A Research Platform for Location-Based Applications

Dominic Heutelbeck and Matthias Hemmje
Computer Science Department, University of Hagen
{dominic.heutelbeck|matthias.hemmje}@fernuni-hagen.de

1 Introduction

This paper presents a short overview of the platform ArWorx, a platform for location-based applications. Among others, it addresses distributed geographical data-sharing, geographical resource discovery, network services, and information visualization.

Location-based services are gaining an increasing importance. In literature, numerous different definitions for location-based services and applications exist, e.g., [1, 2, 4, 3]. They all have in common, that the location of a mobile device or geographical data is used for different tasks in an application to provide some valuable service to the user. In this paper, a very general notion for location-based applications is used:

Definition 1 *A **location-based application** is an application or network service that operates with geographical data from a part of the earth's surface. This part of the earth's surface is called the **service-area** of the location-based application.*

This definition includes all former definitions of location-based services. The word *application* is used instead of *service* to emphasize that the platform *ArWorx* does not only support location-based network services, but also arbitrary applications like geographic information systems, route planning, and location-based games. All these applications operate with geographical data and they may be used in a dynamic geographical context.

2 Requirements

As location-based applications gain acceptance, it becomes increasingly difficult for a user to find the right service. The number of services increases, the service-areas may be small and mobile, and the services may be only temporarily available. For a user, it is important to keep track of location-based services that can aid him in his current spatial context.

Requirement 1 *The platform must support the user in finding locally relevant location-based applications.*

A user may use more than one location-based application at the same time, e.g., he is using a navigation tool and has a friend finder application running at the same time. While this also implies requirements for the underlying operation system, it also means that these applications must be easily accessible to the user through a standard user interface.

Requirement 2 *The platform must support the presentation of multiple concurrent location-based applications in an accessible standard user interface.*

In addition to these requirements from the perspective of the user, it must also be easy for an application provider to publish a new location-based application. In this case, a location-based application may be a large scale information system or a single session of a mobile location-based multiplayer game.

Requirement 3 *The platform must provide an efficient way of publishing new location-based applications.*

The architecture of location-based applications is not limited to simple stand-alone applications or client-server systems. Location-based applications can be arbitrary distributed systems.

Requirement 4 *The platform must support applications with different distributed architectures.*

The hardware in this scenario is very heterogeneous and includes mobile phones, PDAs, Laptops, Tablet PCs, wearable computers, desktop workstations, and large servers. Besides their computational power and network connectivity, these devices have very different user interfaces. A single location-based service will be accessed via different kinds of devices. Thus, a matching user interface has to be provided for each platform. In addition, different classes of user interfaces like window based or augmented reality interfaces are based on completely different interaction metaphors. User interface plasticity [7], i.e., the automatic transformation of user interfaces, is an active research topic. However, in practice the user interface for a single application has to be redesigned for each type of target device.

Requirement 5 *The platform must support an extensible set of target devices and provide standard interaction widgets for each supported device type.*

A platform for location-based application should also support the basic location-based network services.

Requirement 6 *The platform must support basic location-based network services:*

- *Geocasting.*
- *Trigger services.*

These requirements are the key requirements addressed by the platform ArWorx. In addition, the following basic requirements apply:

Requirement 7

- *The platform must provide a driver infrastructure for location sensing and tracking hardware.*
- *The platform must provide a library for basic operations with different geographical coordinate spaces.*
- *The platform must provide a library for handling geometrical objects.*

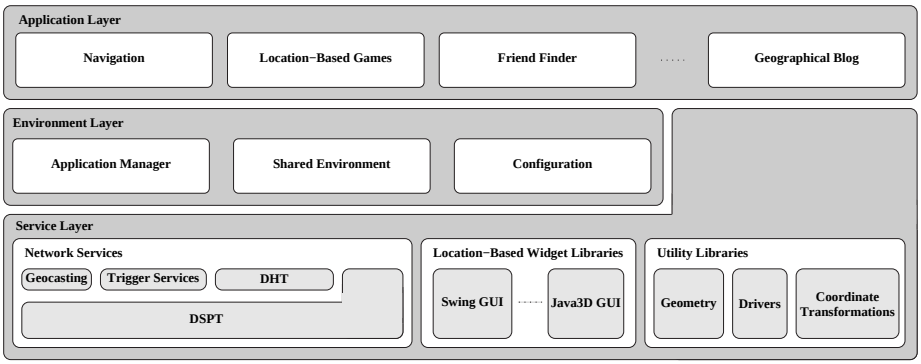

Figure 1: The architecture of ArWorx.

3 The Architecture of ArWorx

ArWorx is based on a three layer architecture as illustrated in figure 1. The base layer of ArWorx is the *service layer*. The service layer consists of three basic components:

- *Network services*: This component consists of the basic network services geocasting, trigger services, a distributed hash table (DHT), and a distributed space partitioning tree (DSPT) for storing geographical data.
- *Location-based widget libraries*: This component is a collection of user interface elements for different classes of mobile devices. The libraries implement location-based elements like interactive map widgets for tablet style devices, or 3D widgets for augmented reality systems.
- *Utility libraries*: This component is a collection of libraries that provide some fundamental functions for location-based applications, e.g., device drivers for positioning systems, tools for handling geometrical objects, and functions to transform locations between different standard geographical coordinate systems.

The second layer in ArWorx is the *environment layer*. The function of the environment layer is to manage the user's personal environment. In ArWorx, the user's environment is defined by the user's spatial context and by the set of active location-based applications. The environment layer offers a *shared environment*. It is used to aggregate the location-based information from different simultaneously running location-based applications. In addition, the environment layer provides an application manager tool that allows to run, stop, and manage location based applications. The tool offers search for location-based applications according to their service-area. Finally, the environment layer also offers users the possibility to customize their user interface and to configure their environment, including available positioning systems. The role of the environment layer is similar to the role of a desktop environment for classical applications. The environment layer makes use of the services and libraries offered by the service layer. The final layer is the *application layer* that contains the actual location-based applications.

4 Comparison to Related Work

The two major existing platforms for location-based applications are Nimbus [6] and Nexus [5]. The approach of the platform ArWorx differs significantly. Some problems addressed by these platforms, such as geocasting and trigger services, are also addressed by ArWorx. However, the three platforms are based on three different architectures that have different strengths. RectNet is a peer-to-peer platform, Nimbus is a completely decentralized self-organizing infrastructure of fixed servers, and Nexus is a federation of servers with some central components. Nexus and Nimbus are both based on a distributed location model and the central purpose of these platforms is to support applications dealing with this model. The motivation for ArWorx is different. It is based on the thesis that the number of location-based applications and their individualization will increase drastically in the future. Thus, users will require support in handling these location-based applications. ArWorx offers this support. In addition, ArWorx introduces the concept of shared environments to help users to manage multiple simultaneously active location-based applications. These aspects are not addressed by the platforms Nimbus and Nexus.

5 Conclusion

The Platform ArWorx was realized using Java, and a number of interesting location-based applications were realized. The developers of the applications, such as a friend finder, a geographical messenger, a location-based Pong game, and a geographical blog, were relieved from many standard tasks, such as the mapping between geographical coordinate systems, or the management of distributed data. The developers were able to concentrate on the actual application logic. The shared environment proofed to be a powerful concept to aggregate location-based information and applications. The application manager enabled users to search for location-based applications in an intuitive and efficient way.

In future publications we will describe the different solutions offered by ArWorx in detail. E.g., the shared environment, or distributed space partitioning tree (DSPT), a peer-to-peer datastructure for dynamic geometrical objects that is used to realize distributed geographical data-sharing.

References

[1] T. Bychowski, M. Mbrouk, H. Niedzwiadek, J. Herring, and J.-F. Gaillet. OpenGIS Location Services (OpenLS): Part 6–Navigation Service. Technical report, Open GIS Consortium Inc., 2003.

[2] GSM-World. GSM - Location Based Services. *PRD SE. 23*, 2002. www.gsmworld.com.

[3] Ovum Ltd. *Mobile Location Services*. 2000.

[4] M. Mabrouk, T. Bychowski, H. Niedzwiadek, Y. Bishr, J.-F. Gaillet, N. Crisp, W. Wilbrink, M. Horhammer, G. Roy, and S. Margoulis. OpenGIS Location Services (OpenLS): Core Services. Technical report, Open GIS Consortium Inc., 2004.

[5] University of Stuttgart. Sonderforschungsbereich – Umgebungsmodelle für mobile Kontextbezogene Systeme. www.nexus.uni-stuttgart.de.

[6] Jörg Roth. A Decentralized Location Service Providing Semantic Locations. Habilitation thesis, October 2004.

[7] D. Thevenin and J. Coutaz. Plasticity of User Interfaces: Framework and Research Agenda, 1999.