Benefits of Context Models in Smart Environments

Daniela Nicklas, Susanne Bürklen, Tobias Drosdol, Nicola Hönle
Universität Stuttgart, Institute of Parallel and Distributed Systems
Universitätsstraße 38, 70569 Stuttgart, Germany
<firstname>.<lastname>@informatik.uni-stuttgart.de

Abstract: Ongoing technologic advances drive the emergence of *smart items*, everyday objects with embedded microcomputers and communication capabilities. In a *smart environment* a multitude of such smart items exist to assist its users. In this paper, we will show how smart environments can benefit from the concepts of the Nexus platform, an open pervasive computing system that supports various kinds of context-aware applications by providing a federated, potentially global context model.

1 Introduction

The vision of *ubiquitous computing* has drawn increasing attention of researchers in the past years. Ongoing technologic advances drive the emergence of *smart items*, everyday objects with embedded microcomputers and communication capabilities that can sense their current state or changes in their environment by integrated sensor systems. Such devices can range from smart household appliances to smart cups or picture frames. Smart items cooperating with each other can form a *smart environment*, in order to assist users in their daily work and simplify their life. Examples are smart homes that help elderly people [KOA+99] or smart offices that provide new methods for interacting with devices and information [SGH+99] [RC00].

Yet, the realization of such an environment is a difficult task. It typically incorporates a multitude of very heterogeneous devices that need to be set up and configured individually. Cooperation between these devices furthermore requires interoperability of data formats and exchanged communication messages. In the following, we will therefore show how smart environments can benefit from the concepts of the Nexus platform and which use cases can be easily developed using its support.

In its beginning, the Nexus project[1] [Nexus] focused on location-aware applications that support mobile users and take into account their spatial context, i.e. their position. Such applications typically run on a mobile client that is equipped with location sensors and is capable of wireless communication. The spatial range of these applications can be quite large, from campus up to city or country level. Examples are tourist guides [LKAA96] [CDM+00] or mobile navigation applications [BKW02] [HHT+01].

Unlike many other approaches in that area, the Nexus project focuses not on a specific application, but on platform support for all kind of location-aware applications. The main idea is to provide a global-scale context model that is federated over autonomous local context models [NM04].

In the following section, we briefly introduce the Nexus platform. Then we show how smart environments can benefit from context models that can be managed by the platform.

[1] The Nexus project was funded 1999-2002 by the German Research Association (DFG) under grant 200989 and is now continued as Center of Excellence (SFB) 627.

2 The Nexus Platform

In this section, we briefly describe our present context managing approach—the Nexus platform [NM04].

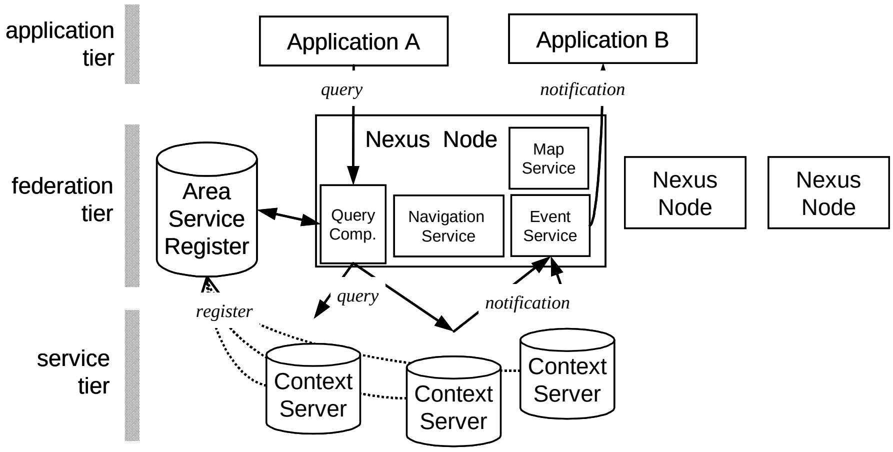

Figure 1: Architecture of the Nexus platform

The Nexus platform (Figure 1) is an open pervasive computing system that supports various kinds of context-aware applications by providing a federated, potentially global context model. To achieve this goal, local data providers (so called context servers) offer their context data in a standardized yet extensible way: they implement a given interface (the Augmented World Query Language, AWQL) and present their data according to a given model (the Augmented World Modeling Language, AWML). There can be many different implementations of a context server [GBH+05]. For providing large scale static models, we used a spatially enhanced database, but even small scaled sensor platforms or sensors itself can be used as context servers.

The model is object-based: there are data objects that have attributes. Up to now we have defined a basic set of object types, of which we think that they are essential for context-aware applications. This basic set includes object types for real objects as well as for virtual objects. We can model map data like streets and buildings, ground plans of buildings, and sensor objects with sensor measurement values. Objects can be given in different levels of detail.

All objects of a data provider lie within the boundaries (service area) of a so called Augmented Area. Augmented Areas can overlap in space and also in the objects they provide. Therefore, the same real world object can be part of different Augmented Areas and can be modeled using different data objects. We call them multiple representations. Context servers register their Augmented Areas and contained object types in the Area Service Register.

A federation node mediates between applications and context servers. It has the same interface as a context server, but does not store models. In addition to the query functionality, every federation node supports value-added services. They use the federated context model to implement advanced services having their own interfaces. The map service, for example, renders maps based on a selected area. With GeoCast, you can address a message to a geographic area, sending it to every mobile and stationary host that is currently located in this region.

3 Applying Context Models to Smart Rooms

In this section, we introduce some use cases for realizing smart environments by using context models. All these use cases could also be implemented and maintained without using a context model. However, the use of context models, as provided by the Nexus platform, offers appealing benefits. The context model acts as a layer of abstraction between applications and the installed hardware, e.g., sensors. Thus, all interaction between different components is based on a common interface, easing the deployment of new items and allowing applications to be used likewise in any smart room. In the following, this is illustrated by means of different sample use cases:

- *Service discovery:* Discovering available services in a smart environment will typically rely on a service registry, which allows publishing and querying installed services, their offered features and the protocols they support. Instead of using a local registry for every smart room, this task could be carried out by a common context model as well. Defining the services and all their properties within such a model additionally enables applications to select an appropriate service based on their current context (e.g., selecting services with a certain spatial scope). For example, a context-aware search for a presentation service can always provide the output device that is currently located closest to the user.
- *Object Positioning:* Automated positioning of mobile entities is another important foundation for many features in a smart environment. This can be achieved by a wide range of different approaches. For example, RFID-readers can determine tagged items in their vicinity, installed cameras can identify moving objects through scene analysis, and a pressure-sensitive floor can locate users. All such techniques typically require matching the observed data with substantial reference values. Using a context model, such reference data must be maintained exactly once per object and can be shared across different smart environments. Furthermore, the accuracy of positioning can often be enhanced by utilizing additional context information. E.g., if an object is known to be in the room, then its position is bordered by the wall.
- *Object Identification:* In many situations, visitors of a smart room can be assisted by providing detailed information about certain objects (like a painting on the wall). Identifying the object of investigation should thereby be as easy as pointing to it. This could, for instance, be realized by detecting the target of a laser pointer with installed cameras. The selected entity at that position can then be determined by querying a detailed context model, which contains the locations of all available objects. This model can also provide detailed information about the object.
- *Integration of new hardware:* Without a common model, applications and services of a smart environment have to be tailored to the technical details of all installed hardware. This requires adaptation of the software whenever hardware is added or replaced. Using a context model, this integration can be done more easily in a context aware manner, since these details only have to be specified once in the context model.
- *Remote control of entities:* A lot of scenarios require remote access to the dynamic information and controls of a smart room. For example, during a teleconference one participant might wish to control the smart board of the distant party to visualize his statements. Using a common context model, no customized remote control for this particular item is required. Instead, it can be accessed directly through the Nexus platform.
- *Context events:* Another possibility of interacting with the Nexus platform is the asynchronous notification about the occurrence of certain event conditions. E.g., the user can register to be notified when the temperature in a room exceeds a certain value, or two persons meet. By integrating smart environments into the Nexus platform, this mechanism can be shared throughout all different environments.

4 Conclusion

The presented use cases show clearly, how the realization of smart environments can benefit from utilizing the Nexus platform. A shared context model provides a valuable abstraction of a specialized smart item. It thus forms the basis for interoperability between all kinds of smart items and allows for smooth interaction with them, superseding their manual adaptation and

configuration. This eases the deployment and modification of a smart environment as well as the integration of new items into it. Additionally, mobile devices can easily be used in very different smart environments, without prior tailoring of their applications. These applications can utilize the shared context model to gather information about their current environment and to interact with available services in a well defined way.
At the same time, this approach also offers the possibility to integrate a digital model of the smart environment into the Nexus platform without additional effort. This allows for accessing its dynamic information for other location-aware applications, that can seamlessly use the smart room context data together with other, spatially overlapping context models (e.g. from the building or the surrounding environment).
Within our Nexus-Lab at the University of Stuttgart, we are currently working on a prototypical realization of a smart room based on the Nexus platform. Amongst others, this will feature an Everywhere Display (a steerable video projector), a 3D-sound system, a smart board, several pivoted cameras, IR-beacons, RFID-readers, and a multitude of sensor systems. We are looking forward to provide an insight into the practical experiences gained throughout this project in the future.

References

[BKW02] J. Baus, A. Krüger and W. Wahlster: A resource adaptive mobile navigation system. Proceedings of the International Conference on Intelligent User Interfaces, San Francisco, USA, 2002

[CDM+00] K. Cheverst, N. Davies, K. Mitchell, A. Friday and C. Efstratiou: Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences. Proceedings of CHI 2000, Netherlands, 2000

[GBH+05] M. Grossmann, M. Bauer, N. Hönle, U.-P. Käppeler, D. Nicklas and T. Schwarz: Efficiently Managing Context Information for Large-scale Scenarios. Proceedings of the 3rd IEEE Conference on Pervasive Computing and Communications (PerCom 2005), Kauai Island, Hawaii, USA, 2005

[HHT+01] T. Höllerer, D. Hallaway, N. Tienna and S. Feiner: Steps towards accommodating variable position tracking accuracy in a mobile augmented reality system. IJCAI-2001 Working Notes: AI in Mobile Systems (AIMS), 2001

[KOA+99] C. Kidd, R. Orr, G. Abowd, C. Atkeson, I. Essa, B. B. MacIntyre, E. Mynatt, T. Starner and W. Newstetter: The Aware Home: A Living Laboratory for Ubiquitous Computing Research. Proceedings of the Second International Workshop on Cooperative Buildings (CoBuild), 1999

[LKA+96] S. Long, R. Kooper, G.D. Abowd and C.G. Atkeson: Rapid prototyping of mobile context-aware applications: The cyberguide case study. Proceedings of the Second Annual International Conference on Mobile Computing and Networking (MobiCom '96), Rye, New York, USA, 1996

[Nexus] The Nexus Project Homepage: http://www.nexus.uni-stuttgart.de

[NM04] D. Nicklas and B. Mitschang: On building location aware applications using an open platform based on the NEXUS Augmented World Model. Software and Systems Modeling, 2004

[RC00] M. Roman and R.H. Campbell: GAIA: Enabling Active Spaces, Proceedings of the 9th ACM SIGOPS European Workshop, Kolding, Denmark, 2000

[SGH+99] N. A. Streitz, J. Geißler, T. Holmer, S. Konomi, C. Müller-Tomfelde, W. Reischl, P. Rexroth, P. Seitz and R. Steinmetz: i-LAND: An interactive Landscape for Creativity and Innovation. ACM Conference on Human Factors in Computing Systems (CHI'99), Pittsburgh, Pennsylvania, USA, 1999