Design and Implementation of a Generic Coordination Architecture for Mobile/Pervasive Environments

Manfred Bortenschlager
JKU Linz, Department of Telecooperation &
Salzburg Research Forschungsgesellschaft m.b.H.
+43 650 805 1982
manfred.bortenschlager@salzburgresearch.at
www.salzburgresearch.at

PhD-Supervision: **Prof. Gabriele Kotsis**

Doctoral Seminar, Linz, 2006

Abstract
Current middleware solutions combining heterogeneous entities of pervasive environments do not explicitly address the coordination issue and usually represent ad hoc and proprietary attempts. However, by separating the programming from the coordination concerns and providing problem-specific coordination patterns, the development process of pervasive computing systems can be improved through modularity, reusability, exchangeability and extensibility similar to the idea of software design patterns. Hence, this work proposes a generic two-layered coordination architecture particularly aiming at the specific characteristics of pervasive computing environments. This architecture comprises (i) the coordination media layer adhering to the space-based computing paradigm, and (ii) the coordination pattern layer proposing several reusable and problem-specific patterns.

Keywords
Coordination Theory, Pattern-based Coordination Models, Mobile/Pervasive Environments, Peer-to-Peer Networks, Space-based Computing

Problem Statement and Research Question

Pervasive computing environments are considered as physical environments saturated with computational elements and communication technologies. The challenge is to integrate the human users gracefully 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 more or less specialised computing entities that will interact with the users and among themselves [11]. 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 [17]:

(1) *Invisibility*: All the involved entities are coordinated in the background and hence, the user should be distracted as little as possible. The pervasive computing environment strives for meeting the user expectations and avoids surprising him.

(2) *Scalability*: As argued above, the number of involved entities is increasing. Thus, the frequency of interactions increases too requiring more bandwidth, energy and processor resources. A pervasive computing system has to be able to cope with scalability issues by appropriately coordinating the possibly scarce resources.

(3) *Adaptability*: Pervasive environments are characterised by dynamically changing conditions, due to the required support of mobility. The occurring, usually uneven conditions have to be masked by according mechanisms and by adapting to new situations.

(4) *Effective Use of Services*: In pervasive computing, the physical world and the abstract world of paradigms and concepts of computer science are merged resulting in smart spaces which should offer useful services to users. Here, the term "effectiveness" is used as Peter Drucker, Professor of social sciences and management, refers to it, namely as "to get the things done *right*" [4]. Hence, the opportunities arising from the convergence of physical spaces, human beings and pervasive computing have to be effectively exploited in order to provide real value-added services in an appropriate quality level.

Considering these thrusts and the high number of involved entities, 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 [6][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 generic coordination architecture for pervasive computing environments, the invisibility, scalability, adaptability and thus, the effectiveness of pervasive systems can be significantly increased.

In this PhD project, a generic coordination architecture is proposed with the objective to improve the service of pervasive computing environments by addressing the above mentioned four thrusts.

Approach and Methodology

As argued in [18], coordination is based on communication. Hence, two major issues are considered in the design of the coordination architecture leading to a two-layered approach:

Layer 1 - Coordination Media:
In order to address the communication requirements of pervasive computing environments [17][14], the coordination media component (i.e. the communication channels) of the system has been modelled according to the decentralised space-based computing approach (SBC) [10]. SBC is very similar to Linda-like systems [9]. The idea behind this paradigm is to have a commonly shared object space where processes can have access. Objects (i.e. tuples) can be read, stored in or removed from the space. By using this simple mechanism and further inherent spatial, temporal and referential decoupling [5], a great many problems can be addressed in a more elegant and effective way by having the possibility of choosing one of many feasible communication facets as opposed to the C/S approach, which always implies a direct communication. With respect to this layer the Coordinated Shared Objects (CORSO[1]) technology has been deployed.

Layer 2 - Coordination Mechanisms:
Building up on the communication infrastructure of the first layer, the second layer deals with the incorporation of several problem-specific and exchangeable coordination patterns. For this, particular patterns have been modelled to address specific use cases. Eventually, a coordination pattern catalogue is going to be established comprising the following patterns: Supervisor/Worker, Publish/Subscribe, Blackboard-based, Matchmaker, Broker and Negotiating.

The methodology to achieve the objectives of this project can be subdivided into the following parts:

Pre-phase
In this part, potential scenarios[2] and use cases will be identified and the related requirements and relevant coordination patterns will be analysed. A further extensive and 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

[1] See TU Vienna, Prof. eva Kuehn: http://www.complang.tuwien.ac.at/eva

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

designed in order to have a means to compare coordination mechanisms.

Main phase

Two iterations will be conducted in this part 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 composition of recommendations.

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 critically discussed. By using the comparison methodology and reference system 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 generic P2P-based coordination architecture (design and implementation), recommendations and a critical discussion.

Related Work

In [1], Ciancarini describes a coordination model as a triple of {E,M,L}, where {E} represents the coordinable entities which ask for coordination, {M} stands for the coordination media - representing the first layer of the proposed coordination architecture - and {L} is referred to the coordination laws defining how the interdependences have to be resolved - representing the second layer. Moreover, in [12] the term coordination theory was coined where this issue is considered as an interdisciplinary study. It 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. This definition seems to be sound and is applied to this research, too.

With respect to the first issue, initial research efforts were conducted in 1985 in the domain of parallel computing. Linda [9] was the result as the first system using a central tuple space in order to store and distribute data. It was proposed that coordination has to emphasise a high degree of decoupling among computing entities. The participants share information stored in a globally accessible, persistent data store, typically implemented as a centralised tuple space. The main difference of this system compared to the CORSO framework is the server-based architecture representing a bottleneck and hampering flexibility. CORSO, however, is a Peer-to-Peer technology with replication mechanisms, which makes it completely decentralised. Furthermore, due to the small footprint of its kernel, it is possible to deploy it on portable devices such as mobile phones, PDAs, notebooks or embedded systems. All these features make CORSO a light-weighted, highly dynamic middleware perfectly suitable for P2P and mobile ad hoc networks.

The second issue - the coordination mechanisms - touched in this paper, has also been intensively investigated within the research field of Artificial Intelligence (AI), and there particularly in agent technology [15], where four basic coordination concepts were identified: organisational structuring, contracting, multi-agent planning, and negotiation. These concepts and other research papers [3][2][7][16] about coordination mechanisms have served as a basis for the establishment of the coordination pattern catalogue which will be introduced in this PhD project.

Preliminary Results

In pervasive computing systems, real world objects are monitored by sensors and the real world, in turn, is notified respectively modified by actuators [6]. 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. Current solutions do not explicitly deal with the coordination issue. Usually, these solutions provide ad hoc coordination means and thus, represent not reusable proprietary approaches [13]. Due to the fact that programming concerns are not separated from coordination concerns (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, inconsistency, tight couplings (in space and time) and thus inflexibility, low performance, and finally ineffectiveness [3]. The idea of this PhD project is to design and implement 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 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.

Until now, a prototype implementing the supervisor/worker pattern using CORSO for the coordination media layer was realised. To simulate a pervasive environment, it was deployed on mobile devices (i.e. two Nokia 6630 and an IBM Thinkpad) which communicated via Bluetooth.

Conclusions and Future Steps

The next step will be to conduct an extensive state-of-the-art analysis where the potential benefits and drawbacks of other coordination middleware systems will be investigated. In parallel to that, the prototype will be extended by implementing more patterns and the possibility to choose between several transmission technologies (e.g. WLAN, 3G, IrDa, or RFID). Subsequently, those approaches will be compared with the proposed architecture by deploying the comparison methodology.

Feedback on the proposed architecture and methodology, refinement advices and hints on peculiarities of experts within the field of pervasive computing may be very helpful at this stage of the PhD project.

References

[1] Paolo Ciancarini, Andrea Omicini and Franco Zambonelli. Coordination Technologies for Internet Agents. In: Nordic Journal of Computing, 6(3) 215-240. 1999

[2] Keith Decker, Mike Williamson and Katia Sycara. Matchmaking and brokering. In: Proceedings of the Second International Conference in Multi-Agent Systems (ICMAS'96), Kyoto, Japan. 1996

[3] Dwight Deugo, Michael Weiss and Elizabeth Kendall. Reusable Pattern for Agent Coordination. In: Coordination of Internet Agents: Models, Technologies, and Applications, Springer, 347-368. 2001

[4] Peter F. Drucker. The Effective Executive, HarperCollins Publishers. 1993

[5] Dieter Fensel. Triple-based Computing. Technical Report DERI-TR-2004-05-31, DERI Innsbruck. 2004

[6] Alois Ferscha. Coordination in Pervasive Computing Environments. In: Proceedings of the 12th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE-2003), 3-9. 2003

[7] Sebastian Fischmeister and Wolfgang Lugmayr. The Supervisor-Worker Pattern. In: Proceedings of 6th Annual Conference on Patters Languages of Programming. 1999

[8] Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional. 1995

[9] David Gelernter. Generative communication in Linda. ACM Trans. Program. Lang. Syst. 7(1) 80-112, ACM Press. 1985

[10] eva Kuehn. Virtual Shared Memory for Distributed Architecture. Nova Science Publishers. 2001

[11] Kalle Lyytinen and Youngjin Yoo. Issues and Challenges in Ubiquitous Computing: Introduction. Commun. ACM, 45(12) 62-65, ACM Press, New York, US. 2002

[12] Thomas W. Malone and Kevin Crowston. The interdisciplinary study of coordination. In: ACM Comput. Surv., 26(1) 87-119, ACM Press. 1994

[13] Marco Mamei and Franco Zambonelli. Field-based Approaches to Adaptive Motion Coordination in Pervasive Computing Scenarios. Handbook of Algorithms for Mobile and Wireless Networking and Computing CRC Press. 2004

[14] Friedemann Mattern and Peter Sturm. From Distributed Systems to Ubiquitous Computing -- The State of the Art, Trends, and Prospects of Future Networked Systems. In: Proceedings of KIVS 2003, 3-25, Springer-Verlag, 2003

[15] Hyacinth S. Nwana, Lyndon Lee and Nicholas R. Jennings. Coordination in Software Agent Systems. In: The British Telecom Technical Journal, 14(4) 79-88. 1996

[16] David Rosenblum. Some Open Problems in Publish/Subscribe Networking. Keynote, 2nd International Workshop on Distributed Event-Based ystems (DEBS'03). 2003

[17] Mahadev Satyanarayanan. Pervasive Computing: Vision and Challenges. IEEE Personal Communications, 10--17. 2001

[18] Hans Weigand, Frank van der Poll and Aldo de Moor. Coordination through Communication. In: Proceedings of 8th International Working Conference on the Language-Action Perspective on Communication Modelling (LAP 2003), 115-134. 2003