Augmentation of Content with Context Meta-Data

Andreas Zimmermann and Andreas Lorenz

[1] Fraunhofer Institute for Applied Information Technology
Schloss Birlinghoven
53754 Sankt Augustin, Germany
{Andreas.Zimmermann, Andreas.Lorenz}@fit.fraunhofer.de

Abstract. In the field of wearable computing the information selection and presentation should be adapted to the professional and his current context of use. Nowadays, uncounted Content-Management Systems provide access to huge amount of information, but without context, information is just data. To kick off the next step for intelligent information systems the article introduces a new approach for the management of content in combination with context information. For the achievement of a personalized and contextualized information delivery the presented approach will describe three steps: Context Modelling, Context Collection and Content Presentation.

1 Introduction

Wearable computing empowers professionals to higher levels of productivity. It provides effective forms of access to knowledge and computing power anywhere and in any situation. The new technology-paradigm meets the growing demand on professionals to become more flexible and efficient in an increasingly challenging work environment. Because the mobile computing solutions of today are still too complex, too obtrusive, and too demanding on the user to be seamlessly integrated into complex work-processes, their usability in industrial scenarios and their acceptance by the workers is still limited.

The European project wearIT@work takes the next important step of innovation (http://www.fit.fraunhofer.de/projekte/wearit/): it will develop a new paradigm for wearable mobile computing that supports complex tasks with a minimum of human-machine interaction and thereby enables mobile professionals to keep their attention focused on the interaction with the work environment. A new software and hardware platform will be created in which professionals can be mobile and at the same time fully integrated into the surrounding IT infrastructure. One of the main features of wearable applications is the immediate mobile delivery of information. In fact, "*at any place and any time*" does not sufficiently meet the requirement for wearable computing and this condition has to be strengthened to "*at the right place, the right time and in the right form*".

The observation of a plant manager's behavior in his car production area gives a good example for accurate information delivery: Due to many meetings and inspections on-site the plant manager moves in or outside the plant. His daily work includes regular checks of the production state, in order to invoke countermeasures at an early stage in case of failures and fallouts of the assembly line. When a system failure occurs causing loss of car production, the plant's manager has to be informed immediately. It is not just the fact that is important at that moment, but the following information as well:

- faulty station
- person responsible for that station (supplier contact)
- time available until line comes to halt
- estimated repair time (whenever available)
- estimated cost if repair is not completed in time (cars lost that shift).

The type of information varies between different content types like text, audio or video and consists of usage rate diagrams of every assembly station and production data per day, as well as compilations on the gain and loss of projected car production. Production Control & Monitoring Systems as well as ERP systems (Enterprise Resource Planning, a business management system that integrates all facets of the business, including planning, manufacturing, sales, and marketing) are the source for this information, which has to be filtered and prepared taking contextual parameters like location, daytime and availability of assembly workers into account. The hardware the plant manager uses for this kind of information provision is not necessarily unobtrusive. It may be something like a PDA device or a smart phone and has to show the information in an understandable manner.

Today mobile devices deliver information that is adapted to the current location or network bandwidth of the user's device. Commercial solutions push the mobile market towards providing content and services on a variety of devices and visual displays. Nevertheless in today's location based services the adaptation to location is only rarely combined with a personalization to the individual user [1]. From our point of view it is not enough to supply content or services that consider single environmental or user characteristics but to identify approaches for the integration and interpretation of different sensoring components for modeling the user context more appropriately. The combination of sensor data for context and its combination with user modeling appear to be one of the key challenges for personalized information systems of the future. In this sense delivering information to the point has mainly two facets: *Personalization* and *contextualization* as its extension.

In the paper we will introduce a toolkit for modeling context and creating contextualizing applications. We will display our developments on the basis of the implementation of a mobile museums guide and describe the integration of content and context using the "Mobile Collector" application and the playback of the annotated content with a content player.

2 A Context-Management Framework

Many projects at the Fraunhofer Institute for Applied Information Technology made use of user modeling components and personalization engines in order to make systems adaptable and adaptive to the user's behavior. Many of these components are specialized applications, tailored to one special domain or environment and rarely reusable in other or subsequent applications. Common problems that emerge during the development and the reuse of components are the strong dependency on the domain, no open and standardized interfaces, no uniform representation of user profiles and models, and the use of different and distributed data sources and highly dynamic domains where variable properties change in short time intervals. Generalizing from our experiences in working in different application domains and from the variety of technical implementations lead us to the following definition of Context-Management:

Context-Management allows building, integrating and administrating context-aware behavior. Context-Management considers the definition of relevant context-parameters, the link between these parameters and information sources, their utilization for and the definition of the targeted adaptive behavior.

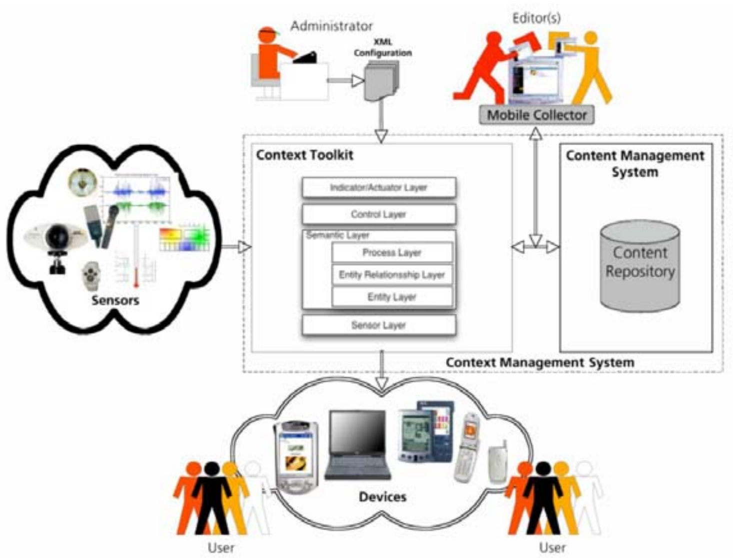

Figure 1 The Components of a Context-Management Framework

Figure 1 illustrates the components a Context-Management System consists of and the roles of active users. In the centre, an administrator builds and maintains the desired application for the end-users. From our previous work ([2], [3], [4]) developing user-adaptive systems, we propose to have four categories of *sensoring*, *modeling*, *controlling* and *actuating* components for the realization of context-aware applications. Furthermore, the contents to be delivered are stored in a Content-Management System, wherefore an editor administrates contents and creates the links to the specific context they relate to.

Resent research in smart sensor-networks enables for placing huge numbers of intelligent senor-components (“smart dust”) in the environment. Smart sensors are equipped with small processors that enable for intelligent information acquisition [5].

In self-organizing networks, such as Intel's iMote approach [6], sensor technologies builds ad-hoc sensor-networks and deliver requested information on demand. At the end of the processing line, a server connected to the internet receives all data, processes the information and delivers inferred knowledge towards desired applications. As for sensor- and actor-networks, networks of specialized software-components process small tasks like delivering one information snippet or displaying data on a particular device. The sensory and the actuating function is therefore distributed over different devices of the mobile user [7] or embedded in her environment, e.g. among others the light sensor of a PDA, the infrared sensor of an automatic door and the GPS-sensor of the car are contributing to the sensor-package regardless to their physical location and environment. In turn, each device potentially hosts different types of components, e.g. a PDA independently hosts sensors for light-conditions, background-noise and pen-input as well as display components and actuators for video-streaming or adjustment of the display-brightness.

2.1 Sensor Package

As information collectors, sensory mechanisms perform an observation of the user behavior and interaction with the system, whereby the system's sensitivity, speed, and accuracy will depend on the technology of the sensing infrastructure. Smart sensors enable for intelligent information acquisition of the environment including the time and location, in which the user interaction took place. Ad-hoc sensor-networks will deliver the requested information on demand like the technology applied for the current interaction, i.e. devices and infrastructure involved. Each mobile application relies on a network of sensors placed within the physical environment and watching indicators for changing situations. A server as a information sink further processes all incoming sensor data.

2.2 Modeling Package

In our understanding a user may have a context, and in turn a context may enclose a user, too. The context model captures the current situation the users act in, their preferences, interests, their social dependencies, their physical and technical environment, and so forth. Overall, there exist many different views on what dimensions such a model has to cover, e.g. *Identity*, *location*, *time,* and *environment* [8] or *user context*, *computing context* and *physical context* in [9] and types of contexts, like *primary and secondary contexts* [10] or *static and dynamic contexts* [11]. To pay attention to the variety of approaches and not limit its potentials we choose attribute-value pairs as a flexible context representation in our approach. The semantic components receive and enrich the sensor data, and assign values to corresponding entity attributes. It supplies the controlling components with an accurate image of the current interaction situation represented by attribute-value pairs. In addition, context change events are sent out in order to inform other components.

2.3 Controlling Components

Based on the available knowledge about the user's context model controlling components generates sequences of commands for actuator components, in order to control the behavior of the application [12]. The commands assembled by the control layer may vary in their level of abstraction: simple *commands* and more complex *strategies*. The selected action or strategy can be seen as the link between these two questions: What information is taken into account for adaptation and which part or functionality of the user interface is adapted and how? Additionally, a direct communication link between the those components and the domain enables answering requests from external systems or applications, and the realization of shared initiative and shared control approaches [13, 14, 15], which have been proposed and used for end user development in the last ten years for adaptive information and learning systems [16].

2.4 Actuators-Package

The actuators handle the connection back to the domain by mapping the decisions taken by the control into real world actions. They are specialized software-components that process the delivering information snippets or displaying data on a particular device. A rendering-engine implements domain-dependent methods that directly change variable parameters of the domain. A mobile device like a PDA may host several display components and actuators for video-streaming and adjustment of the screen backlighting. Depending on the level of integration with the domain, these methods may be part of the target application or the commands are to be transformed into appropriate actions. As a feedback, messages indicating the success or failure of actions are sent back.

3 Developing Context-Aware Systems with Context-Management

The framework described in the previous section forms the basis for the combination of content and context, in order to build contextualized applications. As case study we will use a mobile museums guide and illustrate tools facilitating three main steps in building context-aware systems: definition of an appropriate context model using the *Context-Toolkit*, linking context information with content using the *Mobile Collector* and reproducing contextualized content using the *Content Player*.

3.1 Case Study

The LISTEN project (http://listen.imk.fraunhofer.de) conducted by the Fraunhofer Institute for Media Communication is an attempt to make use of inherent everyday integration of the aural and visual perception [17]. In October 2003 this system was applied for the visitors of the August Macke art exhibition at the Kunstmuseum in

Bonn. As a preparatory work we set up a museums guide with which the visitors of the museum obtain personalized information about exhibits displayed on a Personal Digital Assistant (PDA) they carry around. During the preparation of the exhibition hall, the curator of the museum used the Mobile Collector to annotate the exhibits with adequate information. In front of the paintings the curator selected the content blocks from the database that were to be associated with the exhibit and adjusted the context values appropriately in which these content blocks will be retrieved during operation. For the museums guide the types of content blocks were mainly images, text and sound (spoken text) and the context-attributes were the visitors' position, the time of the day and the weekday. During the operation of the system the Content Player running on each device filters and presents this content while taking into account the visitor's current context. This initial system was designed for eight users.

3.2 Modeling Contexts

The Context-Toolkit developed by the Fraunhofer Institute provides ready-to-use software components that comply with the packages presented in the previous Section, as well as an appropriate authoring tool. Figure 2 shows a screenshot of the *Modeling*-panel, with which an administrator has designed the context model used in the application. This panel allows for the definition of domain entities and several types of context. Furthermore, it supports the allocation of sensors (information sources) to attributes (information interpreters).

By creating a cross-linked network between sensors and attributes, attributes are connected with any number of sensors and in turn sensors deliver information to one or more attributes. Internally, all attributes listen to events sent by sensors they are registered to as listeners. In addition, specific attributes may have no sensor as an information source; these attributes are containers for values determined by the control layer. Since the Context-Toolkit was designed as a modeling tool, its current implementation does not support automatic detection or recognition of sensors.

The left hand side of Figure 2 depicts the dynamic and the static context we added to the list of context types, as well as the attributes they consist of. The static part of the context model contains some meta-information about the contents, like their identification and the category they belong to. Additionally, the allocation of sensors to attributes is shown, like for example the attribute *daytime* depends on the *TimeSensor*.

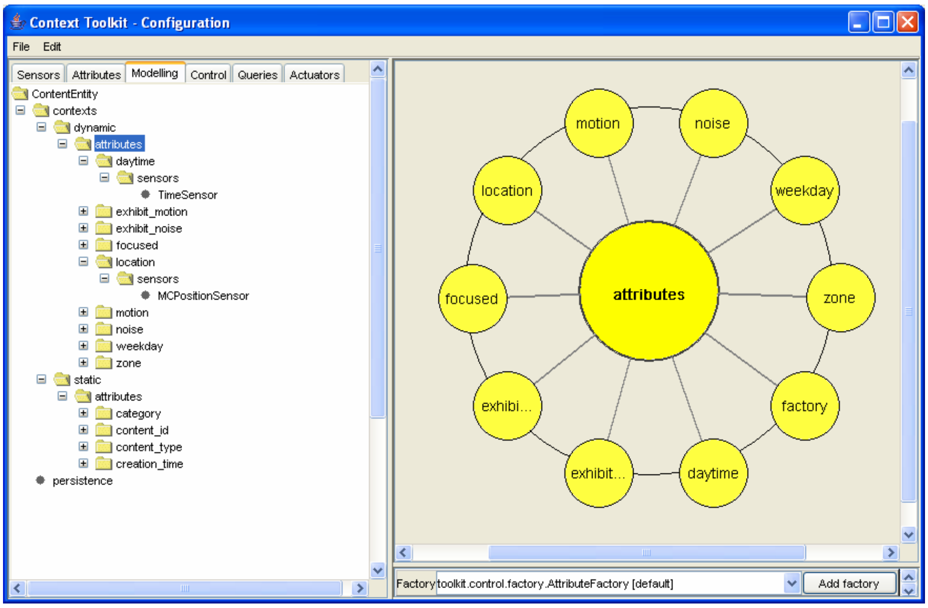

Figure 2 The Context-Modeling Panel

3.3 Annotating Content with Context Information

The annotation of content with context data is a special task which can most effectively be done on the move or directly in the context of use. Mainly our approach supports users with a tool for "recording" or capturing context data together with content from a Content-Management System. The Mobile Collector offers content providers a very efficient tool for the production of contextualized content.

Figure 3 illustrates the Mobile Collector running on a Tablet PC. The right hand side shows the web front-end of the Content-Management System. This screen provides the author with functionality for adding, removing, searching and browsing content such as images, sounds, videos or even entire HTML-pages. The left hand side of the Figure depicts the current context of the device in the lower panel and the current sensor values in the upper panel. If there are any changes to the values of the sensors or context-attributes, both panels are immediately updated. Since the left panel is a browser plug-in, it does not affect the user while browsing the net. It reflects the context model defined with the Context-Toolkit and allows the author to capture the current context (i.e. to freeze its values), to edit context-attribute values and to (un)select attributes that are considered (ir)relevant in this specific situation.

If the correct content is selected and the context is adjusted appropriately, the author can easily create the link between those two by just clicking the snapshot-button and store this link to the persistent memory of the Mobile Collector. By using a context-snapshot approach authors of content or annotators can easily capture context meta-data for content. One context-snapshot consists of current values for each context-attribute in the specific moment the snapshot-button has been pressed. This value vector stored with each content block is the basis for a filtering process that retrieves content in a specific context later on. The retrieval procedure compares the stored context-snapshot with the values of the context the user currently is in and returns the content block associated with the best match. Since the Mobile Collector is a tool for collecting context snapshots and linking them to appropriate contents, it is not used for administrative purposes. Breaking up these links between contexts and contents, i.e. removing of the context annotation for a specific content, requires further treatment using an administrator tool at a desktop PC.

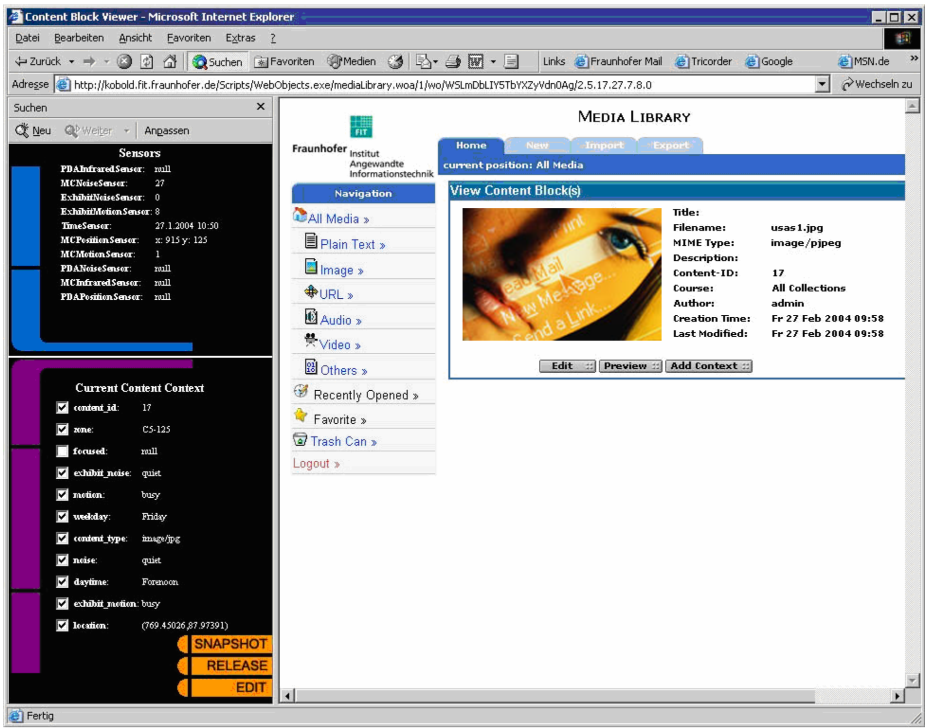

Figure 3 A screenshot of the Mobile Collector displayed on a Tablet PC

3.4 The Content Player

The Content Player is an adapted browser that runs on a mobile device like a Personal Digital Assistant (PDA). Almost like the Mobile Collector, the sensors connected to the mobile device are readout and their values are sent to the server. The server interprets the sensor values and determines the behavior of the targeted device. In this case the determination of the behavior of the targeted device is the selection of contents from the content-management system.

Figure 4 shows the playback of the annotated content blocks with the content player running on a PDA. The content selection is carried out depending on the context the device currently is in. The server sends the contents suiting the current situation to the content player, where the browser refreshes the displayed page with the new contents. The Context-Toolkit supports the determination of the appropriate behavior and the content playback. The contents which were annotated with the help of the Mobile Collector are retrieved for the content player in similar situation. The definition of rules by the Context-Toolkit enables the system is able to control the retrieval.

Figure 4 A picture of a mobile device running a content player

4. Conclusions and Future Work

The presented work gives an overview of current developments towards a context management system in Fraunhofer for Applied Information Technology. Based on the identification of the main functions different packages of components have been introduced and prototypically implemented to support sensor data management, context abstraction, and the control of actuator output. We have described implementations of sample applications for intelligent information distribution and selection and the collection and connection of context data with contents.

For our future work we plan to build a variety of different applications and tools based on the described infrastructure to evaluate the approach. The experiences we gained with the museums domain as a case study will influence on the production application domain in the wearIT@work project. This application domain requires a strong integration of the detection of context and the provision of content. Since the end-users of the overall system will be inexperienced developers, the facilitated production of contextualized content is prerequisite. Especially for semantic enrichment of raw sensor-data we experiment with nested filter components to extract more significant information. From our experience with different control component we see that even a simple set of rules defined by end users can have very complex output.

The demand for context-sensitive functionalities constitutes a crucial challenge for system developers as well as for product managers, application designers and system integrators. They have to handle the system heterogeneity, starting from the hardware and software protocols, including the integration within various mobile and wireless environments, considering seamless integration of services from multiple providers, and release and validation of new protocols. At the same time, managers, developers and system integrators have to react to shortened delivery time for a competitive and dynamic market. In order to integrate the development of the basic components by software engineers with the tailoring of components to build the final application logic, we deliver the framework completely configurable via XML and the user front-end for non-experts in computer science.

References

1. Jameson, A.: User-adaptive and other smart adaptive systems: Possible synergies. Proceedings of the First EUNITE Symposium, Tenerife, Spain (2001) 13–14
2. Zimmermann, A., Lorenz, A., Specht, M.: User modeling in adaptive audio-augmented museum environments. In: Brusilowsky, P., Corbett, A., and de Rosis, F. (eds.): 9th international conference on user modeling, Johnstown, USA, Springer-Verlag (2003) 403–407
3. Zimmermann, A., Lorenz, A.: Listen: Contextualized presentation for audio-augmented environments. 11th Workshop on Adaptivity and User modelling in Interactive Systems, Karlsruhe, Germany (2003) 351–357
4. Lorenz, A.: Towards a new role of agent technology in user modelling. 12th Workshop on Adaptivity and User Modeling in Interactive Systems, Karlsruhe, Germany (2003)
5. Satyanarayanan, M.: Of Smart Dust and Brilliant Rocks. Pervasive Computing 2 (2003) 2-4
6. Culler, D.E., Mulder, H.. Smart Sensors to Network the World. Scientific American, 290 (2004) 52-59
7. Hodes, T.D., Katz, R.H.: Composeable Ad Hoc Location-Based Services for Heterogeneous Mobile Clients. ACM Wireless Networks, 5 (1999) 411–427
8. Gross, T., Specht, M.: Awareness in Context-Aware Information Systems. Mensch&Computer - 1. Fachübergreifende Konferenz, Bad Honnef, Germany, Teubner (2001) 173–182
9. Schilit, B.N., Adams, N.I., Want, R.: Context-Aware Computing Applications. Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, (1994) 85–90
10. Dey, A.K., Abowd, G.D.: Towards a better understanding of context and context-awareness, Technical Report GIT-GVU-99-22, College of Computing, Georgia Institute of Technology (1999)
11. Zimmermann, A., Lorenz, A., Specht, M.: Reasoning From Contexts. 10th Workshop on Adaptivity and User modelling in Interactive Systems, Hannover, Germany (2002) 114–120
12. Liu, J., Chu, M., Liu, J., Reich, J., Zhao, F. State-Centric Programming for Sensor-Actuator Network Systems. Pervasive Computing 2 (2003) 50-62
13. Oppermann, R.: Adaptive User support. Hillsdale Lawrence Erlbaum Associates (1994)
14. Oppermann, R., Thomas, C.G.: Supporting Learning as an Iterative Process. European Conference on Artificial Intelligence in Education, Lisboa, Potugal (1996)
15. Oppermann, R. and Specht, M.: A Context-Sensitive Nomadic Exhibition Guide. Second Symposium on Handheld and Ubiquituous Computing, Bristol, UK, Springer-Verlag (2000)
16. Wulf, V.: Exploration environments: supporting users to learn groupware functions. Interacting with Computers 13 (2000) 265–299
17. Eckel, G.: LISTEN Augmenting Everyday Environments with Interactive Soundscapes. I3 Spring Days Workshop “Moving between the physical and the digital: exploring and developing new forms of mixed reality user experience”, Porto, Portugal (2001)