002)] at the end of the chain. This has been implemented to easily support a wide range of mobile devices with different computation and display capabilities in the terminal layer.

The terminal layer may consist of many different hardware and software platforms, as indicated above. We have experimented with terminal layer platforms ranging from tablet PCs to Symbian phones, and laptops. As sensors and content-capturing devices we have used external Bluetooth-enabled GPS units, built-in sound recorders, and digital cameras (both built-in and external USB cameras).

In order to integrate the context framework with existing online services, the components in the server and terminal layers can communicate and retrieve information from resources other than the storage layer. This enables integration with services providing online maps, weather information, and search engines.

2.3. Data model

The data model implemented in the framework describes the hypermedia structures and context objects used in all layers of the framework. The model shares similarities with earlier hypermedia data models, e.g. the OHSWG navigational data model (Reich *et al.* 1999; Grønbæk *et al.* 2000), but focuses not only on modelling hypermedia objects but also on physical objects, and it utilises hypermedia structures based on the XLink standard (Derose *et al.* 2001). We describe the model in object-oriented terms in this section, but application implementations are not restricted to using this object-oriented model. Mobile applications in the terminal layer may use the CGI interface in the server layer to obtain raw XML representations of the objects, for example, and deal with them in a way appropriate for the specific environment they are running in.

All hypermedia and context objects in the model are described by subclasses of the abstract *AbstractObject* class (see Figure 2). Instances of this class and its subclasses all share some common attributes: they all have globally unique identifiers (GUIDs), a small set of meta-data describing their creator and modification timestamps, and a set of property–value pairs providing extensibility and flexibility to the model.

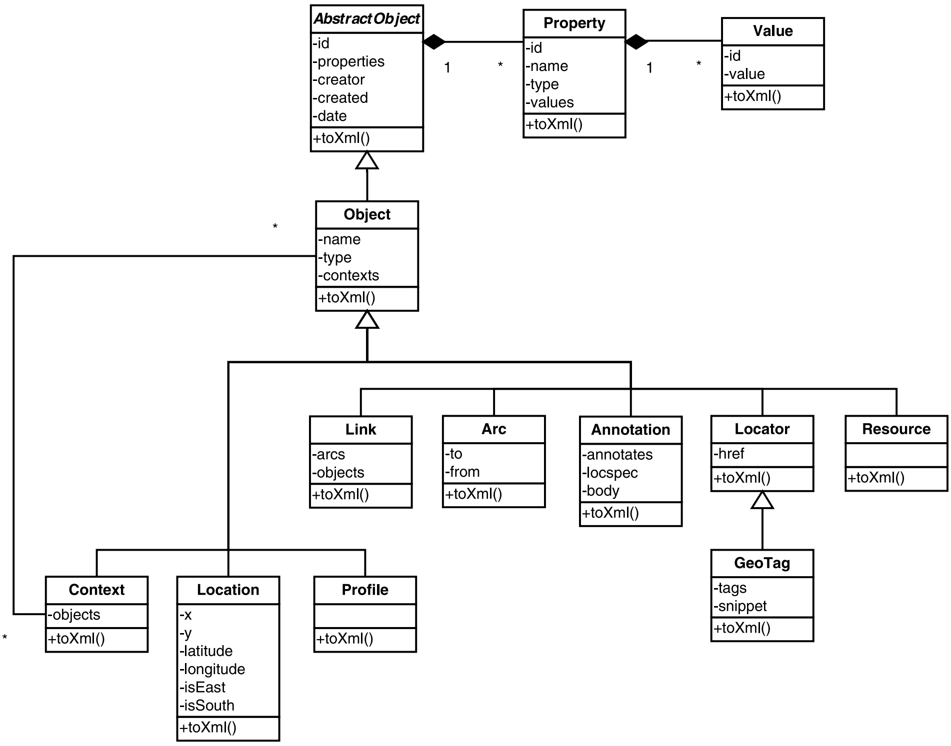


Figure 2 Classes in the HyCon data model.

The generic Object class is used to describe objects not explicitly represented in the model (e.g. new physical objects, persons, things, digital content, etc.). This class makes it easy to implement prototypes dealing with phenomena not represented in the data model without having to change the model.

A number of predefined classes are derived from the Object class. These include: Context, Profile, Location, Link, Arc, Locator, Resource, and Annotation. Instances of the Context class are special objects in the model used to create explicit collections of objects somehow belonging to the same context (e.g. groups of people or available devices in a user's surroundings). Context collections are not disjoined: objects may be part of multiple contexts, and contexts may include several objects. The Profile class describes objects containing preferences of a user or device (e.g. the presentation specification of a terminal device). Objects from the Location class represent points of interest and are used whenever digital information is associated with physical places. The Link, Arc, Locator, and Resource classes describe XLink-based structures on top of the other objects in the system. These classes are discussed further in Section 6. Locator objects are also used to represent digital resources reachable by a URI in the model. The derived GeoTag class is used to tag elements with URIs (e.g. to enable tagging of a location with a URI of a Web page describing the place). Objects from the Annotation class are similar to the GeoTag objects in that they can annotate any element in the model. However, the content of annotations is not limited to a URI, but can include multimedia content such as text, images, sound, and video. Annotations are discussed in Section 5.

2.4. The HyConExplorer prototype

The HyConExplorer prototype is our first implementation of a client application utilising the framework. The prototype is a Java application designed to run on commercially available tablet PCs connected to the network through WLAN or GPRS connections on cell phones (see Figure 3).

The tablet has a mounted camera for capturing low-resolution images, video, and audio in the application. The user's physical location is registered through a Bluetooth-enabled GPS unit light enough to be carried in a shirt pocket. Figure 4 shows the tablet and the HyConExplorer in use, with two students walking our campus area, browsing, and making annotations on points of interest.

As the prototype is developed in Java, it implements the full data model described in the previous section. Communication to services running on the application server is primarily through the SOAP interface, with hypermedia structures and context objects being sent as XML-encoded Java objects. However, upload and download of shared multimedia files are done through the CGI interface.

The prototype implements components handling the communication with the server applications, sensor components handling GPS, and camera-data manipulation. This means that even though our framework is designed to support context information from multiple sensors, we are currently only utilising location

Figure 3 Mobile devices used to run the HyConExplorer.

information (and information derived thereof) in our first prototype. The prototype also implements a map component handling retrieval of public bitmap maps from the Danish chart provider KMS[2]. KMS provides maps in different scales covering every part of Denmark. Based on the user's position, the chosen map scale, and the geometry of the map view in the application, the map component retrieves maps and displays them in an SVG component in the interface. On top of the map, several layers of hypermedia structures and context information are displayed as link markers and other graphical representations.

The HyConExplorer prototype have been used as a test bed for several context-aware hypermedia techniques: context-aware browsing, context-aware searching, and context-aware annotation and linking. These techniques are discussed in the following sections.

3. Context-aware browsing

Navigating information resources by browsing is a typical means of quickly gaining an overview of the information space: readers can flip through the pages in a book and skim the headlines to get an idea of which chapters to read; users of the Web can follow links in HTML pages between related documents and read relevant snippets of text on the pages, etc. Similarly, context-aware browsing is a way to navigate the information stored in a user's digital context by changing parameters in the user's physical context (e.g. changing location).

Figure 4 Context-aware hypermedia on a moblie tablet PC with camera [SitePack (Nielsen *et al.* 2003)]. The tablet has network connection through a Bluetooth- and GRPS-enabled cell phone and reads GPS coordinates from the GPS unit visible in the user's pocket (b).

In our prototypes, we have experimented with both implicit browsing of the digital context and the more common explicitly user-controlled browsing as in Web browsers, for example. When considering mobile hypermedia systems, we can describe these two ways of browsing by adopting the terms *direct physical navigation* and *indirect representational navigation* from Geo-Spatial hypermedia (Grønbæk *et al.* 2002).

Direct physical navigation allows the user to browse information by changing parameters in the physical context. Physically walking or driving from one location to another will affect parameters such as time and location in the physical context, allowing the system to find associated information in the digital context. Systems using this kind of browsing include location-based tourist-information systems and guide systems. We use this kind of browsing in our prototypes to let users equipped with GPS terminals bump into information (annotations, linked documents, or trails of information) when moving about in the physical world. When physically entering a predefined context, the associated information is pushed to the user.

However, it is not always desirable or feasible to have to physically move around to browse information. Indirect representational navigation allows the user to navigate information associated with remote locations by specifying a virtual location in the system. Examples of this kind of navigation include browsing through lists of hotels or restaurants in remote cities while planning a trip home at the office. The GeoTags[3] and GeoURL[4] systems also fall into this category by letting users browse pages by their proximity to a given location. We employ this kind of browsing to allow users to investigate and manipulate the hypermedia structures in the system, even though they are not actually in the physical context associated with the structures. This also addresses the problem encountered with early versions of GUIDE (Cheverst *et al.* 2001 section. 2.1), where only information about physically nearby objects was made available, thus constraining the users' ability to plan ahead or just browse.

Figure 5 illustrates a typical pattern for context-aware browsing based on direct physical navigation. The user, equipped with a mobile terminal (PDA, cell phone, or tablet PC) and GPS equipment walks down a street [Figure 5(a)]. As the user walks, the time and their position are continuously updated in the system. When either has changed by a certain delta, the system makes an update request to the hypercontext server [Figure 5(b)]. The request may include contextual information such as position and time and other parameters such as user ID and nimbus [the region the user projects themself as introduced in the MASSIVE CVE system (Greenhalgh and Benford 1995)]. The context server maps the context information coupled with the user's profile to the hypermedia structures in its database [Figure 5(c)]. These results are returned to the user, and the mobile device is updated accordingly [Figure 5(d)], e.g. with new link markers on a map or items in a list.

In this scenario, no direct user intervention is required—the mobile client samples simple contextual information from the physical environment which is mapped to relevant hypermedia structures by the context server. Especially when dealing with mobile devices with limited user interfaces, where traditional browsing (perhaps

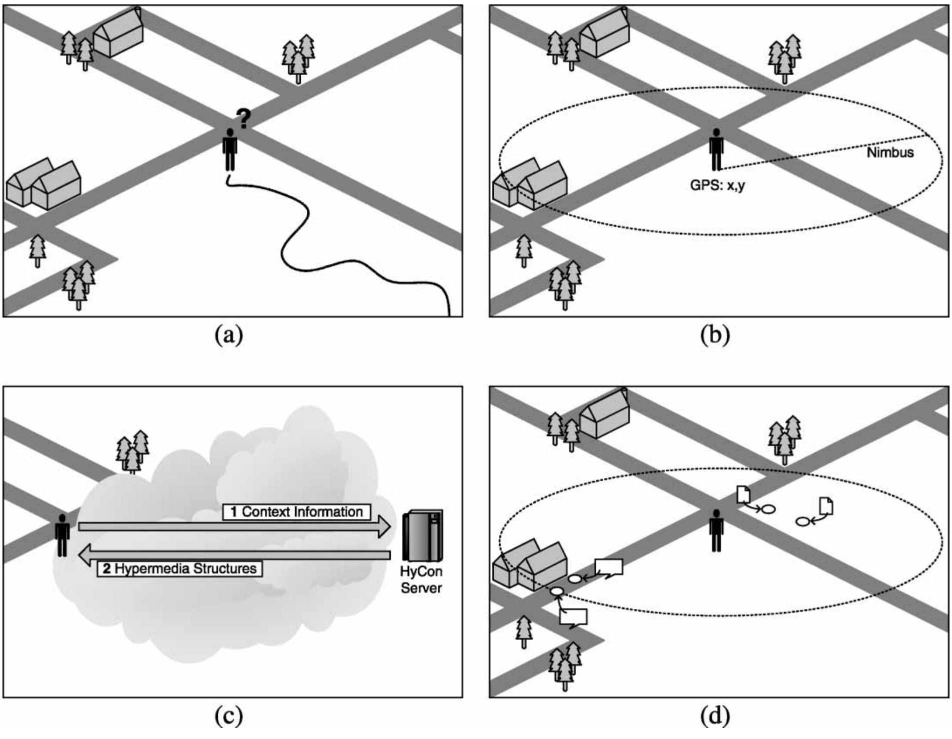


Figure 5 Context-aware browsing.

through multiple layers of menus or lists) quickly becomes tedious, context-aware browsing proves to be an efficient alternative.

4. Context-aware searching

Context-aware searching is based on the general idea of using contextual information as one of the criteria to a search for information. This helps narrow down the information, so the obtained results are more relevant for the user. The contextual information used could be the time of day when looking for shops that are open, the user's location to avoid information of shops in another city, or how many people there are in a group when trying to locate a restaurant where they all can be seated. The benefit of adding criteria based on context information is of course highly dependent on the search engine used and how the context data can be formulated as appropriate criteria.

When using mobile equipment, an inherently interesting part of the context is the location. This single piece of contextual knowledge can be used to locate other resources associated with the location, thus bringing location-relevant information to the user. We have experimented with context-aware searching (Geo-Based Search—GBS) on the Web using location from the gathered context information and keywords on the topic of interest, thus adding a geographical proximity criterion to the search. GBS is an example of a search in existing unstructured information bodies not prepared for context-aware applications in any way.

Figure 6 illustrates a typical pattern for context-aware search. A user is in an unfamiliar neighbourhood and wants to find information about interesting nearby sites [Figure 6(a)]. The location is retrieved from the GPS sensor, and the search radius is determined [Figure 6(b)]. The location information is sent, together with keywords and profile preferences, to the hypercontext server, which forms search terms for the external search engine [Figure 6(c)]. The results from the external search engine are then transformed into virtual hypermedia structures, which can be read and manipulated by the user Figure 6(d)]. Figure 7 shows this kind of context-aware search in the HyConExplorer prototype. Based on a few search terms and the captured context information, the context server has located two roads near the user matching the search. Search hits are marked as dots on the digital map. When activating a link marker, the link endpoint is presented by a snippet of text. If the endpoint looks interesting, it can be viewed in an appropriate external viewer (Web browser, PDF viewer, video player, etc.).

When implementing the GBS service, a transformation from the GPS coordinates to some derived information is necessary, since very few search engines index Web pages by GPS coordinates. However, a database of all public postal addresses in Denmark and their GPS coordinates is freely available from the Danish chart provider KMS. By using this database, postal addresses within a fixed proximity of the location can be determined. Together with the keywords, these postal addresses are much more useful for existing mainstream search engines such as Google. A result of this

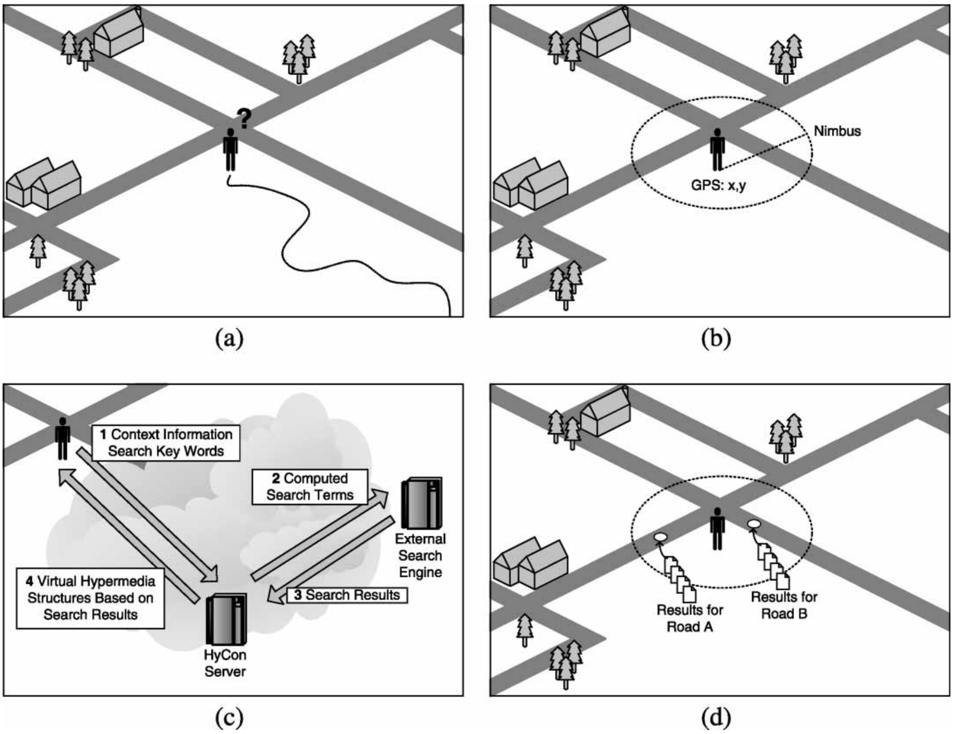


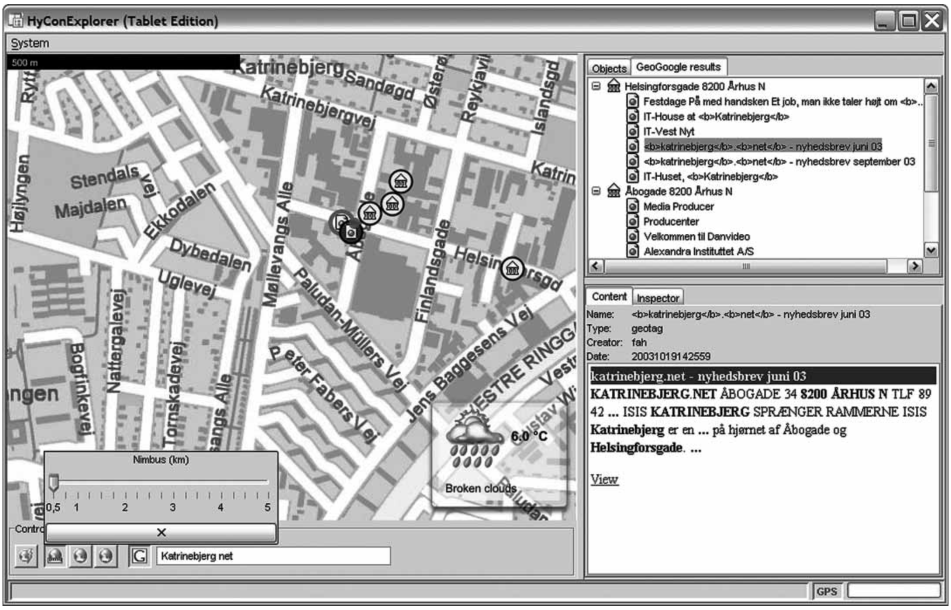
Figure 6 Context-aware search.

kind of search contains Web pages with the keywords and the postal address printed somewhere in the pages, thus presumably covering both the topic of interest and the particular geographical area. The search result is then plotted onto a map over the searched area to intuitively present the connection between geographical locations and Web pages.

An alternative approach to searching for addresses in a fixed distance of the location, is searching by gradually increasing the distance until the first *n* results are found. This technique is somewhat more time-consuming since multiple iterations of searching may be required.

Geo-parsing Web pages by postal addresses is only one of several possible approaches. More sophisticated methods of analysing the content of Web pages can be utilised. McCurley (2001) defines two classes of geo-parsing methods:

- Entity-based, where information of the computing system or the system structure is analysed, e.g. using the URL to locate the server machine that hosts the Web page through domain name registration lookup, or by tracing how IP packets are routed to the host machine and use location information of the backbone routers that is passed on the route. A weakness of this approach is that it assumes a connection between the physical location of a server machine and the material



Helsingforsgade 8200 Århus N
- Festdage På med handsken Et job, man ikke taler højt om ..
- IT-House at Katrinebjerg
- IT-Vest Nyt
- katrinebjerg.net - nyhedsbrev juni 03
- katrinebjerg.net - nyhedsbrev september 03
- IT-Huset, Katrinebjerg

Åbogade 8200 Århus N
- Media Producer
- Producenter
- Velkommen til Danvideo
- Alexandra Instituttet A/S

Name: katrinebjerg.net - nyhedsbrev juni 03
Type: geotag
Creator: fah
Date: 20031019142559

katrinebjerg.net - nyhedsbrev juni 03

KATRINEBJERG.NET ÅBOGADE 34 **8200 ÅRHUS N** TLF 89 42 ... ISIS **KATRINEBJERG** SPRÆNGER RAMMERNE ISIS **Katrinebjerg** er en ... på hjørnet af Åbogade og **Helsingforsgade**. ...

View

Figure 7 Searching and browsing in the HyConExplorer prototype. Dots on the digital map are link markers for information: annotated location, linked documents, and search results. Activating a link marker presents the linked information in the right frame in the window.

provided by it. This might not always be the case, especially when Web pages are placed at Web hotels.

- Content-based, where the content itself is analysed, e.g. determining the language of the page, finding names of geographical sites, finding telephone numbers, finding names of events that only occur certain places, finding names of people, and locating them. Geo-parsing the content of Web pages is only successful if location information is present and is parsed correctly.

Our method of using postal addresses printed in the Web pages falls in the class of the content-based methods. Another content-based method is used by Google's 'Search by Location' service[5] (also described in Section 8), where the postal address is typed in together with the keywords to search for. They describe their geo-parsing method very briefly (in the FAQ) as:

> We analyze the entire content of a page to extract hints or 'signals' that enable us to assign a corresponding physical location, then return results that match the geographic range you specify (e.g. 'near Jacksonville, Florida').

Since none of the geo-parsing methods described here is guaranteed to work for every Web page, they should be combined, thus hopefully resulting in a more robust and detailed analysis of location relevance.

The approach of geo-parsing Web pages to determine a geographic association provides little reliability and poor precision, but the techniques enable integration into existing material not prepared for context-aware search. Another approach is used by the projects Geotags and GeoURL, where meta-tags are inserted into the Web pages defining the corresponding geographic location. This approach provides a very precise and reliable mapping, but relies on the authors of Web pages to embed the meta-tags in their documents, enabling indexing by the service.

5. Context-aware annotation

The browsing and searching techniques discussed in the previous sections present novel approaches to navigation of the information space based on context filtering. This kind of filtering proves especially useful when accessing information on small hand-held devices with limited display and input capabilities such as cell phones and PDAs. However, as stated in the introduction, one of our goals was to go beyond the classical context-aware browsing and navigation systems, and support users as active producers of information in the system.

The combination of context sensors with mobile devices equipped with built-in cameras and microphones allows users to create video, audio, photo, and text annotations, and automatically tag them with captured context information. This supports the creation of material documenting or commenting on an object or situation and automatically having this material inserted and indexed in the system.

We have built different prototypes supporting the creation of multimedia annotations on mobile devices. The first prototype was a simple gallery system (Nørregaard *et al.* 2003) enabling school children (with Nokia 7650 picture phones) to write email messages with image or video attachments and send them to the gallery service. The gallery service created Web pages from the received text messages, attached images, and image meta-data. The pages were used by the pupils to create reports and presentations of their field work (see Section 7). This prototype only utilised simple context information such as the time-stamp in the image meta-data and the sender of the email to establish a measure of the annotation context.

Our second prototype, the HyConExplorer, is based on the framework presented in this paper. When an annotation is created in the HyConExplorer, it is associated with context information captured or computed on the client. This information makes it possible to locate the annotation later by browsing or searching as described earlier.

Figure 8 illustrates a typical pattern for creating context-aware annotations. The user, equipped with a mobile terminal, has arrived at an interesting location which they want to annotate. The HyConExplorer displays link markers on the map for several existing hypermedia objects which they can be annotated: locations, linked documents, annotations, and (virtual) search results [Figure 8(a)]. The user chooses, however, not to annotate any of the existing objects but an area nearby [Figure 8(b)]. After having photographed the area and written a small textual comment, the annotation data are sent as a request to the context server together with collected

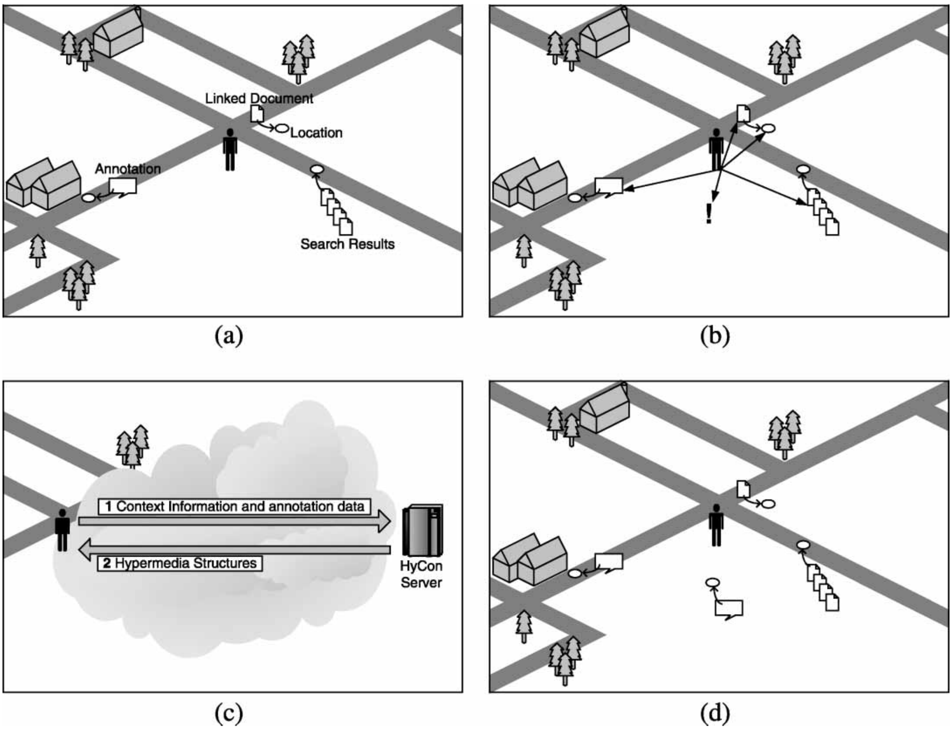
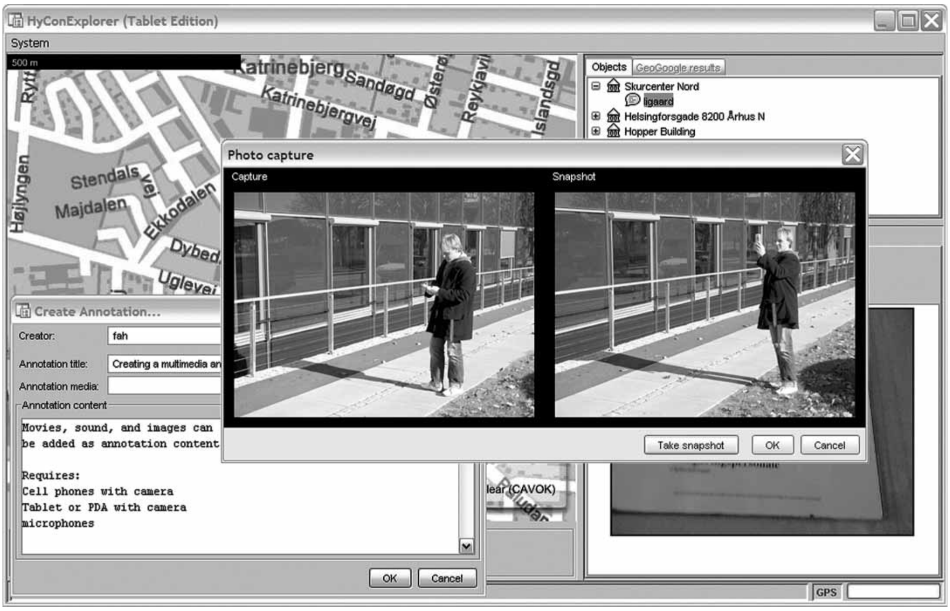


Figure 8 Context-aware annotation.

context information from the application. Since the annotation comments on an object (the location of the new area) not present in the server's database, this object must be created as well as the annotation object. The context server updates the database with the new location and annotation data, and returns the updated structures to the client [Figure 8(c)]. The new annotation is now part of the user's digital context and is displayed in the HyConExplorer [Figure 8(d)]. An example of the creation of annotations in the HyConExplorer prototype is illustrated in Figure 9. The user has typed in a textual comment and is capturing a photo with the tablet PC's camera. Capturing of context information takes place 'behind the scenes' and is not visible to the user—the user only has to deal with creating the content of the annotation.

The data model used in the system is shown in Figure 2. Annotations in the model (represented by the Annotation class) are meta-data objects holding a reference to the annotated object, the textual comment, and a number of properties describing associated media files (video, audio, images, etc.). Since both digital objects (e.g. documents, links, and annotations) and object-representing physical phenomena (real physical objects and physical locations) are modelled as subclasses of the generic Object class in the model, all types of objects in the model can be annotated. Furthermore, this technique allows annotations to annotate other annotations and



Creator: fah

Annotation title: Creating a multimedia an

Annotation media:

Annotation content

```
Movies, sound, and images can
be added as annotation content

Requires:
Cell phones with camera
Tablet or PDA with camera
microphones
```

Figure 9 Creating multimedia annotations in the HyConExplorer prototype.

thus provides a mechanism for creating discussion threads in a given context where users' comments on each other's annotations.

Contrary to many open hypermedia systems, we have chosen to represent annotations as single meta-data objects. Typically, open hypermedia systems support annotations by using fully fledged hypermedia links, linking a local resource from the link-base into some externally stored document (Grønbæk *et al.* 2000; Bouvin *et al.* 2002; Christensen *et al.* 2003). Potentially, this makes it harder to move the annotation from one link base to another, since several components (the local resource, the representation of the remote document, and the link components) have to be computed and assembled before the annotations can be moved. Our approach is more closely related to the RDF-based Web annotations implemented by W3C's Annotea system (Kahan *et al.* 2001). Annotation objects in Annotea are described by a dedicated RDF schema, which includes properties for the type of the annotation, the annotated resource (described by a simple XLink link), a location specifier pointing into the annotated document, the body of the annotation, We have found such a representation to be simpler to handle in the case of annotations. Since the annotations are located as single objects in the link-base, moving and sending them around in the system become more lightweight operation than when dealing with full link structures. Furthermore, since our annotation structures are so closely related to Annotea annotations, it would be trivial to export the structures to RDF for presentation on the Web, e.g. with the Annotea Web browser.

Other context-aware systems concerned with supporting the user as producer of information are the CyberGuide and CyBARguide (Long *et al.* 1996) described in

Section 8. CyBARguide supports users adding new points of interest to the system's database. Though not implemented (Abowd *et al.* 1997) suggest that users should be able to leave comments and impressions as 'virtual graffiti' when visiting a location. This is similar to creating an annotation on a visited location in the HyConExplorer.

6. Context-aware linking

In our prototypes, we have primarily experimented with links which would make good sense in a mobile hypermedia context. This has resulted in simple links (2-ary links) linking external documents to objects in the link base and extended links (*n*-ary links) linking a collection of objects (typically locations) into a single logical trail trough the objects.

The Web presents a massive corpus of information, and much of the information available through the Web may be relevant in a given physical context. Thus, supporting links to existing Web pages is an effective means of integrating this information into the system.

External resources are represented by instances of the Locator class in the data model (see Figure 2). The locator object acts as a document proxy [RefSpec (Grønbæk and Trigg 1995)] with a URI reference to the actual document. The URI may optionally include a fragment identifier (LocSpec) specifying a point or region in the document. The GeoTag class in Figure 2 represents a special simple link connecting a remote resource (given by the URI reference) to an object in the system's database. The GeoTag objects also include a snippet attribute containing a small textual description of the linked page. We have added this attribute to hold extra information about pages when linking search results from external search engines into the system, but the GeoTags also serve as a general mechanism for tagging/linking objects with remote resources.

Guided tours as a hypermedia concept go back to Bush (1954) and have seen many implementations and variations since then, such as tabletops (Trigg 1988), Walden's Paths (Furuta *et al.* 1997), or Scripted Documents (Zellweger 1989). In HyCon, we use extended *n*-ary links to express trails or paths through a collection of objects. In mobile hypermedia, this will typically mean trails through a collection of locations, where the link represents a guided tour and the locations represent stops on the tour. This mechanism has several interesting applications: in a school environment, a teacher may prepare a field trip for the school children by creating a guided tour through an area the children are to investigate. Each stop on the tour may include a description of the site and additional linked material. This material may be divided into several different layers of information: history, archaeology, politics, environmental info, nature guidance, etc. Upon arriving at a stop, the pupils are presented with one or more of the layers, depending on their mode and defined task. Furthermore, the pupils can collect further material at the site and add annotations commenting on the material and on the teacher's predefined material through the linking and annotation mechanisms. At home in the classroom, the trail of information may serve as a way to present the results of the field trip for their class mates. In another scenario, the pupils may diverge from the predefined trail and create their own trail as they walk along. Other groups of pupils working simultaneously in the same area may bump into the newly created trail and instantly observe the results of diverging from the teacher's guided tour.

The linking mechanism used to describe trails is derived from our earlier experiences with XLink-based guided tour structures in the Xspect system (Christensen *et al.* 2003). In Xspect, a single XLink link-type element constituted a guided tour with XLink locator-type elements as stops on the tour and XLink arc-type elements as directed edges (or tour segments) between the stops. In the current data model (see Figure 2), we have generalised the stops on the tour to be instances of the general Object class and thus support easy linking of both Location-type and Locator-type objects. Instances of the Arc class still describe the traversal behaviour of the link, so the result is XLink-based structures on top of the objects in the system. This design makes it easy to export trails in the system to XLink structures which may be used for Web presentations, for example.

7. Use scenarios and evaluation

School projects have inspired the design of the HyConExplorer and the HyCon framework. Initial prototypes were developed in a participatory design process together with kids and teachers from a local school (Nørregaard *et al.* 2003).

Children engaged in learning activities away from the classroom may benefit from context-aware hypermedia services in several ways. Modern pedagogy is project-oriented and stimulates pupils to work in groups where they research a subject and make some report, multimedia production, and/or oral presentation. For such learning activities, pupils and students need to document their exploration of subjects outside the class room. This requires some mobile tools to support the documentation and preliminary production of content for the final reporting. Thus, pupils may carry mobile devices that let them collect, organise, and work with information they produce or find through our context-based hypermedia services. The mobile devices may be used to capture and collect information in the field and as an interface to start presentations of the collected information in interactive room environments in schools, universities, libraries, or other learning environments.

With the HyCon framework presented in this paper, we can provide context-based information services for cities, landscapes, museums and other cultural heritage. Possible layers of information to browse and search based on the context may include: history, archaeology, politics, environmental info, nature guidance, demography, biology, energy, trade, industry, transport, and traffic. All of these subjects may be explored relative to a given context, e.g. location, time, project task, or group members.

The collaboration and linking services can be used by other groups who research on the same subject in the same area. A group may at some stage bump into information created by another group at a particular spot at an earlier stage. The other group may also discover that the previous group has left a trail with a collection of linked information, which can be useful for their own research on the subject with their perspective. During the evaluation of the HyConExplorer, one teacher suggested that data produced by pupils investigating a biotope could be saved year after year. When new pupils return to the biotope, they can compare their findings with the data from earlier years and thus investigate the evolution of the biotope over time.

At this stage, we have conducted a number of smaller experiments with school children (Nørregaard *et al.* 2003) using the gallery service and Nokia 7650 picture phones as the documentation and browsing tool. These experiments have inspired the development of the services for the HyConExplorer. The HyConExplorer has been evaluated by teachers and children from three local schools. The teachers were involved in framing a realistic project work scenario on which the evaluation workshop was based. The theme of the workshop was a project called 'Our City', where groups of pupils were sent out to document on different areas of Aarhus. The evaluation was conducted in four steps.

In the first step, the teachers were shown the software and had time to experiment with it. In this step, they used the HyConExplorer to collect initial teaching material and link it to locations in Aarhus. This first step allowed the teachers to create a foundation for the project and control the material the children would find when moving about in the city. In the second step, a member from our team introduced the children to the hardware and the HyConExplorer, and the teachers presented the field assignment to the children. The children were given a few minutes to explorer the tools and ask questions. In the third step, the children were split into three groups and sent to different parts of the city (Figure 10). The children investigated their part of the city by browsing the material the teachers previously had linked into the system and by creating annotations on the spot. Each group was accompanied by two

Figure 10 Children browsing and collecting material in the city.

persons from our team—one acting as technical backup and one recording the children with a video camera. In the final step, the children returned to the classroom and created presentations with the HyConExplorer's built-in presentation tool (allowing presentations to be compiled directly from the hypermedia structures in the system). Each group presented their findings to the other groups and concluded on their experiences with the use of the system.

Both teachers and children were generally excited about the features provided by the HyConExplorer. The ability to browse and search for information in context and the association between the physical world, the map, and the information seemed to work well. However, several issues became apparent during the workshop.

One group of children chose to split all of the equipment (tablet PC, cell phone, and GPS) amongst them and carry it in their hands. Another group put the cell phone in their bag and strapped the GPS unit to the bag. The first group complained about the amount of equipment they had to carry, and it seemed difficult for them to understand that they had to stay within a few metres from each other for the Bluetooth units to communicate properly. The second group did not have these problems, but both groups complained about the weight of the tablet PC. Thus, we saw a clear need to 'black-box' the various units so that the system would seem simpler. One approach to this could be simply to hide the equipment, as the second group did, so that the focus would only be on the tablet. Another approach would be to use more integrated hardware, e.g. cell phones with built-in GPS receivers, thus effectively decreasing the number of physical devices needed.

Another issue which quickly became apparent was the limited network bandwidth on the GPRS-based cell phones. We experienced data-transfer rates of approximating 20 kbit/s when receiving data and even less when uploading images and other data to the server. This made the communication with the server slow when browsing and especially when creating annotations. The children found the task of creating annotations tedious and boring, as they had to wait for 10–30 s after having created an annotation to ensure that the data were properly stored on the server. As the bandwidth on new phones increases, this issue will become less important, but we are considering implementing local caching and buffering on the clients. This will result in faster operations when producing material and will work better with devices with limited network bandwidth. The local buffer needs to be synchronised with the main system. This can be done either when arriving at a hot-spot offering higher bandwidth connections or by having a separate background thread continuously sending data from the buffer to the server.

8. Related work

This section gives a number of examples of systems and frameworks that in one form or another support context awareness.

Geo-spatial hypermedia (Grønbæk *et al.*, 2002, 2003) is a special case of context-aware information system, where the focus is primarily on associations between locations (in the real and the virtual) and digital material (be it multimedia annotations, links to technical documentation, sightseeing information, etc.). Geo-spatial hypermedia systems can be broadly divided into two categories, depending on the support for user content authoring in the field.

A number of systems have explored the use of location in a hypermedia context. The approaches vary from using location as triggers for other actions to more traditional information systems. Location is not the only kind of context that might be considered for a context-aware system, but it is the most prevalent. The main difference is how the location is used, e.g. for locating the nearest printer, triggering nearby sensors, or recording the location of the user.

The Xerox PARCTab (Want *et al.* 1995), the first context-sensitive computer, was developed in 1992. The PARCTab took advantage of the simple idea of location awareness by using a palm-sized computer (the 'tab') with a pen interface linked to an infrared network. Limited by the processing power of the tab, applications were executed in a full-size computer infrastructure, and the results were transmitted over the IR network and displayed on the display of the tab. The cellular network provided location information on a room-by-room basis. The resulting PARCTab applications became the first location-based applications on PDA-like devices.

Long *et al.* (1996) and Abowd *et al.* (1997) discuss the Cyberguide project, which was an attempt to replicate a human tour-guide service through the use of mobile and hand-held technology and ubiquitous positioning and communication services. Tour guide systems have been built for both indoor and outdoor environments within and around the Georgia Tech campus. One of the prototypes, called CyBARguide, was running on a Newton MessagePad equipped with a GPS receiver. Users driving with the CyBARguide could locate a point of interest that satisfied certain requirements (special offers, free parking, good ambiance). In addition, the authors suggest future support for users leaving comments that would become available to other users, after having visited a point of interest.

The Context Toolkit (Dey 2001; Dey *et al.* 2001) makes it possible to add the use of context to existing non-context-aware applications and to evolve existing context-aware applications. The toolkit is built on a layered architecture, and the architecture makes context-aware applications resistant to changes in the context-sensing layer. It encapsulates changes such that applications do not need to be modified. The architecture is built on the concept of enabling applications to obtain context parameters independently of how the context was acquired. The architecture also introduces a so-called *context widget*. A context widget acquires a certain type of context information and it makes that information available to applications in a generic manner, regardless of how it is actually sensed. Applications can access context from widgets using traditional poll and subscribe methods, similar to graphical user interface (GUI) widgets. In addition to context widgets, the Context Toolkit supports *interpreters*, which transform context information by raising the level of abstraction, and *aggregators* grouping logically related context information.

Geographical Information Systems (Laurini and Thompson 1992) have been used for many years to combine layers of data with maps. The uses for such systems have diversified, and they have become quite widespread. In comparison with this work described herein, most GIS applications differ by the focus on overview rather than localised context. Most GIS systems are still data-heavy applications mainly used on stationary machines.

An example of a mobile GIS application is described by (Stockus *et al.* 1999). consisting of a Java applet running on a laptop computer with a GPS unit and connected to the Internet through a cell phone. The system allowed the user to see a representation of themselves superimposed on layers of maps downloaded to the computer.

The Topos system (Grønbæk *et al.* 2002, 2003) is a document-centred collaborative virtual environment originally aimed at architects. It enables users to collect and arrange collections of documents, models and other material in 3D spaces. The system has explored a number of interfaces, such as 3D displays and large touch-sensitive screens. In order to explore the coupling between the virtual and the physical, the system has also found use on GPS-equipped tablet PCs. By coupling the virtual 3D spaces (often containing architectural models and many related documents) with the real world and the buildings virtually depicted, it becomes possible for users to interact with documents, e.g. as they inspect a building site (and for other users to see on-site users' whereabouts). By using the ARToolKit, users of Topos can, by placing markers in the field, judge the impact of new buildings, as 3D models are overlaid on the recorded video of the location.

Nielsen *et al.* (2003) present a mobile AR system, called the SitePack supporting architects in annotating and visualising models in real time. The SitePack applies a novel combination of GPS tracking and vision-based feature tracking in its support for architects. One example of use is Zone of Visual Impact assessment in outdoor environments. Another example is indoor facility management support, where maintenance workers may retrieve and visualise models of electrical and heating installations, for example, based on the context in which they work.

Furuta and Na (2002) introduce the notion of responsive hypertext provided by a version of the petri-net based Trellis system caT (context-aware Trellis). caT supports specifying characteristics of the external environment in which the hypertext is being used (e.g. the reader's location, time of day, user characteristics such as age and job title, etc.). caT relies on the user providing the context information, and the hypertext must be prepared by its authors to respond to context parameters.

The Ambient Wood (Weal *et al.* 2003) is a project aimed at augmenting schoolchildren's learning experience in a woodland setting. As the students move about in the wilderness, their handheld computer sets off various sensors, allowing their progress to be tracked as well as creating an interactive experience (through hidden speakers, etc.). When the students have finished their exploration of the setting, they can return to the classroom and reflect and report on their exploration based on the recorded data. Thus, while the students create content by moving and acting in an open space, the editing takes place afterwards.

The GUIDE project (Cheverst *et al.* 2002) provides tourists with information about sites in Lancaster and suggests other sites that might be of interest. The GUIDE system calculates its position relative to 802.11 base stations placed at Lancaster attractions. This makes for relatively coarse resolution but averts the problems inherent with GPS navigation between buildings. The GUIDE system has been tested successfully in real use (Cheverst *et al.* 2000), thus supporting the concept of electronic tourist guides.

A shared characteristic of many AR and location-based systems described herein is the emphasis on navigation support and information access. Systems aimed at supporting the *authoring* of material in the field are much less common. While the ability to find one's way or access prerecorded data is certainly valuable, recording one's own impressions is similarly important. This allows the user to be active rather than a passive receptacle. While boards of tourism and other official bodies doubtlessly can create high-quality content, it is crucial that the ability to author (and therefore to critique) content is generally available. The proliferation of blogs and other 'grass root' Web sites illustrates the willingness of people to freely create content for general consumption (e.g. the user-submitted (and unpaid) reviews found on Amazon). A tourist might, for example, visit a nearby restaurant based on the recommendation of an official city guide and then later rate the restaurant by adding a new annotation to the system.

Geo-spatial hypermedia systems in general allow users to search for and browse existing annotations and structures related to a specific location. The ability to create a search engine query based on an address generated from a GPS location is a novel approach (as described in Section 4) that provides access to much online material that has not been specifically authored for geo-spatial use or registered into the system.

Alexa[6] is an example of a context-aware system not utilising location. Alexa is the system behind 'What's Related' found in many Web browsers to help users discover new Web pages. Alexa creates a context around a Web page primarily by storing the browsing trails generated by its many users. Web sites close to each other on these trails are judged to be related and are thus presented as contextually close to other users. For a further discussion on this and similar techniques, see Fu *et al.* (2000).

Web sites dynamically adapting to their users are quite common—a well-known example is the book recommender in Amazon[7]. This system recommends new books to the customer based on the customer's previous buying habits and the aggregated buying habits of customers with similar tastes.

The Google Search by Location service (described in Section 4) is similar to our approach inasmuch as it allows the user to search for Web pages pertaining to a given address. The difference is that the Google system requires the user to enter the address in question, whereas we allow the user to search for 'here' (the user might not know the current address), leaving the system to compute the closest addresses and then feed that to the search engine. As such, the Google Search by Location would work very nicely in conjunction with our system by moving the burden of generating a correct query for a given address from our system to Google. Similarly, geo-tagged Web pages (see Section 4), should they become widespread, would fit well with our approach.

9. Future work

Context-aware hypermedia is still a relatively new field, and there are a number of areas for future development. The area is very much pushing the edge of what is technologically feasible, and with technological progress come new possibilities and challenges.

The continuing miniaturisation of electronics will ensure the integration of GPS into common handheld devices—already, GPS is available in high-end cell phones and PDAs. Similarly, the bandwidth available for wireless devices will increase with the adoption of 3G phone networks and the proliferation of urban WiFi hotspots.

We are currently actively pursuing the development of a Symbian[8] version for mobile phones. This development takes place in a joint project between the University of Aarhus, the University of Aalborg, and two private Danish companies; Innovation Lab[9] and Euman[10]. One of the objectives of this project is to be able to present information such that the context-aware hypermedia applications can be provided for a broad spectrum of platforms and user interfaces. This requires development of a cross-platform concept for user interfaces for context-aware services. Here, we are investigating interfaces described solely in SVG. We see great promise in utilising SVG as an interface language, since advanced interfaces may be produced on the server side (from XML data and XSLT stylesheet transformations) and sent to a variety of different clients. This can be done without implementing specific interfaces for each client platform, thus raising the level of abstraction and decreasing development time.

Another interesting area to be further investigated is the concept of context-aware collaboration where the user's context information within a project group is explicitly and/or implicitly shared to support collaboration.

A simple yet effective mechanism is the 'Pal Finder', where each member of a group has a list in their profile of other group members. The group members are presented with appropriate aspects of their contexts, e.g. their physical position plotted onto a map, their current activity, or live video broadcasts from a camera on their terminal device. In this way, the Pal Finder supports synchronous sharing of context information between members of a project group.

Participating in synchronous sharing of context information requires the users to be working at the same time. An alternative approach is to save the context information on a shared store, so other group members at a later time can access it and reconstruct the situation. In mobile hypermedia, the recorded context information could be structured as trails, as described in Section 6. Group members can, after having recorded a guided tour annotate, link and manipulate it, as any other hypermedia structure. So, by making trails of context information organised as guided tours, asynchronous sharing is supported. This resembles Bush's (1954) original vision of using trails to share information, but the trails we have described take the form of paths between physical locations.

When collaborating using shared resources such as trails, notifications to group members of changes made to the resources allow group members to be aware of others' work and progress without inspecting all parts of the shared resources. The concept could be extended to include notifications of changes in group members' context information, e.g. a notification could be sent when a specific group member turns thier live video broadcast on.

The current prototypes have not featured many annotations at one spot—broader adoption would entail far more common spatial collisions of annotations and other structures. While abundance of content is a boon, it quickly becomes unmanageable and confusing. This may be addressed by adapting to the user's preferences and prior behaviour, as done in adaptive hypermedia. Location and history may give good indicators of relevance, e.g. a newcomer/tourist to an area may be in interested in material not as pertinent to the local residents. These areas of research are still open to us.

10. Conclusion

In this paper, we have introduced the HyCon framework for context-aware mobile hypermedia. We have surveyed a number of different systems and frameworks that deal with context awareness in different areas. Based on this and previous work on open hypermedia, we have proposed and implemented a context-aware mobile hypermedia framework with a layered architecture, data structures, servers and clients providing open hypermedia support for a variety of mobile devices that integrates facilities to sense a set of context parameters. The HyCon architecture includes interfaces for an open Sensor tier which models relevant sensors like GPS, thermometers, RFID antennas, and the like. HyCon provides its services in two versions, namely as SOAP interfaces and as CGI interfaces. It represents the hypermedia objects as structures derived from the XLink and RDF standards. For the visualisations in the interface, the device-independent SVG format is utilised, where possible, to keep the client code device independent. A prototype, called the HyConExplorer, built within the framework is presented, and it is illustrated how classical hypermedia features such as browsing, searching, annotating, linking, and collaboration are supported in context-aware hypermedia. Among the features of the HyConExplorer are real-time location-based searches via Google collecting hits within a specified nimbus around the user's GPS position. Finally, use scenarios for the use of the HyConExplorer in public school projects are discussed. The school projects focus on outdoor exploration of cities or landscapes. An example is exploring an area where archaeological findings have been geo-tagged in a database, such that a user of the HyConExplorer may ‘bump into’ the findings on the mobile device and make picture annotations and links between relevant information.

We see many prospects of further development of these technologies; the HyConExplorer will be further developed also for cell phone devices and will be utilised in the recently started project on interactive school environments under the Centre for Interactive Spaces[11].

Acknowledgements

The work has been funded by the Danish National Centre of IT research through project #333, ContextIT. ContextIT is associated the Centre for Pervasive Computing[12]. We wish to thank all of our colleagues in the ContextIT project, in particular Ole Iversen and Christina Nielsen for their inspirational empirical work with the school kids. Finally, we wish to thank Architect Gunner Kramp, for his customisation of the Tablet PC to run the HyConExplorer.

Notes

[1] http://www.mysql.com/
[2] http://www.kms.dk
[3] http://www.geotags.com/
[4] http://www.geourl.org/
[5] http://labs.google.com/location/
[6] http://www.alexa.com/
[7] http://www.amazon.com/
[8] http://www.symbian.com/
[9] http://www.innovationlab.net/
[10] http://www.euman.dk/
[11] http://www.interactivespaces.net/
[12] http://www.pervasive.dk/

References

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), pp. 421–433, 1997.

I. Beaumont, “User modeling in the interactive anatomy tutoring system ANATOM-TUTOR”, *User Modeling and User-Adapted Interaction*, 4, pp. 21–45, 1994.

N.O. Bouvin, “Augmenting the Web through open hypermedia”, *The New Review of Hypermedia and Multimedia*, 8, pp. 3–26, 2002.

N.O. Bouvin, P.T. Zellweger, K. Grønbæk and J.D. Mackinlay, “Fluid annotations through open hypermedia: Using and extending emerging Web standards”, in *Proceedings of the 11th International World Wide Web Conference, Honolulu, May, W3C*, New York: ACM Press, 2002, pp. 160–171.

P. Brusilovsky and R. Rizzo, “Map-based horizontal navigation in educational hypertext”, in *Proceedings of the 13th ACM Hypertext Conference, College Park, MD, June*, K.M. Anderson, S. Moulthrop and J. Blustein, Eds., New York: ACM Press, 2002, pp. 1–10.

V. Bush, “As we may think”, *The Atlantic Monthly*, 176(1), pp. 101–108, 1945.

K. Cheverst, N. Davies, K. Mitchell, A. Friday and C. Efstratiou, "Developing a context-aware electronic tourist GUIDE: some issues and experiences", in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York: ACM Press, 2000, pp. 17–24.

K. Cheverst, N. Davies, K. Mitchell and C. Efstratiou, "Using context as a crystal ball: Rewards and pitfalls", *Personal and Ubiquitous Computing*, 5(1), pp. 8–11, 2001.

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

B.G. Christensen, F.A. Hansen and N.O. Bouvin, "Xspect: bridging open hypermedia and XLink", in *Proceedings of the 12th International World Wide Web Conference, Budapest, Hungary, May, W3C*, New York: ACM Press, 2003, pp. 490–499.

P. De Bra, G.-J. Houben and H. Wu, "AHAM: a Dexter-based reference model for adaptive hypermedia", in *Proceedings of the 10th ACM Hypertext Conference, Darmstadt, Germany, February*, K. Tochtermann, J. Westbomke, U.K. Wiil and J.J. Leggett, Eds., New York: ACM Press, 1999, pp. 147–156.

N.M. Delisle and M.D. Schwartz, "Contexts—a partitioning concept for hypertext", *ACM Transactions on Information Systems*, 5(2), pp. 168–186, 1987.

S. Derose, E. Maler, D. Orchard and B. Trafford (Eds.), XML Linking Language (XLink). W3C Recommendation 27 June 2001, W3C, June 2001. http://www.w3.org/TR/xlink/.

A.K. Dey, "Understanding and using context", *Personal and Ubiquitous Computing*, 5(1), pp. 4–7, 2001.

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), pp. 97–166, 2001.

S.R. El-Beltagy, W. Hall, D.D. Roure and L. Carr, "Linking in context", in *Proceedings of the 12th ACM Hypertext Conference Århus, Denmark, August*, H. Davis, Y. Douglas and D.G. Durand, Eds., New York: ACM Press, 2001, pp. 151–160.

R. Furuta, F.M. Shipman III, C.C. Marshall, D. Brenner and H.-W. Hsieh, "Hypertext paths and the World-Wide Web: Experiences with Walden's Paths", in *Proceedings of the 8th ACM Hypertext Conference, Southampton, April*, M. Bernstein, L. Carr and K. Østerbye, Eds., New York: ACM Press, 1997, pp. 167–176.

R. Furuta and J.-C. Na, "Applying caT's programmable browsing semantics to specify World-Wide Web documents that reflect place, time, reader, and community", in *Proceedings of the 2002 ACM Symposium on Document Engineering*, New York: ACM Press, 2002, pp. 10–17.

X. Fu, J. Budzik and K.J. Hammond, "Mining navigation history for recommendation", in *Proceedings of the 5th International Conference on Intelligent User Interfaces*, New York: ACM Press, 2000, pp. 106–112.

C. Greenhalgh and S. Benford, "MASSIVE: a collaborative virtual environment for teleconferencing", *ACM Transactions on Computer–Human Interaction*, 2(3), pp. 239–261, 1995.

K. Grønbæk and R.H. Trigg, "Design issues for a Dexter-based hypermedia system", *Communications of the ACM*, 37(2), pp. 40–49, 1994.

K. Grønbæk and R.H. Trigg, "Toward a Dexter-based model for open hypermedia: Unifying embedded references and link objects", in *Proceedings of the 7th ACM Hypertext Conference, Bethesda, MD, March*, 1996, pp. 149–160.

K. Grønbæk, L. Sloth and N.O. Bouvin, "Open hypermedia as user controlled meta data for the Web.", in *Proceedings of the 9th International World Wide Web Conference, Amsterdam, May W3C*, 2000, pp. 553–566.

K. Grønbæk, P.P. Vestergaard and P. Ørbæk, "Towards geo-spatial hypermedia: Concepts and prototype implementation", in *Proceedings of the 13th ACM Hypertext Conference, College Park, MD, June*, K.M. Anderson, S. Moulthrop and J. Blustein, Eds., New York: ACM Press, 2002, pp. 117–126.

K. Grønbæk, J. F. Kristensen, P. Ørbæk and M. A. Eriksen, "'Physical hypermedia': organising collections of mixed physical and digital material", in *Proceedings of the 14th ACM Hypertext Conference, Nottingham, UK, August*, L. Carr and L. Hardman, Eds., New York: ACM Press, 2003, pp. 10–19.

L. Hardman, D.C.A. Bulterman and G. Van Rossum, "Links in hypermedia: the requirement for context", in *Proceedings of the 5th ACM Hypertext Conference, Seattle, WA November*, New York: ACM Press, 1993, pp. 183–191.

F.G. Halasz and M. Schwartz, "The Dexter hypertext reference model", *Communications of the ACM*, 37(2), pp. 30–39, 1994.

D. Jackson and J. Ferraiolo, Scalable Vector Graphics (SVG) 1.1. W3C working draft, W3C, February 2002. http://www.w3.org/TR/SVG11/.

J. Kahan, M.-R. Koivunen, E. Prud'Hommeaux and R.R Swic, "Annotea: An open rdf infrastructure for shared web annotations", in *Proceedings of the 10th International World Wide Web Conference*, *Hong Kong, May, W3C*, 2001.

R. Laurini and D. Thompson, *Fundamentals of Spatial Information Systems*, London: Academic Press, 1992.

S. Long, R. Kooper, G.D. Abowd and C.G. Atkeson, "Rapid prototyping of mobile context-aware applications: the Cyberguide case study", in *Proceedings of the 2nd Annual International Conference on Mobile Computing and Networking*, New York: ACM Press, 1996, pp. 97–107.

K.S. McCurley, "Geospatial Mapping and Navigation of the Web", in *Proceedings of the 10th International World Wide Web Conference, Hong Kong, May, W3C*, 2001, pp. 221–229.

D.E. Millard, L. Moreau, H.C. Davis and S. Reich, "Fohm: a fundamental open hypertext model for investigating interoperability between hypertext domains", in *Proceedings of the 11th ACM Hypertext Conference, San Antonio, TX May 2000*, K.M. Anderson, Ed., New York: ACM Press, 2000, pp. 93–102.

D.E. Millard, D.T. Michaelides, D.D. Roure and M.J. Weal, "Beyond the traditional domains of hypermedia", in *Proceedings of the International Workshop on Open Hypermedia Systems Core Concepts and Research Directions, Pre-Conference Workshop at the ACM 13th International Conference on Hypertext and hypermedia (HT02), University of Maryland, College Park, MD June 2002*, D. Millard, J. Haake and S. Reich, Eds., Informatik Berichte, FernUniversität Hagen, Hagen, 2002, pp. 26–32.

M.B. Nielsen, G. Kramp and K Grønbæk, "Mobile augmented reality support for architects based on feature tracking techniques", in *Proceedings of the Workshop on Interactive Visualisation and Interaction Technologies, Krakow, Poland*, Berlin: Springer, 2004.

P. Nørregaard, J. Andersen, C. Dindler, J. Frich, O.S. Iversen and C. Nielsen, "Networking news—a method for engaging children actively in design", in *Proceedings of the 26th Information Systems Research Seminar in Scandinavia (IRIS)*, 2003, p. 26.

S. Reich, U.K. Wiil, P.J. Nürnberg, H.C. Davis, K. Grønbæk, K.M. Anderson, D.E. Millard and J. Haake, "Addressing interoperability in open hypermedia: the design of the open hypermedia protocol", *The New Review of Hypermedia and Multimedia*, 5, pp. 207–248, 1999.

B.N. Schilit and M. Theimer, "Disseminating active map information to mobile hosts", *IEEE Network*, 8, pp. 22–32, 1994.

B.N. Schilit, N.I. Adams and R. Want, "Context-aware computing applications", in *Proceedings of the Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, December*, IEEE Computer Society, 1994, pp. 85–90.

A. Stockus, A. Bouju, F. Bertrand and P. Boursier, "Integrating GPS data within embedded Internet GIS", in *Proceedings of the 7th ACM International Symposium on Advances in Geographic Information Systems*, New York: ACM Press, 1999, pp. 134–139.

R.H. Trigg, "Guided tours and tabletops: tools for communicating in a hypertext environment", in *Proceedings of the 1988 ACM conference on Computer-Supported Cooperative Work*, New York: ACM Press, 1988, pp. 216–226.

R. Want, B. Schilit, A. Norman, R. Gold, D. Goldberg, K. Petersen, J. Ellis and M. Weiser, "An overview of the PARCTab ubiquitous computing experiment", *IEEE Personal Communications*, 2(6), pp. 28–43, 1995.

M. Weiser, "The computer for the 21st century", *Scientific American*, 265(3), pp. 66–75, 1991.

M.J. Weal, D.T. Michaelides, M.K. Thompson and D.C. Deroure, "The ambient wood journals: replaying the experience", in *Proceedings of the 14th ACM Hypertext Conference, Nottingham, UK, August*, L. Carr and L. Hardman, Eds., New York: ACM Press, 2003, pp. 20–27.

U.K. Wiil, D.L. Hicks and P.J. Nürnberg, "Multiple open services: A new approach to service provision in open hypermedia systems", in *Proceedings of the 12th ACM Hypertext Conference*, H. Davis, Y. Douglas and D.G. Durand, Eds., Århus, Denmark, August, 2001, pp. 83–92.

P.T. Zellweger, "Scripted documents: a hypermedia path mechanism", in *Proceedings of the 2nd ACM Hypertext Conference, Pittsburgh, PA, November*, New York: ACM Press, 1989, pp. 1–14.