Configuration of Applications for the 3rd Generation Mobile Communication

Roman Englert[1] and Armin B. Cremers

Institute of Computer Science
Rheinische Friedrich-Wilhelms-Universität Bonn
Germany
{englert|abc}@informatik.uni-bonn.de

Abstract. In the context of UMTS mobile applications require the consistency of information, and the configuration of interfaces and applications. These requirements can be depicted by subtle problem fields as they are described in this paper. In the mid 90th multi-agent systems (MAS) have been invented into the field of mobile applications. The desired alleviation of the problem fields can be achieved by the expansion of MAS by planning. Preliminary first ideas of these methods are discussed for mobile applications. A simple prototypical scenario for the configuration of mobile interfaces demonstrates the promising combination of MAS and planning.

1 Introduction

In mobile communication systems exist a heterogeneous landscape of architectures and technologies. Despite the standardization efforts for network and application technologies, namely the 3rd generation mobile communication standard UMTS for the mobilization of the internet [1] and CAMEL, the connecting technology for mobile communication networks[2], the endeavors resulting from the different architectures and technologies are standing. Applications are embedded in varying environments due to hand-overs, roaming, etc. These environments, like the *virtual home environment* (VHE) of users, can be described by parameters, e.g. signalling is activated or roaming is enabled. Another important challenge emerges from the four *Quality of Service* (QoS) classes of UMTS: conversation, streaming, interactive, and background class. These QoS classes enable one to implement a variety of mobile applications and thus lead to a huge number of services satisfying all user necessities.

Mobile communication applications synthesize the compatibility of different architectures and technologies, and additionally, the handling of a huge number of different services for users. Both challenges can be characterized as follows: First, the compatibility of different technologies depends on the system capability to adapt configurations of applications and interfaces onto each other. Note, in general users

[1] Please contact the first author
[2] Customized Applications for Mobile Network Enhanced Logics [1]

cannot assist in the configuration process due to the complexity of technical details. Thus, this procedure has to be performed automatically without any user assistance. Second, due to a broad offer of services users may apply to several applications in parallel. Therefore, the management of several applications on the same bearer must be possible with regard to resources like the network capacity.

In the mid 90th *multi-agent systems* (MAS) have been invented into mobile communication in order to enable a flexible execution of mobile applications [3]. The abilities of MAS can be described briefly by the following statements [4]: "…gives you just what you need, works while you don't, works where you aren't…". In this sense agents are mobile, cooperative, able to negotiate, and unbounded, to name a few important properties resulting in action competence, capability to handle different resources as they occur in the telecommunication and different service ontologies, and finally, the prevention of problems by acting pro-active versus reactive.

How to alleviate the above depicted challenges? The core thesis depicted in this paper is based on the assumption that planning methods known from the field of Artificial Intelligence can be integrated into the execution of mobile applications based on MAS. Hence, planning facilitates the solution of conflicts like ambiguities. In the following a brief introduction into the UMTS standard is given with focus onto MAS, and the four UMTS QoS classes are described (Sect. 2). Then, two subtle problem fields of 3rd generation mobile communication systems are described resulting from the heterogeneous architectures and technologies within mobile communication (Sect. 3): the configuration of interfaces and applications, and the execution of mobile applications with inconsistent information like distributed calendars. Planners are briefly depicted in the subsequent Sect. 4 with a simple example of mobile applications (when familiar with planners readers may skip this section). In Sect. 5 first ideas for the examination of the problem fields are described. A preliminary prototypical solution based on the STRIPS planner Blackbox [9] for the problem field concerning the configuration of applications and interfaces is provided. We conclude in Sect. 6 and discuss related work for MAS.

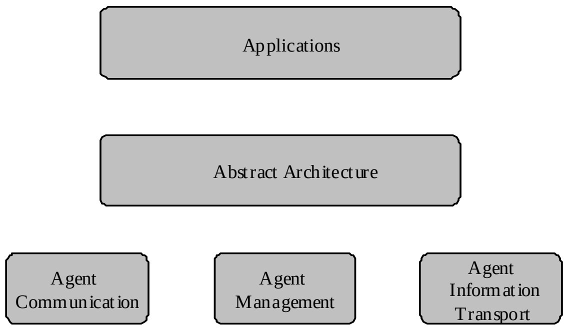
2 UMTS Standard and Quality of Service Classes

In this section a brief introduction into UMTS (universal mobile telecommunication standard) is given[3]. UMTS is known for its higher bit rates (packet-switched up to 2 Mbps) compared to GMS. Higher bit rates facilitate new services like video telephony and quick downloading of data. Furthermore, UMTS allows a negotiation of the properties of a radio bearer. The properties are characteristics of the data transfer like throughput, transfer delay, etc. At the start of the UMTS era the overriding traffic will be voice, but later the share of data will increase. Often the requested information is available on the internet, which demands for effective handling of TCP/IP in the UMTS network.

[3] A comprehensive description of UMTS is out of the scope of this paper and can be read in [1, 17, 18], where the latter reference contains also the specification for applications and services.

For the development of mobile applications exist several platforms and de facto standards like GRASSHOPPER [13] or the Open Service Architecture (OSA)[4]. For a moment we will focus on OSA which specifies the application programer interfaces (API) for 3rd generation mobile applications: The technical infrastructure of networks is today non-standardized resulting in several implementations of an application for different networks. This obstacle leads to a slow-down in the application development process. To overcome this hinderance the telecommunication industry has begun to standardize the *intelligent network* (IN) platform [19]: Services like prepaid and conditional call forwarding are realized on this platform. However, the development of applications on the IN platform requires knowledge about the telecommunication infrastructure and the used protocols. Alleviation is brought by OSA which standardizes APIs for call control, location information, etc. Additionally, OSA provides a specification for a network interface that provides information about network capabilities and services[5].

Fig. 1. Layer model for agents.



In the telecommunication world applications and data are distributed in general [6]. An example is a calendar which exists in several mobile devices and additionally, in the mobile network where the data are collected. Due to the above described heterogeneous network infrastructure specific requirements to agents are emerging (Fig. 1): on top the (mobile) applications, in the middle the abstract architecture, and at the bottom the agent communication, management, and information transport [20]. In the following we will focus on the agent information transport which contains complex communication of agents like negotiation and monitoring. The UMTS standard provides no specification about agents. But the platform GRASSHOPPER [13] contains an API for the bottom layer of Fig. 1 which makes the management of the Virtual Home Environment (VHE) parameters of mobile users

[4] 3GPP TS 23.127 [18]
[5] Services are network functions provided to users and developers, e.g. location information.

possible. The VHE concept for UMTS was introduced in the standardization process for the provision and delivery of personalized services across network and terminal boundaries with the same look and feel for users.

Unfortunately, GRASSHOPPER assumes that agents know network properties like throughput, etc. These characteristics require a global view of agents which in general is not present. This shortcoming can be eased through the invention of a so-called *Meta Agent* (MeA). The MeA knows the network properties and has the information about all agents (status, type, etc.).

Finally, we finish these considerations with a description of the four UMTS QoS classes: conversational, streaming, interactive, and background class. The main factor distinguishing these classes is how delay-sensitive the traffic is: the conversational class is meant for very delay-sensitive, while the background class is the most delay-insensitive. Real-time conversation is characterized by a low end-to-end delay (less than 400ms). Streaming is a technique for transferring data in such way that it can be processed as a steady and continuous stream. In the interactive class either a human or a machine requests on line data from remote equipment. Finally, in the background class data are delivered generally with delay, since for these data no immediate action is required.

3 Subtle Problem Fields of Mobile Communication Systems

UMTS leads to a huge variety of mobile applications. The use of these applications will be accompanied by two subtle problem fields resulting from the heterogeneous architectures and technologies within mobile communication: the configuration of interfaces and applications, and the execution of applications with unknown interface configurations. In the following both problem fields are depicted schematically in an own section continuing with a description, an example, and a brief discussion. Finally, each problem field is related to the four UMTS QoS classes [1].

3.1 Incompatible Configuration of User Interface Agents

The execution of applications in mobile devices requires a data exchange with the network via an *over-the-air* (OTA) interface. Both, the network and the mobile device need the *user interface agent* (UIA) which performs the data exchange and the communication with each other. Often the situation arises where the configuration of both UIAs is incompatible. To solve this problem the agents have to be configured, or more precisely the network UIA must be adapted to the configuration of the UIA in the mobile device. This can be done by choosing an UIA in the network with a configuration compatible to the one in the mobile device.

In general agents do not know anything about each other [5]. Hence, an agent is needed which knows the features of all other agents in the network. This agent will be called *meta agent* (MeA). The MeA in the network enables UIAs to ask for an UIA with specific features, i.e. in this case for an UIA with a compatible configuration. The example in Fig. 2 depicts this situation: Suppose the user adds a date to the local calendar of the mobile device. Afterwards the calendar application activates an UIA for the subsequent synchronization process (1). Yet, the UIA is incompatible to the one in the network (2). Thus, in step 3 the MeA is demanded to provide a compatible UIA. The MeA selects a corresponding UIA (4) and activates it in step 5. If necessary the new UIA must be equipped with additional information (6), e.g. parameters about the UMTS network like throughput, and finally, is sent to the interface of the network in order to correspond with the UIA of the mobile application.

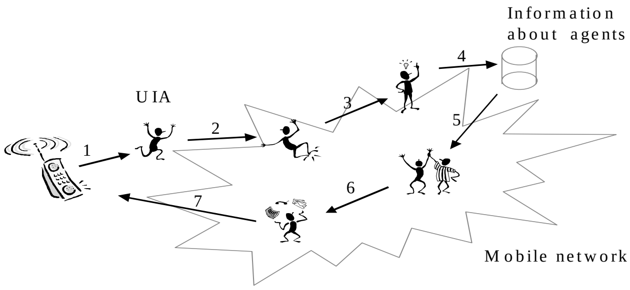


Fig. 2. Incompatible configuration of user interface agents.

Note, the configuration of the agent must be performed during the execution of the application without any assistance of the user. The configuration process is complex, since the resources can be distributed within the network. This motivates the use of planning in order to ease and stabilize the execution. However, agents have in general only a local view of their environment, i.e. they totally lack interrelated knowledge. The MeA solves this problem, when the status of each agent is registered in a database of agents.

The configuration of interfaces and applications is fundamental for all four UMTS QoS classes: conversation, streaming, interactive, and background.

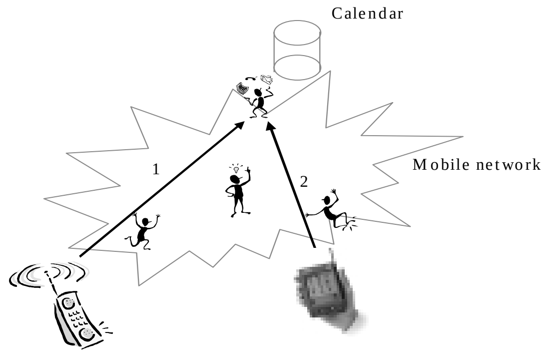
3.2 Inconsistent Information

Mobile telecommunication systems are distributed systems [6]. In general they contain in general various databases, which may lead to inconsistencies, e.g. data in an user calendar are distributed to a central calendar in the network and local calendars in mobile devices. An example is depicted in Fig. 3: Suppose a date is added to several mobile devices, e.g. a mobile phone and a PDA[6], at different points of time. For the synchronization both calendar applications send information agents to the central calendar in the mobile network (1,2). The inconsistency due to the overlapping points of time must be analyzed and solved: After a negotiation of the information agents with the display agents of the calendar applications in the mobile

[6] Personal digital assistant

devices (the latter may involve the user) one date will be rejected and the corresponding calendar must be updated, or both calendars are marked with an inconsistent date.

Fig. 3. Inconsistent data due to distributed calendars in several mobile devices.



Note, the synchronization of both calendar applications can also be done at different points of time. In this situation the central calendar in the mobile network must log source parameters of the dates, e.g. mobile device and time of adding. Furthermore, inconsistencies can be due to time and semantics. The latter requires for its solution a huge data modeling and inferencing effort [7].

Calendar systems demand that inconsistencies should be solved as soon as possible. However, in mobile applications to achieving this goal can be difficult since network resources are limited, e.g. the network can be overloaded during the activation time for the synchronization.

Inconsistent data may occur in two important UMTS QoS classes: Conversation and interactive. The streaming and background classes may also transport inconsistent data. But, both classes incorporate no interaction with the user.

4 Planning to Diminish the Problem Fields

Planning in the area of mobile communication applications has several motivations: First, resources often are distributed and thus the aggregation of data requires several actions. Second, inconsistent data lead to negotiation and the coordination of several parties. Finally, bearer have a limited number of channels, which can require the execution of several actions via one channel. These problems need planning of actions, resp. scheduling of actions. In the following the planner Graphplan [4], and its data representation are described. An example is given for agents transporting information.

Graphplan is a compiling planner based on STRIPS like domains [8]. Data are represented as a graph and a plan is computed by propagating constraints through the graph. The graphs can be computed in polynomial time. However, STRIPS planning is PSPACE hard. Fortunately, there is empirical evidence for planning problems providing a promising perspective [4].

As an example agents with limited resources are considered. There are three operators: agent LOADs information, agent TRANSFERs information, agent UNLOADs information. Furthermore, an agent can only transfer information, when resources are still available, and the transfer decreases its resource. The operator TRANSFER can then be expressed as

```
(define (operator transfer)
     :parameters ((agent ?a) (loc ?start) (loc ?goal))
     :precondition ((:neq ?start ?goal)(resource ?a))
     :effect((reached ?a ?goal) (:not (resource ?a))))
```

A typical planning task for this domain has several agents and several data items at the starting point. The data have to be transferred to the goal. Recall the synchronization example in Sect. 3.2 for which we assume that there is only one channel available: starting point is the central calendar in the mobile network, data items are inconsistent dates, which are to be represented on mobile devices, and the goal are different mobile devices to which the transfer of the inconsistencies must be performed.

Many actions are possible for mobile applications: Some important actions for agents are the request for compatible configurations, the transfer of information, the status modification of applications / agents (wait, active, …), negotiation with users / agents, and the monitoring of actions. These actions can be part of plans which must be computed during the execution of the mobile application with regard to distributed resources, the availability of agents, etc. In Sect. 5 some open questions for the described planning framework and their examination are discussed.

5 How to Examine and Solve the Problem Fields

In this section planning is described for the examination and solving of the above depicted problem fields for mobile communication applications: planning compilation, an exemplary scenario, and open questions for this planning problem are considered

Graphplan [4] was the first compiling planner. However, due to performance reasons BlackBox [9] compiles onto a SAT representation and then uses a fast SAT solver. This planner gives empirical evidence for real problems as well as Graphplan does [4]. In the following the problem field described in Sect. 3.1 - incompatible configuration of user interface agents - is applied to BlackBox. The task is to find a plan, where one or more of the following situations can occur:

1. The UIA of the mobile application and of the mobile network are both incompatible. Thus, the MeA must be asked for a new UIA which has to be sent to the mobile application.
2. None UIA can be transferred within the mobile network. This situation evolves due to limited network resources.
3. Both former conditions are not given and hence, no new UIA is required.

4. Suppose an user starts *n* applications with *n* incompatible UIA. Consequently, *n* new UIA are required. If only *n-1* UIA are available, then one application cannot be executed immediately.

The latter problem has several solutions. The principle *first come first serve* can be applied, or more sophistically, the execution of applications can be priorized. For reasons of demonstration the first case is assumed. The domain has seven actions, which look as follows (compare with the plan below):

```
(:action conf-incompatible
         :parameters (?a ?a_f)
         :precondition (and (activation-checked ?a)
            (activation-checked ?a_f)
            (not (= ?a ?a_f)))
         :effect (not (compatible ?a ?a_f))
 )...
(:action inform-user
         :parameters (?a ?a_f)
         :precondition (and (not (compatible ?a ?a f))
(sent-to-user ?a) (set-agent-wait ?a))
         :effect (user-informed ?a)))
```

The first action checks whether two agents carry compatible configurations, with the precondition that the agents are activated. Note, the effect includes the closed world assumption. The second action states that the user must be informed for a delay in the execution, when the configurations are incompatible and thus a new UIA must be selected. The used facts look as follows:

```
(:init (sent-candidate agent-user)
       (wait-candidate agent-user)
       (activated agent-user)
       (activated agent-network)
       (uia-exist agent-user agent-by-meta)
 ... )
```

For example, the first fact states that the agent `agent-user` can be sent to a mobile device. The goal can then be formulated as

```
(:goal (execute-application-with-agent agent-user))
```

which leads to the following plan

```
(set-agent-status-to-wait agent-user)
(set-message-to-user agent-user)
(check-activation agent-user)
(check-activation agent-network)
(conf-incompatible agent-user agent-network)
(inform-user agent-user agent-network)
(request-meta-agent agent-user agent-by-meta)
(sent-agent-to-mobile agent-by-meta agent-user)
(execute-application agent-by-meta)
```

As this simple example demonstrates, the generated plan is short. However, in mobile communications millions (approximately 40-60) users are expected with the need to generate plans. Positive, this prototypical example gives evidence for the applicability of planners to the described problem field. However, several core questions remain:

1. Real-time: Can planning be applied to the configuration during the execution of mobile applications for UMTS? In general users are only waiting a couple of seconds for an answer, and most of this time is required for the execution of network operations.
2. Interactivity: Can agents negotiate with users in order to acquire more information before/ during the planning process [15]? A comprehensive example for the performance planning of elevators is given in [21].
3. Scaling: How many entities may be contained in a planning process (see above)?
4. Solvability: Are planning domains of mobile applications solvable? Due to reasons of usability users must be served within a specific time limit.
5. Scratch planning: Exist identical subsequences of actions in different plans which enable one to build a library of partial plans [16] or is planning from scratch without a library more appropriate for the execution of mobile applications?

6 Related Work and Conclusion

A large number of multi-agent systems exists [5]. First applications from the mid 90th are e-mail filters and entertainment recommendations [12]. The former agents have the task to priorize, to delete, to forward, and to archive e-mails based on a technique similar to memory-based reasoning. Whereas in the latter application agents select cinema films, books, etc. from different users. The core idea is that users preferring similar books also prefer similar videos (correlation). MAS have also been applied to telecommunication applications with distributed resources [2]. Unfortunately, the above described challenges (Sect. 2) cannot be solved with existing MAS since they do not incorporate adaptivity of e.g. interfaces to the varying environment of mobile applications. Adaptivity of AI systems can be reached e.g. by planning. However, to the authors' knowledge no planner exists today for mobile applications that incorporates capabilities of agents like negotiation into the planning process. Current examination is the extendibility of planners by MAS for the solution of conflicts like inconsistent information. Subtle problem fields for these tasks have been identified, related to the four UMTS QoS classes and described with examples. These challenges will be applied to planning with MAS. A simple example with the STRIPS planner BlackBox has been shown. In a next step, this domain will be converted to ADL (PDDL+) and applied to a modern planner like FF or Talplanner. Additionally, this example must be refined by CAMEL and IN technology in order to migrate mobile applications from 2.5G to 3G (for a brief introduction to MAS and the migration we refer to [13]). The open questions depicted in Sect. 5 are in the main focus of our current research.

References

1. Holma, H./Toskala, A. (eds.), WCDMA for UMTS - Radio Access for 3rd Generation Mobile Communications, Wiley & Sons, 2000
2. Jennings, N./Wooldridge, M. (eds.), *Agent Technology: Foundations, Applications, and Markets*, Springer-Verlag, Berlin, Germany, 1998
3. Shneiderman, B., *Looking for the Bright Side of User Interface Agents*, in ACM Interactions, January, pp. 13 – 15, 1995
4. Blum, A./Furst, M., *Fast Planning Through Graph Analysis*, Artificial Intelligence, Vol. 90, pp. 281 – 300, 1997
5. Bradshaw, J. (ed.), *Software Agents*, MIT Press, Boston, U.S.A., 1997
6. Busuioc, M., *Distributed Intelligent Agents – A Solution for the Management of Complex Telecommunications Services*, Chapter 4 in Hayzelden, A./Bigham, J. (eds.), Software Agents for Future Telecommunication Systems, Springer, 1999
7. Englert, R., *Learning Integrity Constraints*, 21st Conference on Artificial Intelligence, Springer, Germany, 1995
8. Nilsson, N., *Principles of Artificial Intelligence*, Series Symbolic Computation, Springer-Verlag, Berlin, Germany, 1982
9. Kautz, H./Selman, B., *Planning as Satisfiability*, Proceedings of the 10th European Conference on Artificial Intelligence, Austria, 1992
10. Ripley, B., *Pattern Recognition and Neural Networks*, Cambridge University Press, UK, 1996
11. Englert, R./Rosendahl, T., *Customer Self Services*, Kapitel 3.1.5 in *E-Commerce*, Weiber, R. (ed.), Gabler, Wiesbaden, Germany, 2000
12. Maes, P., *Agents that Reduce Work and Information Overload*, Communications of the ACM, Vol. 37, No. 7, pp. 31 – 40, 1994
13. Farjami, P./Görg, C./Bell, F., *Advanced Service Provisioning Based on Mobile Agents*, Computer Communications, Special Issue: Mobile Software Agents for Telecommunication Applications, Vol. 23, pp. 754 - 760, 2000
14. Jing, J./Helal, A./Elmagarmid, A., *Client-Server Computing in Mobile Environments*, ACM Computing Surveys, Vol. 31, No. 2, pp. 117 – 157, 1999
15. Perini, A./Ricci, F., *An Interactive Planning Architecture*, in Ghallab, M./Milani, A. (eds.), New Directions in AI Planning, Vol. 31 in Series Frontiers in Artificial Intelligence and Applications, pp. 273 -- 283, IOS Press Omsha, Amsterdam, NL, 1996
16. Jonsson, P./Bäckström, C., *Incremental Planning*, in Ghallab, M./Milani, A. (eds.), New Directions in AI Planning, Vol. 31 in Series Frontiers in Artificial Intelligence and Applications, pp. 79 – 90, IOS Press Omsha, Amsterdam, NL, 1996
17. UMTS forum: www.umts-forum.org
18. 3GPP Third Generation Partnership Project: www.3gpp.org
19. European Telecommunication Standards Institute: www.etsi.org
20. Finin, T./Labrou, Y./Peng, Y., *Mobile Agents Can Benefit from Standards Efforts on Interagent Communication,* IEEE Communications Magazine, Vol. 36, No. 7, pp. 50 – 56, 1998
21. Köhler, J., *Von der Theorie zur Praxis: Verkehrsplanung für Hochleistungsaufzüge*, Künstliche Intelligenz, Heft 2/01, S. 25 – 30, 2001