



HAL
open science

Towards a peer-to-peer middleware for context provisioning in spontaneous networks

Tuan Dung Nguyen, Siegfried Rouvrais

► **To cite this version:**

Tuan Dung Nguyen, Siegfried Rouvrais. Towards a peer-to-peer middleware for context provisioning in spontaneous networks. ESF-MiNEMA 2007: 5th Workshop on Middleware for Network Eccentric and Mobile Applications, Sep 2007, Magdeburg, Germany. pp.54 - 57. hal-01333271

HAL Id: hal-01333271

<https://hal.science/hal-01333271>

Submitted on 28 May 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards a Peer-to-peer Middleware for Context Provisioning in Spontaneous Networks

Tuan Dung Nguyen, Siegfried Rouvrais
GET / ENST Bretagne
Technopôle de Brest Iroise, 29238 Brest, France
{td.nguyen,siegfried.rouvrais}@enst-bretagne.fr

Abstract

Context-awareness has been widely considered important for applications to provide adaptable services to changes in dynamic environments. Classical approaches have mainly focused on gathering and representing internal contextual information from integrated sensors. Recent work has shown that useful external contexts can also be acquired from the surrounding environment in a distributed manner. However, the dynamism and heterogeneity of spontaneous networks raise new challenges for the provisioning of such information. This paper describes our ongoing work on a peer-to-peer middleware for abstracting contexts as logical peers in independent overlay networks. Context data retrieval then becomes transparent to applications and other middleware services. The proposed middleware will support efficient construction and maintenance of these overlays and offer common interfaces to applications. We discuss in detail its architecture and the open issues to be resolved.

1 Introduction

The increasing popularity of wireless computing devices has resulted in the spontaneous networking paradigm where a large number of nodes (e.g. PDAs, smart phones) collaborate to provide services without neither predefined infrastructures nor any manual interventions. Such environments are characterized by node's heterogeneity, potential mobility, unpredicted disconnection and limited resources. Systems and applications should be able to provide appropriate services based on user's current context (e.g. location, time, social situation).

Dey [3] defined context as any information that can be used to characterize the situation of an entity (e.g. person, place, and object). This information helps applications to dynamically adapt their behavior according to their require-

ments and to changes in the environment. Contexts can be either internal (e.g. user's profile, device's battery level) or external (e.g. nearby resources, temperature). They can also be static (e.g. user agenda) or dynamic (e.g. network bandwidth). For instance, a car may need various contextual information such as weather, nearby petrol stations, road status.

Context provisioning mainly consists of contextual information acquisition, sharing and interpretation. Internal contexts are often captured by inherently integrated physical and/or logical sensors. External contexts are acquired from the surrounding environment either in a centralized or distributed manner. In the former approach, specific entities are used for context storage and lookup while in the latter case, context data is provided by any entities in proximity of an ad hoc network [13]. The centralized approach is not relevant to spontaneous networks due to the lack of infrastructure and the frequent mobility.

Overlay networks are fundamental building blocks of peer-to-peer systems where nodes are connected by logical links on top of underlying physical networks. Their self-organization and sharing capability is a good support for dynamic environments.

Our work is highly motivated by the following requirements for context provisioning in spontaneous networks:

- **Context representation.** Various contexts can be gathered from different sources. An adequate representation should respect the separation of concerns design principle for context management and enable transparent context provisioning to upper applications.
- **Context sharing.** Resource-constrained and heterogeneous devices have only a limited number of sensors for capturing contextual information. Peer-to-peer context sharing allows nodes to obtain needed contexts not only from their integrated sensors but also from their neighbors.

The contribution of this paper is a peer-to-peer middle-

ware for managing different contexts on independent overlays. This framework allows resources-constrained nodes to maintain only currently needed contexts according to their available resources and their own non-functional requirements.

The rest of this paper is organized as follows. Section 2 presents our middleware architecture for multiple context overlays. Next, Section 3 reviews related work and finally, Section 4 concludes the paper by discussing some open questions.

2 Dynamic overlays for context provisioning

A physical device can have several integrated sensors for capturing contexts. However, as illustrated in Figure 1, a device can have a number of context peers (CxtPeer) corresponding to its needed contexts. According to application's needs and available resources, logical peers can be created and destroyed on-the-fly. Peers are connected in independent self-organized overlays. The context data stored in each peer can be provided by integrated sensors or by querying other peers. The content can be cached in a peer during a defined period for later lookup operations.

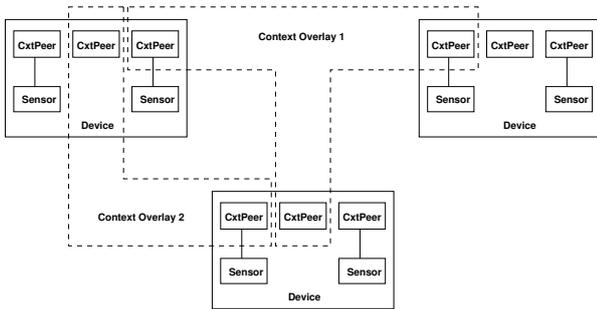


Figure 1. Dynamic overlays of context peers

Figure 2 presents our middleware architecture. The main components are described as follows.

- Context Manager.** This component is responsible for dynamic context peer creation and destruction. New context peers can be created from a proposed specification language. It also provides applications a common interface to query context data. The context query processing can be synchronous or asynchronous, i.e. applications can submit a context query and receive the result after some acceptable delay.
- Context Overlay Manager.** This component is responsible for overlay maintenance due to node's mobility in the proximity. Here, CxtPeer components are

grouped based on its context semantic. The maintenance can also be triggered on the creation/destruction of a new context peer.

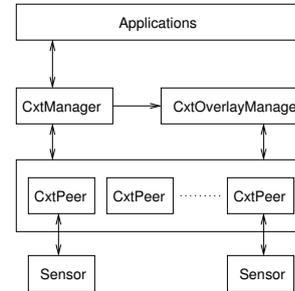


Figure 2. Middleware architecture

The software model of our proposed system is depicted in Figure 3. CxtPeer maintains list of neighbors representing a local view of the corresponding overlay. These neighbor lists are dynamically maintained by CxtOverlayManager. A CxtPeer has a Context data which can be associated or not with an integrated sensor.

When a context lookup query arrives at CxtManager, it is forwarded to the corresponding CxtPeer. Then the context data can be retrieved from a cached content, an integrated sensor or by querying other peers. These operations are carried out in the middleware layer and transparent to applications. By this way, some context data can be retrieved thanks to node's mobility even in partitioned networks.

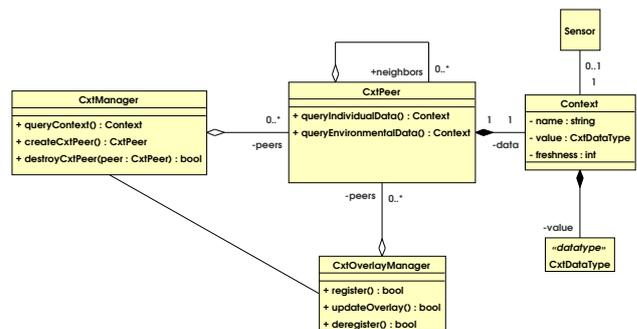


Figure 3. Context provisioning model

Fractal [4] is a modular, extensible and language-independent component model to design, implement, deploy and reconfigure middleware systems and applications. Our middleware model based on the Fractal is depicted in Figure 4. CxtPeer is a primitive component representing a context. These components are managed in a composite component called CxtManager. Its content part is composed

of a finite number of CxtPeer components. Its controller interfaces (namely CC, LC) permit the creation and destruction of primitives peers on-the-fly. A CxtPeer has a server interface providing the corresponding contextual information. Its client interface is connected to the client interface of its surrounding component CxtManager. Different context overlays are maintained dynamically by distributed binding between server and client interfaces of CxtManager components (e.g. using Fractal RMI [4]). These maintenance operations are carried out by CxtOverlayManager component.

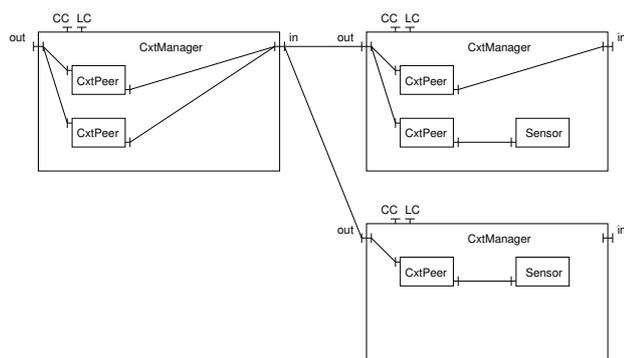


Figure 4. Fractal component-based model

3 Related work

The tuple-space model was first presented with Linda [5] and then adapted to mobility with LIME [12], TinyLIME [2] or ErgoSpaces [10]. These efforts have proved the interest of this data-sharing model to dynamic environments. However, existing works lack the capacity to efficiently manage different context sources according to various application's needs.

Using peer-to-peer overlays for context lookup was first presented in [9]. The authors proposed ContextBus, an architecture in which context producers are grouped based on the semantic of their data. An improved version namely Semantic Context Space (SCS), was also proposed to reduce the maintenance cost when the number of semantic groups increases using a one-dimensional ring space [7, 8]. Unlike this work, we propose a more abstract concept because any devices can have a context peer even if they do not directly produce context data. Moreover, we support dynamic overlay management and effective mechanisms for overlay maintenance.

Research work has recently put great attention on the coexistence of multiple overlays. Gridkit [6] is a component framework for pluggable overlay networks even at runtime. Behnel et al. [1] combined Gridkit with an over-

lay design modeling framework to enable the generation of specific code from platform-independent design of overlay networks. Maniymaran et al. [11] presented a mechanism to reduce the multiple overlays maintenance overhead without sacrificing the performance. The authors proposed that the maintenance of one overlay can be leveraged to partly maintain another overlay with no extra cost. These approaches form a good base for the overlay construction and maintenance in our framework. However, the coexistence of multiple context overlays needs further investigation.

4 Discussion and open questions

We have presented a peer-to-peer middleware architecture for context provisioning in spontaneous networks. The proposed system offers a means for context abstraction and sharing in such dynamic environments. We plan to evaluate our proposition with a prototype working on real PDAs. By comparing it to existing tuple-space based works, we will demonstrate its capacity to answer to its design requirements and its overall overhead is acceptable. We have also identified the three following issues to be investigated.

1. **Context peer specification and deployment.** We are investigating a specification language and a component framework for the dynamic management of overlays. The language will allow applications to specify new context peers based on their current requirements. The component-based framework enables on-demand deployment of different context peers. To avoid ambiguity among different contexts, ontology-based approaches can be investigated in future work.
2. **Context overlay construction and maintenance.** As context peers are connected in unstructured overlays, the maintenance operations involve two phases: a peer sampling phase providing candidates for the neighbor list establishment based on the semantic of overlays. We plan to exploit epidemic-style approaches [14] for peer sampling service in spontaneous networks. Then the coexistence of multiple overlays can be leveraged to improve the maintenance efficiency by exploiting the common peer sampling service and the semantic relations among overlays, i.e. binding of subcomponents in a common composite component. The maintenance principle consists of updating neighbors in each overlay through dynamic interface binding of distributed components and assuring data consistency among context peers.
3. **Context overlay lookup and composition.** Context provisioning to applications is realized through lookup operations in overlays. Therefore, we need effective lookup mechanisms to improve system performance.

Moreover, a query for higher-level contexts may concern several context peers which results in the composition of multiple overlays. So far, few work has put attention on this issue and further investigation is needed.

References

- [1] S. Behnel, A. Buchmann, P. Grace, B. Porter, and G. Coulson. A Specification-to-Deployment Architecture for Overlay Networks. In *Proc. 8th OTM Int. Symp. on Distributed Objects and Applications (DOA'06)*, pages 1522–1540, Montpellier, France, Oct. 2006. Springer-Verlag.
- [2] C. Curino, M. Giani, M. Giorgetta, A. Giusti, A. L. Murphy, and G. P. Picco. Mobile Data Collection in Sensor Networks: The TinyLime Middleware. *Pervasive and Mobile Computing*, 4(1):446–469, Dec. 2005.
- [3] A. K. Dey. Understanding and Using Context. *Personal Ubiquitous Computing*, 5(1):4–7, 2001.
- [4] Fractal. <http://fractal.objectweb.org/>, 2006.
- [5] D. Gelernter. Generative Communication in Linda. *ACM Computing Survey*, 7(1):80–112, Jan. 1985.
- [6] P. Grace, G. Coulson, G. Blair, L. Mathy, W. K. Yeung, W. Cai, D. Duce, and C. Cooper. GRIDKIT: Plugable Overlay Networks for Grid Computing. In *Proc. 6th OTM Int. Symp. on Distributed Objects and Applications (DOA'04)*, pages 1463–1481, Agia Napa, Cyprus, Oct. 2004. Springer-Verlag.
- [7] T. Gu, H. K. Pung, and D. Zhang. A Peer-to-Peer Overlay for Context Information Search. In *Proc. 14th IEEE Int. Conf. on Computer Communications and Networks (ICCCN'05)*, San Diego, CA, USA, Oct. 2005. IEEE CS Press.
- [8] T. Gu, H. K. Pung, and D. Zhang. A P2P Context Lookup Service for Multiple Smart Spaces. In *Proc. 4th ACM/USENIX Int. Conf. on Mobile Systems, Applications, and Services (MobiSys'06)*, Uppsala, Sweden, June 2006. ACM Press. Poster paper.
- [9] T. Gu, E. Tan, H. K. Pung, and D. Zhang. A Peer-to-Peer Architecture for Context Lookup. In *Proc. 2nd Int. Conf. on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous'05)*, San Diego, CA, USA, July 2005. IEEE CS Press.
- [10] C. Julien and G.-C. Roman. Egospaces: Facilitating rapid development of context-aware mobile applications. *IEEE Transaction on Software Engineering*, 32(5):281–298, May 2006.
- [11] B. Maniymaran, M. Bertier, and A.-M. Kermarrec. Build One, Get One Free: Leveraging the Coexistence of Multiple P2P Overlay Networks. In *Proc. 27th IEEE Int. Conf. on Distributed Computing Systems (ICDCS'07)*, Toronto, Canada, June 2007. IEEE CS Press.
- [12] A. L. Murphy, G. P. Picco, and G.-C. Roman. LIME: A Middleware for Physical and Logical Mobility. In *Proc. 21st IEEE Int. Conf. on Distributed Computing Systems (ICDCS'01)*, pages 524–533, Arizona, USA, Apr. 2001. IEEE CS Press.
- [13] O. Riva. Contory: A Middleware for the Provisioning of Context Information on Smart Phones. In *Proc. 7th ACM/IFIP/USENIX Int. Middleware Conference (Middleware'06)*, pages 219–239, Melbourne, Australia, Dec. 2006. Springer-Verlag.
- [14] S. Voulgaris and M. van Steen. Epidemic-style Management of Semantic Overlays for Content-Based Searching. In *Proc. 11st European Conf. on Parallel and Distributed Computing (Euro-Par'05)*, Lisboa, Portugal, Aug. 2005. Springer-Verlag.