A Peer-to-Peer Based Infrastructure for Context Distribution in Mobile and Ubiquitous Environments*

Xiaoming Hu[1], Yun Ding[1], Nearchos Paspallis[2], George A. Papadopoulos[2], Pyrros Bratskas[2], Paolo Barone[3], Alessandro Mamelli[3], Yves Vanrompay[4], and Yolande Berbers[4]

[1] European Media Laboratory, Schloss-Wolfsbrunnenweg 33, 69118 Heidelberg, Germany
[2] Department of Computer Science, University of Cyprus, CY-1678, Nicosia, Cyprus
[3] HP Italy, Via G. Di Vittorio 9 20063 Cernusco sul Naviglio, Italy
[4] K.U.Leuven, Celestijnenlaan, 200A B-3001 Leuven, Belgium
{xiaoming.hu,yun.ding}@eml-d.villa-bosch.de,
{nearchos,george,bratskas}@cs.ucy.ac.cy,
{paolo.barone,alessandro.mamelli}@hp.com,
{yves.vanrompay,yolande.berbers}@cs.kuleuven.be

Abstract. The increasing proliferation of mobile devices has raised the expectations for user-customized and environment-aware services. However, mobile context-aware systems inherently feature characteristics of distribution and heterogeneity which pose great challenges to their developers. This paper focuses on context distribution in mobile and ubiquitous computing environments. We propose a peer-to-peer based context distribution approach and evaluate it against the derived requirements.

Keywords: Context awareness, peer-to-peer systems, JXTA.

1 Introduction

An increasing wealth of context information is steadily becoming available in modern computing systems. Our living room becomes aware of our presence and mood, and our car becomes aware of rain conditions, etc. However, little has been done to enable greater dissemination of such context information across distributed devices. Allowing individual context-aware devices to communicate with each other can provide increased levels of synergy in terms of resource saving and reusability. In this paper we describe initial results from our work on context distribution with particular emphasis on mobile and ubiquitous environments. We have identified the following requirements: *heterogeneity* implies the involvement of various devices, a plethora of network protocols and different context modeling technologies. Large numbers of involved devices impose the need for scalable, decentralized approaches such as *localized scalability* [1]. *Security* provisions are needed to guarantee the *privacy* of sensitive context information. *Robustness* is required to dynamically detect and cope

* The work is partially funded by the European Union (6th Framework Programme, contract number 35166) in the scope of the MUSIC project. The authors from the European Media Laboratory GmbH are supported by the Klaus Tschira Foundation.

R. Meersman, Z. Tari, P. Herrero et al. (Eds.): OTM 2007 Workshops, Part I, LNCS 4805, pp. 236–239, 2007.

with spontaneous changes of context providers and consumers. The context system must be *light-weight* to be deployed in resource-constrained embedded devices of various sizes and capabilities. Finally, aiming to reduce the complexity of the development of context-aware applications, the context system must be *easy-to-use*.

2 A Peer-to-Peer Based Context Distribution System

We divide the entities of a context distribution system into three categories according to their device capabilities and functionalities (Fig. 1(a)). *Sensor peers* generate low-level raw context data from distributed sensors. *Disseminator peers* are resource-rich devices which act as context processors, distributors and consumers. *Consumer peers* are only context consumers. Usually, they have limited resources, and can not afford to process (i.e., aggregate and reason about) context data.

Fig.1 (b) illustrates the proposed peer-to-peer (P2P) based infrastructure for context distribution. Each disseminator peer hosts the infrastructure, while a consumer peer can omit the repository due to its resource constraints. The *Core* provides essential P2P services. *Discovery* allows a peer to actively poll for other peers and services, or be notified of their arrivals and departures. *Group formation* enables a peer to create, join and leave peer groups. Peers will be notified of new or lost group members. Furthermore, *peer communication* allows a peer to send messages to others, or to propagate messages within a peer group. The *Context Service* is designed as a single interface for context providers to insert context information, and for context clients to acquire context information. The *Repository* maintains context information. The *Distribution Service* is an internal service, which enables federated context services. Context requests which cannot be satisfied by the local *Context Service* are forwarded to connected, remote *Context Services* via the *Distribution Service*.

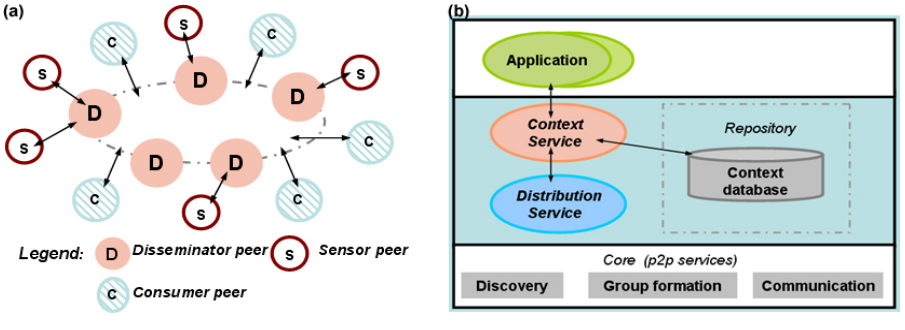

Fig. 1. (a) Three types of peers in the context distribution system (b) A peer-to-peer infrastructure for context distribution

2.1 Context Network and Distribution Service

Consumer and disseminator peers dynamically form groups to exchange context data. The *Context Network* is the default group which each peer joins during its bootstrap.

The *Context Service* and the *Distribution Service* which are provided by the *Context Network* are so-called "group services", meaning that they are provided collectively and can be only accessed by group members. The collectiveness provides fault-tolerance in a dynamic environment. If one of the group members fails, other members can still continue providing their services.

Whenever entering the network, a disseminator peer publishes its self-description in order to be discoverable by other peers. It also searches for existing disseminator peers. Each disseminator peer maintains a disseminator view which is a list of known disseminator peers in the group. Since peers can appear and disappear at any time, we use an algorithm which is similar to the one used by JXTA rendezvous peers [2] to keep the local views loosely consistent among the peers. Fig. 2 (a) shows the federation of the disseminator peers. When a consumer peer joins the *Context Network*, it connects to one or multiple disseminator peers in this group.

Multiple instances of the *Context Services* are connected via the federation of their associated peers. Upon receiving a request for context information, the *Context Service* of a disseminator peer checks the local context repository. If the required information is not available, the *Context Service* utilizes the *Distribution Service* to find the information from remotely connected *Context Services*. The request message is firstly sent to its known disseminator peers. The receiving remote disseminator peer passes the incoming request to its local *Context Service* for processing, and possibly sends a positive answer back to the requesting disseminator peer. If a consumer peer propagates a context request into the group, this request is directly forwarded to its connected disseminator peers in order to locate the answer. Additionally, the *Distribution Service* is responsible to propagate the context change events to remote peers within the group.

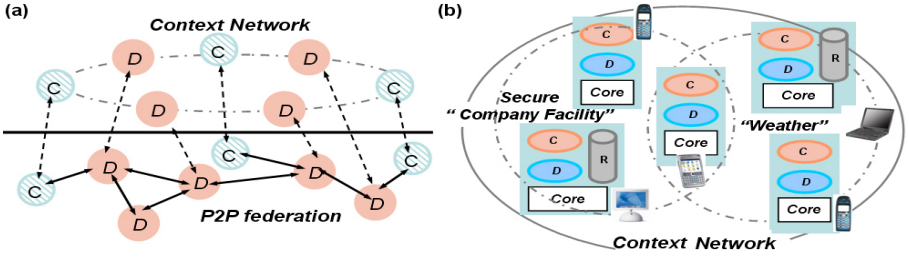

Fig. 2. (a) Federation of disseminator peers (b) Hierarchically organized Context Groups

2.2 Scope of Context Distribution

The root group *Context Network* can be divided into subgroups to restrict information sharing. Each subgroup *Context Group* consists of a dynamic set of peers that have common interests for certain context data, and have agreed upon a set of policies (e.g., authentication of membership). Each peer may simultaneously belong to several subgroups. The *Context Network* and its subgroups are hierarchically organized (see Fig.2 (b)). A subgroup inherits all the group services of its parent. The *Context Group* provides a scoping mechanism for restricting the propagation of search requests and update events. Interests of consumer peers in context data might change at runtime.

The *Distribution Service* maps the interests to dedicated *Context Groups*, and joins or leaves them on behalf of the associated peer. Moreover, in order to protect private, sensitive context information, a specific policy can be enforced within a *Context Group* to control the access right. Peers should be approved by a group authentication service before being allowed to join the group and utilize the group services.

3 Evaluation

A prototype of the proposed infrastructure has been partially implemented[1]. It is set on top of JXTA, a standard technology that allows connected devices on the network to collaborate in a P2P manner [3]. This section evaluates our approach by comparing it with the requirements identified in Section 1.

The proposed system inherits the support for *heterogeneity* from JXTA technology, which allows the creation of peer networks independent of operating systems, programming languages, network protocols and the nature of the devices acting as peers. The differentiation of roles of peers based on the capabilities of their hosting devices, and the mechanism of group formation allowing context sharing in restricted scope, enable a high degree of *scalability*. Leveraging JXTA technology, the system also provides a mechanism for *access control* over the information offered as well as a group authentication service to ensure *privacy.* Taking advantage of the discovery service provided by JXTA, our proposed system satisfies the requirement for *robustness*, arising from the instability of network connections which is common in mobile networks. Additionally, the implementation on top of JXME [3], a JXTA implementation for J2ME, provides a *light-weight* solution that can run on resource constrained devices. The single interface for accessing context data grants *ease of use.*

4 Conclusions

This paper introduced a peer-to-peer based context distribution system. It builds upon a thoroughly-studied P2P standard, the JXTA framework, which provides many facilities for discovering, communicating and safeguarding to meet the identified requirements. For future work, the prototype implementation will be completed and validated in a set of case studies.

References

1. Satyanarayanan, M.: Pervasive Computing: Vision and Challenges. IEEE Personal Communications, 10–17 (2001)
2. Abdelaziz, M., Pouyoul, E., Traversat, B.: Project JXTA: A Loosely-Consistent DHT Rendezvous Walker (2004), http://research.sun.com/spotlight/misc/jxta-dht.pdf
3. JXTA Community Projects, https://jxta.dev.java.net, https://jxta-jxme.dev.java.net/

[1] http://www.igd.fhg.de/igd-a1/dynamite-project/software/tools/ubinet-javadoc.zip