

Mobile Context Aware software and MAS

Raian ALI ali@dit.unitn.it

April 22, 2007

Abstract

The vision of what is called nomadic, ubiquitous, pervasive, or wearable computing is becoming more than a fantasy. The advances in communication technology (wireless internet, Bluetooth, WiFi) and the increasing power of handheld devices (3G mobile phones, PDA) have given a great pulse to get further step in this direction. Exploiting these innovative technology is still in its infancy phase. Software development models and methodologies were made for what we can now call Desktop software. But for the next generation software, it is foreseen that software becomes personalized, context aware, autonomous, instead of being statically defined a priori.

Agent Oriented Computing and Multiagent systems (MAS) address many aspects regarding the vision of mobile, context-aware, personalized, and social software. The two areas of research tend to use very similar terms, and specially those of software adaptation, autonomy, flexibility, and situatedness. In this proposal we try to check the usefulness of agent oriented paradigm when it is employed for developing this kind of software, with special attention to the context aware capability.

1 Introduction

Desktop computers, or laptops even after the introducing of wireless internet access and other communication technologies, are no longer the only ways for accessing, spreading, and generating information. The degree of exploitation of the current technology advances is challenged by the current software development methods, besides the lack of models and experiences and the imagination about how the new software should or can look like. One of the interesting areas in research today is to achieve the computing called some times: Anytime Anywhere, ubiquitous, pervasive computing, ambient intelligence, and CSCW which theoretically can be achieved. In this vision, mobility is an essential dimension, and to be mobile means that something around changes. Context, in its general meaning, is meant to be considered as an input implicitly provided, sensed, derived, and of course it might be manually provided.

Context was defined differently by several research groups, and there is no consensus about what makes part of the context and what does not. One

general definition can be found in [2]: "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between user and application, including the user and application themselves".

Getting context was classified also in different ways. In [4] the attempt was to model the context information representing entities as objects and associating context information to them. Context was subdivided into two main categories. The first one is the static context, which refers to a property that is fixed regarding one entity, like for example: device type. The second is the dynamic that covers all the rest of context information. It is subdivided into three categories, the first is the sensed (like user location, temperature degree). The second is the derived that refers to a context derived using a derivation function. An example of the derived context is the Is Located Near relationship which is derived from two sensed location contexts, the coordination parameters. The third subcategory is the profile context that covers information supplied by user, for example user name and the name of people user is working with. The classification of context that is done in [8] uses 3D model to classify context. Context is divided into three parts:

- Self: e.g. device state, physiological, cognitive
- Environment: the physical and the social environment.
- Activity: the behavior and task being done.

The mobility is strongly related to the context change, the frequent common changes that have potential ability to affect software task are listed by [3] as shown in the following list, which is one of the most concrete and comprehensive lists of context information classification:

- The *spatio-temporal context* describes aspects related to time and space. It contains attributes like time, location, direction, speed and track.
- The *environment context* captures the entities that surround the user, for example, physical objects, services, temperature, light, humidity, and noise.
- The *personal context* describes the user state. It consists of the physiological and the mental contexts. The physiological context may contain information like pulse, blood pressure and weight, as well as personal abilities and preferences. The mental context may include elements such as mood, expertise, anger, and stress
- The *task context* describes what the user is doing. The task context may be described with explicit goals or the tasks and task breakdown structures.
- The *social context* describes the social aspects of the user context. It may, for instance, contain information about friends, neighbors, coworkers and relatives. The role that the user plays is an important aspect of the social

context. A role may describe the user's status in this role and the tasks that the user may perform. The term social mobility refers to the ways in which individuals can move across different social contexts and social roles and still be supported by technology and services.

- The *information context* describes the information space that is available at a given time.

Context awareness system was defined in [2] in this form: "*A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user task*".

From the other hand agent computing is a hot topic in academic research of today, it is motivated by the need for a software can act on behalf of the user, software that is able to implements the concept of agency. A well accepted definition of agent is found in [10]: "*An agent is an encapsulated computer system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives*."

A multi-agent system (MAS) is known as a system composed of several agents, collectively capable of reaching goals that are difficult to achieve by an individual agent or monolithic system. In the next section we discuss the possible power of agent and multi agent system when they are employed for developing mobile context aware software.

2 Context Awareness and Agent Oriented Computing: is there a coincidence?

First of all, we noticed while reading the literature of mobile context-aware computing, sketched above, the use of the same terminologies and concepts used in agent literatures. In both of these research areas, the talk is about being adaptive, flexible, autonomous, and even social according to the situation, environment or (context) you are in. It becomes common trend in Agent Oriented Software Engineering to consider environment as a first class-entity [11] [5]. The environment is defined as all the non-agents elements . Such elements typically include DB, Web services, communication infrastructure, and physical devices. From the perspective of mobile context aware software there are other factors that contribute formulating the environment, those factor are included in the definition of context. Current practice in MAS considers the environment as an implicit part of the MAS that is often dealt with in ad hoc way. But the need to consider environment is more apparent when we talk about a multi agent system that is going to run in different contexts factor associated or independent of agent or other agents in contact also. When considering a MAS as made of a set situated agents, and when we talk about nomadic, context aware, socially and physically mobile software, it makes sense that software knows about the environment in which it has to be situated. This environment

covers a variety of context concepts (or vice versa), such as the social context, the personal context.

The interaction amongst agents is done according to the context each agent has, and the environment they are situated in which makes a part of the context. This interaction in turn affects the environment. The real problem is that the context is dynamically changing, and to model dynamicity is not almost easy task.

MyCampus [1] is an interesting project being developed in Carnegie Mellon University; it is intended to build an open agent oriented environment for context aware mobile computing within the student campus. Students might use their mobile devices and might pull agents from the context server. Each agent has a specific task, such as organizing a meeting, and each user has profile and changing context. The dynamicity here is handled by using ontologies for describing context, user preferences and profile; this makes it possible developing agents that can automatically understand user context and preferences.

3 Proposal for a research

In spite of the exciting nature of the mobile context aware software, most of the work is still until now inside the boundary of research projects. There is no real commercial software of this type. There is a need for a step further; this step involves new techniques coming from different areas. What we are arguing is that agent and multi agent systems are natural way of thinking when we talk about mobile, context aware software development. Of course there is a need for many other research areas such as logic for deriving and inferring context, modeling language for modeling context. But we see those as contributions to the main software engineering methodology that uses agent theory.

The current research in its major part is focusing on how to build infrastructure for getting related context information, what kind of sensors and communication technology we need, how the connection can be stable and how to exploit better the bandwidth, how to get logically context information, and how to derive them. But very few research handle the problem of modeling context, modeling interaction amongst users upon different context, modeling the changing business process route when context changes.

There are attempts to adapt existing modeling techniques to be able of modeling context and provide a base for context aware software such as the work [4], and the work done in [6] that suggest using colored Petri nets as a promising technique which is known for its ability for modeling the dynamicity. Another work is found in [7] that try to adapt conceptual graph to represent context. Other work [9] proposes the Context Modeling Profile as a lightweight UML extension. To our mind, Mobile Context Aware software needs more than adapting the existing work which was not done with the mobility and context awareness in mind.

To our best knowledge, there is no software engineering methodology was natively done for mobile, context aware software development. There are some

work done on pieces of the development process such as simulating context awareness, or modeling context as mentioned above, but no integrated work is until now done.

We are anticipating that agent computing and agent oriented software development can do well when used for Mobile Context aware software. Tropos, as an agent oriented software engineering methodology, as well as other agent and non agent software methodologies, was not natively made for such kind of software. So the research is for sophisticated methodology based onto agents for developing such kind of software,

The proposed research is to think about mobile context-aware software using the mentality of autonomous agents and multi-agent system. The employment of agent can be done on all the level of software development process from the early phases of analysis until implementation using agent oriented programming.

The initial research agenda can be:

1. The new features mobile context-aware have compared with traditional one.
2. How can current software engineering state address the new challenges come with this kind of software.
3. How current modeling languages can fit to represent the context information.
4. Agent Oriented computing, is it the adequate for developing mobile context aware software?
5. What kind of Agent models do we need to cover all aspects of mobile context aware software?
6. How agents might be specified, and then transformed into efficient implementation, and then verified.

As we foresee, mobile context aware software is a fruitful areas to be investigated using agent oriented computing from both of software engineering and development perspectives.

References

- [1] <http://www.cs.cmu.edu/~sadeh/mycampus.htm>.
- [2] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *HUC*, pages 304–307, 1999.
- [3] John Krogstie et al. Research areas and challenges for mobile information systems. *International Journal of Mobile Communications 2004 - Vol. 2, No.3 pp. 220 - 234*, 2004.
- [4] Karen Henriksen, Jadwiga Indulska, and Andry Rakotonirainy. Modeling context information in pervasive computing systems. In *Pervasive*, pages 167–180, 2002.
- [5] Nicholas R. Jennings. On agent-based software engineering. *Artif. Intell.*, 117(2):277–296, 2000.
- [6] Oh Byung Kwon. Modeling and generating context-aware agent-based applications with amended colored petri nets. *Expert Syst. Appl.*, 27(4):609–621, 2004.
- [7] SuTe Lei and Rang Zhang. Mobile context modelling using conceptual graphs. In *WiMob'2005*, 2005.
- [8] Albrecht Schmidt, Kofi Asante Aidoo, Antti Takaluoma, Urpo Tuomela, Kristof Van Laerhoven, and Walter Van de Velde. Advanced interaction in context. In *HUC*, pages 89–101, 1999.
- [9] Christof Simons. Cmp: A uml context modeling profile for mobile distributed systems. In *HICSS*, page 289, 2007.
- [10] Michael Wooldridge. Agent-based software engineering. *IEE Proceedings - Software*, 144(1):26–37, 1997.
- [11] Franco Zambonelli and Andrea Omicini. Challenges and research directions in agent-oriented software engineering. *Autonomous Agents and Multi-Agent Systems*, 9(3):253–283, 2004.