A Flexible Coordination Language for Pervasive Computing Environments

Manfred Bortenschlager

Salzburg Research
Jakob-Haringerstr. 5/III
5020 Salzburg, Austria
manfred.bortenschlager@salzburgresearch.at

Abstract. Due to recent technological advances, the possibilities of pervasive computing applications have been considerably increasing. As a consequence, the dynamics, the diversity and the magnitude of entities involved in such systems are increasing, too. To address the resulting raised claim for coordination, this work proposes a two-layered coordination language. By this, a separation between programming and coordination concerns can be achieved, which shall beneficially affect the service quality.

1 Description of Purpose

This section describes the purpose of the PhD project by motivating the topic, and outlining the limitations of current systems and the significance of this work.

1.1 Motivation

Pervasive computing environments are considered as physical environments saturated with computational elements and communication technologies. The challenge is to appropriately consider the human users and their needs, and to distract them as minimally as possible. In the future, the environments where we will spend our daily lives will contain a network of specialised computing entities that will interact with the users and among themselves [12]. Besides the complexity of social structures—which must not be neglected—such as people and organisations involved and their individual preferences, a great magnitude of heterogeneous, technical entities determines pervasive computing environments. These are represented by small (or tiny), mobile and embedded devices respectively hardware, a variety of transmission technologies, and software and communication paradigms. To assist human beings in managing their daily lives with a minimal user distraction, these "smart" objects equipped with computing and communication facilities have to autonomously interact in the background. Furthermore, to provide the desired mobility of the users—i.e. their unhindered physical motion, seamless and unobtrusive support by the pervasive system has to be delivered.

In order to address this vision of pervasive computing, Satyanarayanan defined the following four research thrusts [18]: *invisibility*, *scalability*, *adaptability* and *effectiveness* (with respect to effective use of services and resources). Considering these thrusts

Tom Pfeifer et al. (Eds.): Advances in Pervasive Computing 2006
Adjunct Proceedings of Pervasive 2006, Dublin, 7-10 May
books@OCG.at Vol. 207, ISBN 3-85403-207-2

and the high number of involved entities, potential threats in system design for pervasive environments may be communication overhead, incompatibility, inconsistency, tight couplings, inflexibility, and poor modularity, reusability, exchangeability and extensibility. In order to address these threats, it appears essential to incorporate some form of coordination in order to improve pervasive computing environments. To achieve this, it seems to be promising to separate programming issues from coordination issues [5,9] and consequently, allow for a more modular and standardised way of development. Hence, the hypothesis of the PhD project is stated as followed:

> By separating the programming from the coordination concerns and for this, introducing a flexible coordination language for pervasive computing environments, the modularity, reusability, exchangeability and extensibility of coordination components shall be improved and thus, the effectiveness in developing pervasive computing systems shall be significantly increased.

In this PhD project, a flexible coordination architecture is proposed with the objective to improve the service of pervasive computing environments by addressing the above mentioned four thrusts. The "linguistic embodiment" of a coordination architecture is referred to as a coordination language [10]. The goal of this work is to eventually provide a coordination language offering the possibility to tackle coordination problems in pervasive computing.

1.2 Limitations of Current Solutions

Current solutions do not explicitly deal with the coordination issue. Usually, these solutions provide ad hoc coordination means and consequently, represent self-contained approaches which cannot be applied to other, sometimes similar problems [14]. Due to the fact that programming concerns are not separated from coordination concerns (i.e. pattern-oriented coordination models) reusability, modularity, exchangeability and extensibility can not be exploited. Hence, these proprietary solutions are far from ideal. Characteristic shortcomings resulting from inappropriate coordination would be a high number of message exchanges (i.e. communication overhead), incompatibility with other systems or protocols, inconsistency, tight couplings (e.g. in space and time) and thus inflexibility, low performance, unreliability, and finally ineffectiveness [2].

1.3 Significance of the Problem Statement

In pervasive computing systems, real world objects are monitored by sensors and the real world, in turn, is notified respectively modified by actuators [5]. The rules which determine this process and the reactions are defined within a middleware. This architectural component is responsible for the behaviour of the whole system. Hence, this would be the right place to incorporate pattern-based coordination models which can be applied to specific problems. The idea is to have standardised coordination patterns very similar to the idea of software design patterns [8] where a three-part rule expresses the relation between a certain context, a problem and a suitable solution. The proposed coordination architecture (see Section 2.2) will provide a modular way of applying coordination patterns to specific coordination problems in such environments. The inappropriate way of addressing the coordination issue in current pervasive computing systems offers a significant field of improvements. Due to the approach presented in this paper, the PhD project contributes to the pervasive computing community by incorporating the methods and means of the interdisciplinary study of coordination into the field of pervasive computing.

2 Goal Statement

The goal of this research work is to examine the hypothesis stated in Section 1.1. This will be accomplished by designing, implementing, and subsequently, evaluating a novel, flexible coordination architecture approach manifested in a coordination language which particularly focuses on the requirements of pervasive computing environments comprising heterogeneous entities.

2.1 Coordination Theory

The term *coordination* connotes a rather abstract concept where people have an intuitive sense what is actually meant [13]. The coordination issue is not only restricted to computer-based systems but is also of great significance to other fields of research like social sciences, organisational theory, anthropology, biology, sociology and many more [16]. However, it is not trivial to define this abstract concept. In [13] the term *coordination theory* was coined where this issue is considered as an interdisciplinary study. This study of coordination is concerned with properly defining the coordination concept in a domain-independent way and with elaborating generally applicable mechanisms and patterns. Hence, Malone and Crowston defined coordination as “managing dependencies between activities”. It is argued that coordination only makes sense where tasks are interdependent. In fact, it can only occur in situations where this is the case. Coordination theory suggests that standardised *coordination mechanisms* can be applied to coordination problems. As mentioned in Section 1.3, the pervasive middleware is responsible for the correct triggering of the actuators. For this, objects and their intentions have to be recognised and identified, first. Second, coordination is essential to beneficially resolve the interdependences between the occurring activities [5], which is subject to this work. The following section illustrates the proposed architecture.

2.2 Proposed System Design

According to [19], direct coordination is based on communication. Thus, this thesis will address the two major issues hindering enhanced effectiveness in pervasive computing environments—namely communication and coordination—by proposing a flexible two-layered coordination architecture:

(i) *Coordination Media:* Due to the great diversity of entities involved, the communication facets become more diverse. Other approaches than the classic client/server

(C/S) paradigm, which will not sufficiently address the aforementioned requirements, are necessary. In order to address the communication requirements of pervasive computing environments [15], the communication component of the system will be modelled according to the decentralised space-based computing approach (SBC) [3]. SBC is very similar to Linda-like systems. Processes use a commonly shared object space for communication and information exchange. Tuples can be read, written into or removed from the space. By using this simple mechanism, many problems can be addressed in a more effective way by choosing one of many feasible communication facets. The shared data objects are used for communication, too. Due to inherent decoupling mechanisms, software processes can synchronise and coordinate their activities in a much more flexible way[9]. These mechanisms are threefold: (i) *spatial* decoupling: two processes can reside in completely different computational environments and do not need to address each other as long as they know how to access the space, (ii) *temporal* decoupling: the processes do not have to be available at the same time in order to communicate (i.e. asynchronous communication), and (iii) *referential* decoupling: there is no necessity of local copies of information; this is handled by the space by implicitly providing references [4]. As opposed to the original Linda system and many Linda-like systems [7, e.g.], which assume an existing central and persistent datastore, the approach taken in this project is completely decentralised. Due to the fact that support of mobility is inevitable in the proposed system, dependence on central entities has to be avoided. Hence, the system design will adhere to the P2P concept.

The communication infrastructure of choice are the Coordinated Shared Objects (CORSO)[1] which is a P2P technology. The beneficial properties of this framework (see [3]) go far beyond the initial functionalities provided by the Linda system. The challenge will be to port the CORSO framework to small, portable devices and to adapt the framework according to the specific requirements of pervasive environments.

(ii) *Coordination Patterns:* Building up on the communication infrastructure, the second layer deals with the incorporation of several situation-dependent and exchangeable coordination patterns. As argued in Section 1.3, specific patterns will be modelled to address specific use cases. The following collection of several patterns can be found in [2]: blackboard pattern, meeting pattern, market maker pattern, and negotiating agents pattern. Further agent-based patterns (e.g. organisational structuring, multi-agent planning, contracting) are described in [16], and matchmaking and brokering in [1]. [6] describes the supervisor/worker pattern, and details about the publish/subscribe pattern are given in [17]. Moreover, insights of social sciences [11] will be evaluated, modelled, and applied to pervasive environments. These patterns will have to be investigated and the appropriate ones will be described, implemented, and incorporated into the coordination language.

[1] See TU Vienna/Institute for Computer Languages, Prof. eva Kuehn: http://www.complang.tuwien.ac.at/eva/Research/researchPublications.html

2.3 Research Issues

There are a number of research issues related to coordination theory in pervasive environments. In particular, the following will be investigated during this research project:

- Unobtrusive coordination of entities to achieve the common goal
- Resolving interdependences between activities
- Description of problem-specific coordination patterns
- Flexible communication according to external conditions
- Avoidance of dependencies from central authorities/entities

3 Approach

The approach taken to achieve the objectives of this project as stated in Section 2, can be subdivided into the following parts:

1. Pre-phase
   In this part, potential scenarios[2] will be identified and the related requirements and relevant coordination patterns will be analysed. A further targeted state of the art analysis will follow. Furthermore, appropriate test cases for following validations will be specified. Finally, a comparison methodology and reference system will be designed in order to have a means to compare coordination mechanisms.
2. Main phase
   Two iterations will be conducted in this phase of the project. The first one will comprise the software design, implementation, quantitative data acquisition and comparison, a discussion of the outcomes and a proposed incorporation into the next iteration. Consequently, the second iteration is comprised of a software re-design according to the outcomes of the first iteration, implementation of the open issues, quantitative data acquisition and comparison, discussion of the outcomes and recommendations.
3. Evaluation/discussion
   The last part will deal with an evaluation of the outcomes. The final system will be validated by experienced user groups. The overall system behaviour and the feedback of the user validation will be discussed. By using the comparison methodology designed in the pre-phase, a benchmark with other approaches and similar systems will be conducted. The behaviour of the investigated systems will be evaluated with respect to the specified hypothesis of this PhD project.

The methodology chosen will support the proper achievement of the objectives. As mentioned above, the outcomes will be demonstrated by the implementation of a software prototype and validated by users. Eventually, the outcomes of this project will be: a catalogue of specific requirements of pervasive computing environments correlated with adequate coordination patterns, a state of the art analysis of similar approach, a flexible P2P-based coordination language (design and implementation), recommendations and a critical discussion.

[2] The coordination language will be applied to event/emergency scenarios, where the involved people such as responses or guards should be unobtrusively assisted in their coordination process.

4 Acknowledgments

The author wants to thank Prof. Gabriele Kotsis (University of Linz) and Univ.-Doz. Sigi Reich (Salzburg Research) for supervision, and John Nealon (Oxford Brookes University) for his support. Furthermore, the research cooperation with Prof. eva Kuehn (TU Vienna) is highly appreciated.

References

1. K. Decker, M. Williamson, and K. Sycara. Matchmaking and Brokering, 1996.
2. D. Deugo, M. Weiss, and E. Kendall. Reusable Pattern for Agent Coordination. In A. Omicini, F. Zambonelli, M. Klusch, and R. Tolksdorf, editors, *Coordination of Internet Agents: Models, Technologies, and Applications*, pages 347–368. Springer, 2001.
3. e. Kuehn. *Virtual Shared Memory for Distributed Architecture*. Nova Science Publishers, 2001.
4. D. Fensel. Triple-based Computing. Technical Report DERI-TR-2004-05-31, DERI Innsbruck, 2004.
5. A. Ferscha. Coordination in Pervasive Computing Environments. In *Proceedings of the 12th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE-2003)*. IEEE Computer Society Press, June 2003.
6. S. Fischmeister and W. Lugmayr. The Supervisor–Worker Pattern. In *6th Annual Conference on Patters Languages of Programming*. On-line proceedings, 1999.
7. E. Freeman, S. Hupfer, and K. Arnold. *JavaSpaces Principles, Patterns, and Practice*. Pearson Education, 1999.
8. E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1995.
9. D. Gelernter. Generative Communication in Linda. *ACM Trans. Program. Lang. Syst.*, 7(1):80–112, 1985.
10. D. Gelernter and N. Carriero. Coordination Languages and Their Significance. *Commun. ACM*, 35(2):96, 1992.
11. P. H. Gulliver. *Disputes and Negotiations - A Cross-cultural Perspective*. Academic Press, 1997.
12. K. Lyytinen and Y. Yoo. Issues and Challenges in Ubiquitous Computing: Introduction. *Commun. ACM*, 45(12):62–65, 2002.
13. T. W. Malone and K. Crowston. The Interdisciplinary Study of Coordination. *ACM Comput. Surv.*, 26(1):87–119, 1994.
14. M. Mamei and F. Zambonelli. Field-based Approaches to Adaptive Motion Coordination in Pervasive Computing Scenarios. In *Handbook of Algorithms for Mobile and Wireless Networking and Computing*. CRC Press, 2004.
15. F. Mattern and P. Sturm. From Distributed Systems to Ubiquitous Computing – The State of the Art, Trends, and Prospects of Future Networked Systems. In K. Irmscher and K.-P. Fähnrich, editors, *Proc. KIVS 2003*, pages 3–25, Springer-Verlag, Feb. 2003.
16. H. S. Nwana, L. Lee, and N. R. Jennings. Coordination in Software Agent Systems. *The British Telecom Technical Journal*, 14(4):79–88, 1996.
17. D. Rosenblum. Some Open Problems in Publish/Subscribe Networking, 2003.
18. M. Satyanarayanan. Pervasive Computing: Vision and Challenges. *IEEE Personal Communications*, pages 10–17, Aug. 2001.
19. H. Weigand, F. van der Poll, and A. de Moor. Coordination through Communication. In *8th International Working Conference on the Language-Action Perspective on Communication Modelling (LAP 2003)*, Tilburg, The Netherlands, 2003.