The Smart Box Application Model

Matthias Lampe and Christian Flörkemeier[1]

Abstract

This paper proposes the usage of a software framework, based on the Smart Box application model, for the development of applications or appliances of the Smart Box application domain. This class of ubiquitous computing applications can be described as automatic content monitoring applications using Auto-ID technologies such as Radio Frequency Identification (RFID). The Smart Box application model is based on the concepts of location and object to represent a kind of a box and the items and persons that are related to the box. States defined through conditions abstract from the content of the box and limit notification of the application to exceptional cases. Implementing the model and framework we could show that the model is adequate to describe the Smart Box application domain and that the framework decreases the development effort of Smart Box applications.

1. Introduction

In daily life, health care or industrial environments, we are often faced with the need to check the content of some kind of container or box to take appropriate action in case of an "abnormality". Examples include making sure that a toolbox or a first aid kit is complete or checking the contents of a medicine cabinet. Often these checking tasks are experienced as cumbersome and annoying, since one is required to check many times more frequently than any abnormalities are actually detected. In industry these labor-intensive tasks tend to be costly and error-prone.

Coming from Mark Weiser's vision that embedded technology would calm our lives by removing the annoyances [18], several ubiquitous computing appliances and applications have been developed (see section 2) that can be classified as Smart Box applications [4]. Smart Box applications have in common that they automatically and unobtrusively perform the content monitoring task using Auto-ID technologies such as Radio Frequency Identification (RFID) [3].

Existing applications were developed with no reuse in mind duplicating major parts of the same development process over and over again. Since the Smart Box applications have many common features, we propose the usage of a software framework based on the Smart Box application model presented in this paper. The framework can provide a reusable design, architecture and predefined software components for the Smart Box application domain to reduce the development, testing, and maintenance costs of future Smart Box applications.

[1] Institute for Pervasive Computing, ETH Zurich, 8092 Zurich, Switzerland, {lampe,floerkem}@inf.ethz.ch

The remainder of this paper is structured as follows. Section 2 presents the Smart Box application domain and the results of the domain analysis. Section 3 gives a detailed description of the application model. A software framework architecture based on the model is briefly discussed in Section 4. Section 5 presents related work and section 6 draws the conclusion.

2. Smart Box Application Domain

An analysis of the application domain was performed to be able to design an application model and framework that keeps the balance between flexibility to cover a wide range of possible applications and complexity. The following applications, demonstrators and application ideas, were analyzed to better understand the Smart Box application domain:

- *Web Luggage* [2]. Observes everyday items placed in the luggage by the owner and the content can be visualized by an external display.
- *Magic Wardrobe* [15]. Knows about clothes placed inside and suggest clothes that fit together the best based on entries in the personal calendar (e.g. business meetings).
- *Smart Shelf* [12, 13]. Products on a shelf are monitored to enhance replenishment and identify misplaced products.
- *Magic Medicine Cabinet* [4, 14]. Monitors the medicine that is placed in or taken out by a patient or pharmacist and displays information about prescriptions, incompatibilities or recalled medicines.
- *Smart Surgical Kit* [4]. The usage of bandages and swabs during a surgical operation is monitored and the usage status is displayed to avoid leaving any operation tools in the patient.
- *Smart Toolbox* [4, 7]. Monitors the tools in a toolbox and the mechanic. Displays the content of the toolbox and tool usages and warns if tools are missing or placed in the wrong toolbox.
- *Smart Fridge* [8]. Keeps track of its content, suggests recipes based on content and warns if goods reach expiry date or contain any allergic ingredient.

The analysis yielded the following commonalities:

- The application settings are clearly defined. All objects are either in some kind of a box or are outside the box. If they are inside, they are detected by the system and actions associated with certain system states are triggered. If they are outside, they are not detected and some assumptions are made. It is obvious that the absence of items can also trigger actions.
- The behavior of the system often depends on the user that is currently located near by the system.
- The system is able to sense environmental conditions (e.g. temperature).
- A set of specific system states or characteristics are of explicit interest.

- The system provides additional services that often depend on the currently identified users or objects.

The applications differ in the following points:

- Structure of the box or container (i.e. single box or partitions).
- The types of sensors and objects and their properties and the roles of persons.
- The additional services (e.g. keeping a history of the content, or making the content available for remote access).

3. Smart Box Model

The Smart Box application model is the core of the framework. It represents the key observations of the real world with respect to the Smart Box applications and reflects the commonalities and differences of the application domain. Figure 1 illustrates an instance of the model using the example of a smart medicine shelf in a hospital.

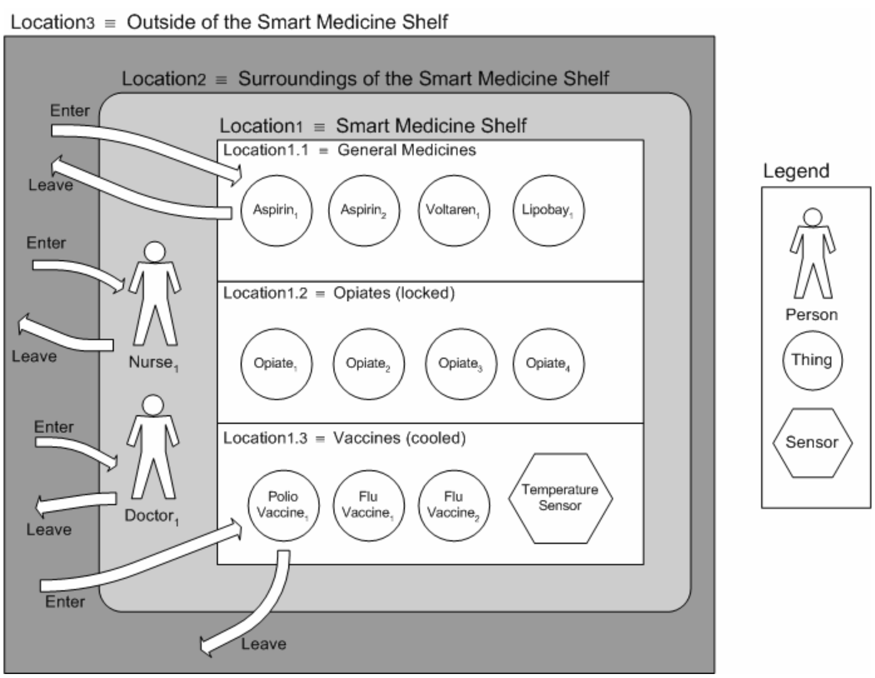

Figure 1 Instance of the application model for the smart medicine shelf

The Smart Box application model is a specialization of the world model of the ubiquitous computing infrastructure presented in [1] for Smart Box applications and uses the two important concepts: objects and locations, where objects can be persons and things.

Persons are human beings that can be identified by the system and that fulfill a certain role in the application. Things are objects of the real world, which can be identified by the system, and that correspond to the specific application domain. Things can have instance and class properties (e.g. name, descriptions, expiration date). A location is a closely bordered area that is clearly delimited from any other location and can contain objects. As shown in Figure 1, there are only three major locations in a Smart Box Model: Location1, Location2, and Location3. The dark gray shaded area labeled Location3 represents the area where an object is out of sight of the system and cannot be recognized (outside the box). This area has a boundary, which delimits it from Location2. Location1 is the actual opposite of Location3. It represents the inside of the box. Objects that are within Location1 are detected and monitored by the system. Location1 can be partitioned in sub boxes. Every sub box is able to detect objects located inside the specific sub box. Location2 softens the boundaries between Location1 and Location3 in a conceptual manner. It acts as the surroundings of the box, where objects can be detected and monitored by the system.

In the Smart Medicine Shelf example, the “box” consists of three compartments: one for general medicines accessible to nurses and doctors, one for opiates accessible only to doctors, and one for vaccines, which is cooled and accessible to nurses and doctors.

In the Smart Box model, each box can compare its current content or sensor readings with a desired configuration and trigger predefined actions, if a specific state has been reached. This allows abstracting from the content of the box and sensor readings and limits notification of the application to exceptional cases. If a set of conditions defined on the content is satisfied, the state in which the system stays usually differs from the one in the case of an unsatisfied set of conditions. A desired configuration can be achieved if a desired state has been reached. To reach a specific state, a set of conditions on the content of the box has to be satisfied. Conditions can involve presence or absence of a set, classes, or number of objects of a box and physical properties sensed in a box. It is possible to logically combine different conditions to describe a system state. Often used states are:

- Complete. All expected objects or classes of objects are inside the box.
- Incomplete. At least one of the expected objects or classes of objects is missing.
- Correct. Only expected objects or classes of objects are inside the box or the value of a physical property is within a predefined range.
- Incorrect. At least one of the expected objects or classes of objects is not expected to be within the box or the value of a physical property is not within a predefined range.

In the smart medicine shelf example, a correct state of location 1.3 would be: the temperature is between 11 and 13 degree Celsius, all objects are of class vaccine, and no object has reached its expiration date. Access to the opiates could be triggered by the condition that a person with the role doctor has to be present in the surroundings. If the number of a class of object (e.g. Aspirin) drops below a certain value a notification could be sent to an automatic replenishment or reorder system.

4. The Smart Box Software Framework

In order to demonstrate the flexibility of the Smart Box application model and as a proof-of-concept, we implemented a software framework based on the model as the framework core. The architecture of the framework consists of the following major components:

- The *Auto-ID and sensor* infrastructure makes the connection to the real world and automatically senses physical properties and identifies persons and things at certain locations. As suggested by [16], the framework uses RFID technology to bridge the physical and virtual world and sense the contents of a box.
- The *transformation and interpretation* component that interprets, transforms and corrects the data from the Auto-ID infrastructure according to deterministic or probabilistic rules in order to reduce errors and uncertainties. It also maps the resulting data into the model.
- The *model* as described in the previous section.
- The *application logic* provides the model with the conditions and states the application is interested in. The application is notified by the model about state changes and triggers the preferred action such as displaying a warning on a user interface (e.g. an indicator such as a green and red LED) or the automatic replenishment of objects (e.g. important vaccine).

5. Related Work

Ubicomp system infrastructures (e.g. GAIA [9], Aura [10], or Nexus [5]) and context research (e.g. Context Toolkit [11], or [6]) provide general models and application abstractions that focus on concepts such as persons and devices, user mobility, intelligent environments, and the processing and modeling of context information. The goal of these very general and abstract approaches is to support the construction of a great variety of Ubicomp applications. However, due to the generality of these approaches, the effort to build applications is still high.

Our approach focuses on a particular class of Ubicomp applications (i.e. Smart Box applications), and provides a restricted but simple model and framework in order to minimize the effort to build these kind of applications and maximize the reuse.

6. Conclusion

Using the software framework built on the Smart Box model, we were able to implement the applications of the examined application domain and the presented smart medicine shelf with few additions to the application logic, mainly user interface components. This shows that the model is adequate to describe the Smart Box application domain and that the framework decreases the development effort of Smart Box applications. However, more Smart Box applications have to be implemented using the proposed model in order to find the right balance between a too simple and a too complex model.

Since the objective of our approach was to construct a model and framework to facilitate the implementation of Smart Box applications, it is not well suited for more general location based applications. Industrial and business applications (such as [4, 15, 12, 14, 7]) which provide a set of business rules that can be mapped to the conditions and states of the model, will benefit the most from the Smart Box framework.

Further research will be done to improve the software framework and model: (a) Finding the right abstractions to facilitate the generation of rules for the state conditions in order to support applications with a large number of objects, and (b) examining the application domain with an emphasis on the user interfaces to provide black box user interface components in order to further minimize the development effort.

7. References

[1] T. Dübendorfer, F. Mattern, K. Römer, and T. Schoch, "Smart Identification Frameworks for Ubiquitous Computing Applications," presented at PerCom 2003, March 2003.

[2] A. Ferscha, S. Vogl, and W. Beer, "Ubiquitous Context Sensing in Wireless Environments," presented at Workshop on Distributed and Parallel Systems, Austria, 2002.

[3] K. Finkenzeller, "RFID Handbook," John Wiley and Sons Ltd, 1999.

[4] C. Floerkemeier, M. Lampe, and T. Schoch, "The Smart Box Concept for Ubiquitous Computing Environments," presented at Smart Objects Conference, Grenoble, 2003.

[5] D. Fritsch, D. Klinec, S. Volz , "NEXUS – Positioning and Data Management Concepts for Location Aware Applications," in: Proc. of International Symposium on Telegeoprocessing, Nice-Sophia-Antipolis, France, 2000, S. 171-184.

[6] K. Henricksen, J. Indulska, A. Rakotonirainy, "Modeling Context Information in Pervasive Computing Systems, " in: Proceedings of Pervasive 2002, Zurich, Switzerland, 2002, pp 167-180.

[7] M. Lampe and M. Strassner, "The Potential of RFID for Moveable Asset Management," presented at the Workshop on Ubiquitous Commerce at Ubicomp 2003, Seattle, 2003.

[8] M. Levinson, "All-in-One Appliance - THE REFRIGERATOR," CIO Magazine, February 2003.

[9] M. Roman, C. Hess, R. Cerqueira, A. Ranganathan, R. Campbell and K. Nahrstedt, "Gaia: A Middleware Infrastructure to Enable Active Spaces, " IEEE Pervasive Computing, vol. 1(4), 2002.

[10] J.P. Sousa and D. Garlan, "Aura: An Architectural Framework for user Mobility in Ubiquitous Computing Environments, " in: Proceedings of IEEE/IFIP Conference on Software Architecture 2002.

[11] D. Salber, A.K. Dey, G.D. Abowd, "The Context Toolkit: Aiding the Development of Context-Enabled Applications," in: Proceedings of CHI'99. 1999, pp 434-441.

[12] SAP Corporate Research, "Shelves that Call for Supplies," 2002, http://www.sapinfo.net/public/de/printout.php4/article/Article 49043df892f123d88/en

[13] TecO, "Ubicomp Research & Projects at TecO: SmartShelf," 2003, www.teco.uni-karlsruhe.de/research/ubicomp/smartshelf/

[14] D. Wan, "Magic Medicine Cabinet: A Situated Portal for Consumer Healthcare," presented at International Symposium on Handheld and Ubiquitous Computing, Karlsruhe, 1999.

[15] D. Wan, "MAGIC WARDROBE: Situated Shopping from Your Own Bedroom," presented at Proceedings of the Second International Symposium on Handheld and Ubiquitous Computing, Bristol, UK, 2000.

[16] R. Want, K. O. Fishkin, A. Gujar, and B. L. Harrison, "Bridging physical and virtual worlds with electronic tags," presented at ACM SIGCHI, 1999.

[18] M. Weiser, "The computer of the 21st century," Scientific American, pp. 94- 100, 1991.