NEXUS—A Platform for Context-aware Applications

Frank Dürr, Nicola Hönle, Daniela Nicklas, Christian Becker, and Kurt Rothermel
Institute of Parallel and Distributed Systems (IPVS)
Universität Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany
frank.duerr@informatik.uni-stuttgart.de

Abstract

In this paper we present the Nexus Platform for context-aware applications. This platform allows to share the effort of setting up a global and detailed context model between different providers by federating their partial models. Applications can query data directly, use push-based communication through an event service, or use value-added services like a navigation or map service for special tasks. Additional context-aware services like hoarding or geocast can be implemented on basis of the platform. For the latter we present different approaches for addressing and efficient message forwarding based on the Nexus Platform.

I. Introduction

The proliferation of mobile communication and the integration of sensors like positioning sensors into mobile devices give rise to the class of so-called *context-aware applications*. Such context-aware applications can trigger actions, or select and present information based on their current context, e.g. the current geographic position of the user or his current speed [1]. Typical examples of context-aware applications are friend finders or context-aware tourist guides, which select and present information about sights in the vicinity of the user. Since these two examples focus on location information as the primary context information they are also called *location-based services*.

All kinds of context-aware applications rely on some kind of context model. Since the effort to set up a global and at the same time detailed model is very high, current context-aware applications use relatively coarse-grained or spatially constraint models tailored to the applications' requirements. In contrast to these models, the goal of the Nexus Platform is to support all kinds of context-aware applications by providing a shared global context model. Partial models from various providers are integrated and federated by the platform, and together form a highly detailed so-called spatial world model.

In this paper we present an overview of the Nexus Platform in Section II, and a geographic communication mechanism that builds on this platform in Section III before the paper concludes with a short summary.

II. The Nexus Platform

To achieve the goal of a global spatial world model, the platform depicted in Figure 1 federates local context models from so-called *context servers*. The local models contain different types of context information: representations of real world objects like streets, rooms, or persons and virtual objects that link to digital information spaces. Sensors keep the data of local context models up to date (e.g. the position of a person).

Published in 1. GI/ITG KuVS Fachgespräch Ortsbezogene Anwendungen und Dienste. Informatik-Bericht 317, FernUniversität in Hagen, Juni 2004.

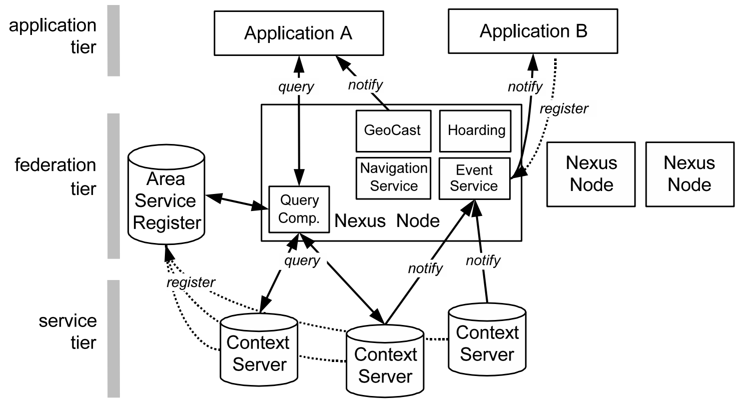

Fig. 1. The Nexus Platform

A. Context Server

A context server stores a local context model. It is comparable to a web server in the WWW. The server has to fulfill two requirements in order to be part of the Nexus Platform: it has to implement a certain interface (allowing simple spatial queries and results in a specified XML language) and it should be registered with its service area and object types with the so-called *Area Service Register* (comparable to a spatially enhanced DNS, see below).

There can be many different implementations of a context server. For providing large scale geographical models, we used a spatially enhanced database. We cope with the high update rates of the positions of mobile users using a distributed main memory system [2]. Even small-scaled sensor platforms like the ContextCube [3] can be used as context server.

B. Federation

A federation node mediates between applications and context servers. It has the same interface as a context server, but does not store models (except caching). Instead, it analyzes an application query, determines the context servers that could answer the query, and distributes the query to these servers. Then it combines the incoming result sets to a consistent view and returns it to the application.

For query distribution and service discovery, a Nexus node uses the *Area Service Register (ASR)*. This service is a directory of the available local context models and stores the address of their context server, their object types, and their spatial extent. More details about the federation tier can be found in [4].

C. Additional Services

In addition to the query features, every Nexus node supports advanced value-added services that have their own interfaces and use the federated context model. Figure 1 shows four different value-added services of the Nexus Platform: The event service monitors spatial events, combining basic events into more complex events. It allows for the processing of spatial predicates, such as ”two of my friends meet”. The navigation service computes multimodal navigation routes across the borders of local context models. With geocast [5], one can send messages to every static or mobile host that is currently located in a specific region by addressing this geographic area. Hoarding is the process of preloading data based on statistical methods to support disconnected operation [6]: if an application is in the act of leaving an area with wireless network communication, the hoarding service transfers context information to these applications in advance that will most likely be needed.

D. Applications

A context-aware application can use the Nexus Platform in three different ways. First, it can send queries to the federation to get information about its surrounding including infrastructure, points of interest, mobile objects (friend finder), and so on. Secondly, the application can register to the event service to receive a notification when a certain state of the world occurs, e.g. when the user has entered a building or the temperature in a room exceeds a certain threshold. Thirdly, it can use value-added services like the map or navigation service to shift common functions to the infrastructure.

III. Geocast Based on Nexus Platform

With geocast one can sent messages to all hosts in a certain geographic area called the target area of the message. This communication mechanism benefits considerably from a detailed and global spatial world model like the one provided by the Nexus Platform.

A. Fine-grained Addressing

In conjunction with a fine-grained addressing scheme like the one presented in [5], geocast messages can be sent to large target areas like whole cities as well as to comparably small target areas like single rooms or even parts of a room. Messages can be addressed geometrically by polygons or by symbolic names like room numbers, street names, etc. With geometric addressing almost arbitrary target areas can be addressed, whereas symbolic addresses are intuitive to use since one can use addresses associated with well-known semantics used in everyday life. The Nexus Platform already supports two-dimensional geometric addressing, and we are working on the integration of symbolic locations, so we can take advantage of both types of addressing. Even combinations of symbolic and geometric addresses (hybrid addressing) are feasible, e.g. to address a small part within a room geometrically whereas the room itself is determined symbolically.

The Nexus world model manages information beyond mere location information like the types of objects and further object attributes. By using this information the receiver group within a certain area can be further refined leading to the concept of geographic multicast. For instance, a message can be sent to all taxis near the main station of a town.

B. Efficient Message Forwarding

Beside a fine-grained addressing scheme, geocast also requires mechanisms to forward messages from the sender to the target area efficiently. Since our addressing scheme allows to address very small target areas with few receivers up to large target areas with many receivers, scalability is a major issue. The following approaches for geocast message forwarding are supported by the Nexus Platform.

The Nexus Platform manages the geographic positions of mobile and static objects. A simple geocast implementation can query for the mobile or static hosts within the target area of the message (i.e. use a range query) and then send the message directly to these hosts using for instance multiple unicast messages. Obviously, this approach is only applicable if the receiver group is small.

Instead of directly sending geocast messages to the receivers, one can also think about a two-phase approach. First, the geocast message is forwarded from the sender to the access networks intersecting the target area. Secondly, the message is distributed within these access networks to the hosts in the target area. This approach requires a model of the geographic coverage of access networks, which will be part of the Nexus Platform. For messages to mobile target areas like a train or a bus, we can query the Nexus Platform for the current position of the mobile object in order to determine the intersecting access networks in a second step. If messages are sent to the access networks via unicast, this approach is applicable for target areas intersecting few access networks.

For large target areas or dense access network coverages, using a unicast mechanism to send geocast messages to the access networks in the target area is prohibitive. Instead, an infrastructure of special *GeoRouters* can be used for efficient forwarding. [7] describes an approach, where GeoRouters are assigned with geographic services areas. GeoRouters form a hierarchy according to the inclusion relationship of their service areas, and messages are forwarded along this hierarchy by comparing the target area of the message and the routers' service areas. Since this approach is tailored to geometric addresses, we have to adapt it to symbolic and hybrid addressing in order to use it together with our fine-grained addressing scheme.

A target area address can also be mapped to a multicast address associated with a certain geographic area before a geocast message is forwarded based on this multicast address. Such a multicast address is shorter than the target area address resulting in less overhead and it can be handled more efficiently. For this approach, the Nexus Platform manages the mappings from geographic addresses to multicast addresses. However, we have to keep in mind that for a fine-grained location model the number of multicast addresses can be huge. Therefore, we have to pay attention that routing tables do not grow to large.

IV. Summary

In this paper we gave a brief overview of the Nexus Platform for context-aware applications. This platform allows to share the effort of setting up a global and detailed context model between different providers by federating their partial models. Applications can query data directly, use push-based communication through an event service, or use value-added services like a navigation or map service for special tasks. Additional context-aware services like hoarding or geocast can be implemented on basis of the platform. For the latter we presented different approaches for addressing and efficient message forwarding.

References

[1] Kurt Rothermel, Dominique Dudkowski, Frank Dürr, Martin Bauer, and Christian Becker, "Ubiquitous computing – more than computing anytime anyplace," in *Proceedings of the 49th Photogrammetric Week*, Stuttgart, Germany, Sept. 2003.

[2] Alexander Leonhardi and Kurt Rothermel, "Architecture of a large-scale location service," in *Proceedings of the 22nd Conference on Distributed Computing Systems (ICDCS 2002)*, Vienna, Austria, July 2002, pp. 465–466.

[3] Martin Bauer, Christian Becker, Jörg Hähner, and Gregor Schiele, "ContextCube—providing context information ubiquitously," in *Proceedings of the 23rd International Conference on Distributed Computing Systems Workshops (ICDCS 2003)*, May 2003, pp. 308–313.

[4] Daniela Nicklas, Matthias Großmann, Thomas Schwarz, and Steffen Volz, "A model-based, open architecture for mobile, spatially aware applications," in *Proceedings of the 7th International Symposium on Spatial and Temporal Databases (SSTD 2001)*, Christian S. Jensen, Markus Schneider, Bernhard Seeger, and Vassilis J. Tsotras, Eds., Redondo Beach, CA, USA, July 2001.

[5] Frank Dürr and Kurt Rothermel, "On a location model for fine-grained geocast," in *Proceedings of the Fifth International Conference on Ubiquitous Computing (UbiComp 2003)*, Anind Dey, Albrecht Schmidt, and Joe McCarthy, Eds., Seattle, WA, Oct. 2003, Lecture Notes in Computer Science 2864, pp. 18–35, Springer.

[6] Susanne Bürklen, Pedro José Marrón, and Kurt Rothermel, "An enhanced hoarding approach based on graph analysis," in *Proceedings of the 5th IEEE International Conference on Mobile Data Management (MDM 2004)*, Berkeley, California, USA, Jan. 2004.

[7] Julio C. Navas and Tomasz Imielinski, "Geocast – geographic addressing and routing," in *Proceedings of the Third Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '97)*, Budapest, Hungary, Sept. 1997, pp. 66–76.