Context-aware applications using personal sensors

Julien Subercaze
INSA de Lyon
LIRIS (CNRS UMR 5205)
20, avenue albert einstein
F-69621 Villeurbanne, France
julien.subercaze@insa-lyon.fr

Pierre Maret
INSA de Lyon
LIRIS (CNRS UMR 5205)
20, avenue albert einstein
F-69621 Villeurbanne, France
pierre.maret@insa-lyon.fr

Ngoc Minh Dang
Department of Human and Engineered Environmental Studies
Graduate School of Frontier Sciences, The University of Tokyo.
Kashiwa, Chiba, Japan

Ken Sasaki
Department of Human and Engineered Environmental Studies
Graduate School of Frontier Sciences, The University of Tokyo.
Kashiwa, Chiba, Japan
ksasaki@k.u-tokyo.ac.jp

ABSTRACT
Context-aware applications require the development of convenient frameworks. Effective mobility requires that mobile applications can integrate new sensors or new types of information. This is not possible within traditional applications, because a re-design phase in necessary. We describe in this article an agent-based framework supporting sensors' data fusion and context-aware information exchanges. An ontology-based representation of data is used. Exchanges in-between components are carried out within so-called virtual knowledge communities. An application has been designed within this framework ('Wake me up', taking place in the metro). It makes use of wearable sensor, transmitters and cell phones. The wearable sensor is used to determine the state of a user. Transmitters provide geographical information, and cell phones are used as personal assistants.

Keywords
Context-aware applications, personal sensors, virtual knowledge communities, ontologies, pervasive computing

1. INTRODUCTION
Context-aware applications are based upon information captured from the environment and from the user. The user can also be considered as belonging to the environment of the application. The context-information is delivered by sensors (environmental or personal) which are fixed or mobile. Applications, which are also fixed or mobile, gather information from the sensors. Such applications require the development of convenient frameworks.

The integration of information delivered by sensors is a big challenge, especially in the frame of mobile applications. Indeed, the types of sensors are heterogeneous, and the use of collected information from sensors must be pre-defined, while sensors and applications have not been designed coordinately. This is actually a crucial difficulty in the development of mobile context-aware applications.

Today, most of the frameworks are built for static context-aware applications. Most advanced frameworks are: [9] concentrating on personal profiles, [5] dealing with the integration of multiple device sensors, [4] focusing on sensor fusion based on bayesian network, [6] presenting an useful toolkit to quickly develop pervasive applications centered on context modeling, [10] considering the context-history of the user. The main drawback of these frameworks is that none of them provides a general solution for a soft integration into the applications of new-coming data types from new-coming sensors. In most cases, applications must be re-designed.

In this paper, we describe an agent-based framework supporting sensors' data fusion and context-aware information exchanges (section 2). Section 3 presents an application which was designed within this framework and which is based on a multi-agent approach. Our application makes use of wearable sensors, transmitter and cell phones. The wearable sensor is used to determine the state of a user. Transmitters provide geographical information, and cell phones are used as personal assistants. We conclude the paper in section 4.

2. FRAMEWORK

Although the use of sensors in context-aware applications is necessary and also well popular, there is a lack of generic and adapted infrastructure for developing (or creating) such applications. Besides, in order to fully incorporate the data provided by sensors, not only the infrastructure (architectural issue) must be taken into account, but also the information exchange issues, as well as the data processing and context analysis for semantic interpretation. In order to better considerate these points and to promote context-aware applications, we propose an infrastructure (depicted in the figure 1) with three key concepts:

- Context-aware applications require emphasizing the components' autonomy. In software-engineering this is compliant with an agent-like approach where autonomy and decentralisation are keystones.
- We define two types of agent: Context Agent, responsible for contextual knowledge capture and dissemination (section 2.1), and the Personal Agent which possesses knowledge about its own user (the users profile) and which is in charge of delivering recommendations to the user (section 2.2);
- Agents are provided with ontology-based representation of data. Exchanges are done using Virtual Knowledge Communities (VKC);

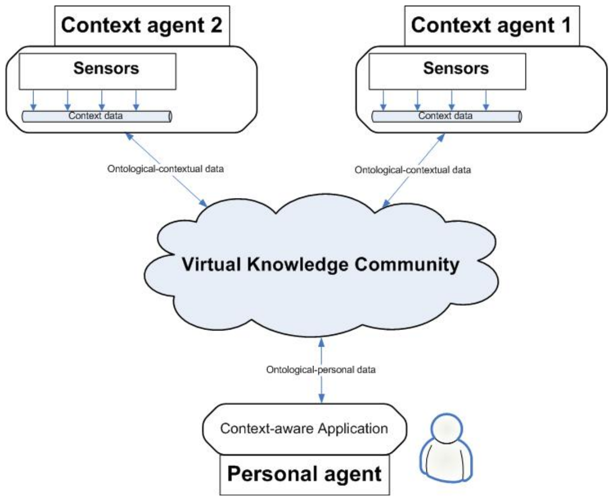

Figure 1: Description of the framework

2.1 Sensor-based Context Agent

We consider that a context-aware application is a complex decision maker. Decisions are based upon the knowledge captured from several sensors. We can note that there are different ways for integrating sensor data in the decision process. Three levels of fusion are commonly distinguished [12]:

1. Data fusion : Fusion of raw data before any semantic processing for the interpretation.
2. Feature fusion : Each signal is interpreted as a feature (ex: temperatures, activity level). Interpretations are then used as inputs of the decision process.
3. Decision fusion : Local decisions are made at each feature level and are inputs for final decision.

Due to the different nature of the information that sensors-based context agents can deliver, we consider that the fusion must be done at the feature level. This means that the context agent interprets the raw sensor data, to deliver a feature. For example, an electrical signal is processed by the context agent to deliver the temperature of a room. This specification is a consequence of the distribution of the sensors and context agents upon the architecture.

We distinguish four levels in the architecture of the sensor-based context agent:

1. **Sensors:** They capture environmental information and most of the time, deliver an electrical signal.
2. **Receiver:** Transmit the data from sensors to low level data.It acts as a first-level filter for contexts, before data is transferred to the next level.
3. **Interpreter:** This level plays a crucial role when raw data is specific for each kind of sensors. At this level the signal is translated into contextual data. For example, in the case of accelermoters, the data received from sensors is the electrical voltage, and the context data is: the person is standing, sitting, running In this case, the interpreter is a HBR (Human Behavioral Recognition) module. Another example is GPS sensor which delivers the location under the format latitude/longitude. Associated context data is for instance the name of the location (city, street, commercial center, station).
4. **Synthesizer:** The most important and innovating level in our approach, contextual data is here translated into knowledge and potentially accessible to other agents.

In a classic approach of sensor-based framework, the introduction of new sensors or new contextual information requires changes into applications relying potentially on these sensors. Contrarily, the Context Agent concept allows this information's integration to be done without any modification on other application's components. Changes regarding sensors-related information are introduced into the corresponding layers (receiver, interpreter and synthesizer). This is possible also since context agents are autonomous and because of the knowledge-based information exchange in-between components.

In addition, since a context agent communicates with other agents (also context agents), it can make use of gathered information to optimize its own interpretation of raw data sensed by its sensor. This re-enforces the autonomy of context agents and consecutively increases the re-usability of components.

2.2 Personal Agent

The personal agent is to be implemented on a wearable device (cell phone, PDA) and it plays the role of a personal assistant. The personal agent gathers context information delivered by context agents and provides a service to the end-user. The user's profile is also considered, consisting of user's preferences, historical data of use, personal information. The algorithm processed by the agent is a decision making process, based on gathers information and on the user's profile. This decision process is formalised in predicates, which can either be implemented at design time or be entered by the user.

2.3 Agent-based information exchanges

Systems composed of heterogeneous and distributed entities (sensors, PDA, software agents) requires efficient information exchange. An agent-oriented approach is convenient for such environment, but it does not solve alone the communication and knowledge exchanges issue.

The concepts of Virtual Communities (VC) [11] and Virtual Organisations [8] have been proposed as a layer for flexible exchange. Regarding knowledge exchanges, we have proposed the concept of Virtual Knowledge Communities [7]. The principles of VKCs are the following. Each agent has its own ontologies-based knowledge and its processes to interpret knowledge from other agents. Agents have a Belief-Desire-Intention structure. Intentions are expressed in terms of knowledge to be gathered or to be spread. When an agent intend to exchange knowledge, it founds a community based on knowledge topic. The community is declared in a yellow page system so that other agents can find it. Knowledge exchanges appears within the community until it is deleted. Security of the exchanges is guaranteed by community policies implemented by the agents. For instance, the policy sets conditions that agents have to fill in order to join the community. Policies can also take trust in consideration.

VKC have been implemented using JADE Platform [1] and JADE LEAP for mobile applications. It is fully FIPA compliant. Agent's knowledge is modeled with RDF ontologies, and message are coded using FIPA-ACL [3] standard. An agent that creates a community is the community leader. It specifies the knowledge topic of the community and the community policy. Exchanges are carried out using a blackboard system. Members of the community, depending on their rights within the community can write on and/or read the blackboard. Note that the interpretation of the knowledge gained through the communities is left to the capabilities of each agent.

3. THE 'WAKE ME UP' APPLICATION

We have implementated our framework in an application called 'Wake me up'. The scenario is the following. A metro passenger is sleeping during his/her going home. The application will help him/her to not miss the station he/she had to stop to. The solution we propose is to detect the user's activity level, and if necessary to make ring his/her cellphone when the right station is reached. This scenario requires some devices: transmitter, activity sensor and cell phone. These devices implement context agent or personal agent, supporting VKC functionalities.

3.1 Context agents implementation

Two context agent types are required for that application. The first one is linked to a transmitter and it delivers a station name. The second one is linked to a foot pressure sensor and delivers user's activity level.

3.1.1 Transmitter

The application requires that the metro company installs wireless transmitter in each station. The transmitter is associated to a contextual agent which is in charge of initiating a knowledge community to deliver information about the station. In our case, the delivered information is the name of the station. Agent's knowledge is represented using ontologies. We used Protégé [2] to create the associated ontologies. As the prototype rely on JADE, we used the RDF ontology format to store the ontologies. A simplified version of the transmitter's ontologies is given hereafter.

- Concept: Station name
- Station name : 'Ginza'

3.1.2 Sensors

We use a foot pressure sensor to measure the activity level of the traveler [13]. An accelerometer could also be used, supposing that the signal could be properly interpreted for activity level of a metro traveler. The sensed data is composed of voltage measures. It has no semantic meaning. Then, the foot pressure sensor is linked to a mobile device running JADE-LEAP and implementing a context agent which determines whether the traveler is asleep or not. An HBR (Human Behavioral Recognition) algorithm is used to classify the user's activity (sleeping, sitting, standing, walking, running). The agent related to the foot pressure sensor is also in charge of information exchanges in the frame VKC. It communicates using bluetooth. A simplified version of the sensor's ontologies is given hereafter.

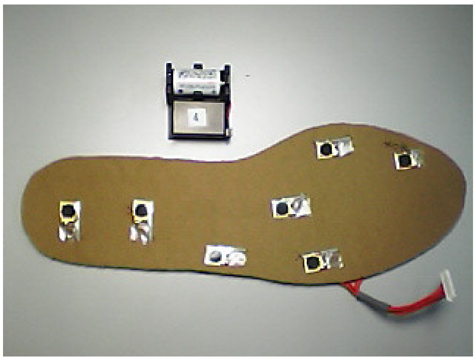

Figure 2: Foot pressure sensor

- Concept: Activity level
- Possible values : 'stands', 'sits', 'sleeps', ...
- Actual activity level : 'sleeps'

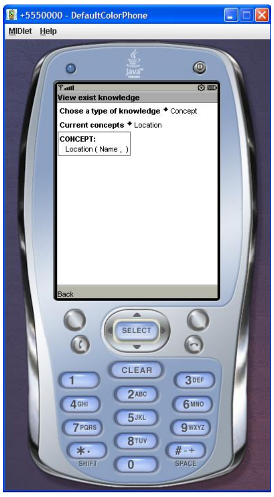

Figure 3: Personal Agent Interface

3.2 Personal agent's implementation

The personal agent is also implemented on JADE-LEAP but it runs on cell phones. The user's profile consists of the desired station. This information can be entered manually or it can be read in the user's agenda. The personal agent can initiate and join knowledge communities.

When the metro reaches a station, the personal agent detects the community initiated by the station's transmitter. It can join this community and then get the actual station's name. Then the personal agent checks whether the current station is corresponding to the desired one. In this case, it checks if the user is sleeping. So it initiates a community related to the activity level. The context agent related to the foot pressure sensor joins this community and delivers then the activity level value. The personal agent starts a decision process and it rings if the user is sleeping. The prototype uses a Nokia N91 phone with bluetooth communication.Figure 3 depicted the UI.

The personal agent owns the following ontology (simplified):

- Concept: Activity, Desired location, Actual location
- Actual activity: No value
- Desired location: 'Ginza'
- Actual location: 'Shibuya'
- Predicate: IsSameLocation(Location1, Location2); IntentionDescription(Desired location, Actual location)

4. CONCLUSION

The use of personal sensors for mobile context-aware applications constitutes a promising field of research and development. Actual applications in this field requires a redesign phase in order to introduce new sensors or new contextual information. We proposed in this paper a framework based on knowledge exchanges in-between application components. These exchanges appear in the frame of so-called Virtual Knowledge Communities. VKC allows information's integration without any modification on other application's components. We defined two types of components into such applications : Context Agent, related to sensors, and Personal Agents related to users. Our approach is fully compliant with pervasive systems and ambient intelligence, because it allows exchanges in-between components which are autonomous and decentralized.

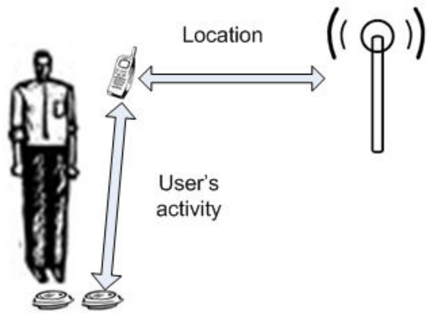

Figure 4: Communication between the agents

Future works will focus on the improvement of this framework for the development of greater applications. Research topics are numerous: knowledge representation, knowledge interoperability models, decision processes, agent's policy and trust, etc. It is also a real challenge to identify new highly relevant application fields and scenarios for context-aware personal assistance in mobility.

5. REFERENCES

[1] Jade - java agent development framework. `http://jade.tilab.com`.

[2] Protégé. `http://protege.stanford.edu/`.

[3] ACL-FIPA. *Fipa Personal Assistant Specification (2000)*. FIPA, http://www.fipa.org/specs/fipa00083/XC00083B.html.

[4] G. Biegel and V. Cahill. A framework for developing mobile, context-aware applications. *percom*, 00:361, 2004.

[5] H. W. Gellersen, A. Schmidt, and M. Beigl. Multi-sensor context-awareness in mobile devices and smart artifacts. *Mob. Netw. Appl.*, 7(5):341–351, 2002.

[6] K. Henricksen and J. Indulska. A software engineering framework for context-aware pervasive computing. *percom*, 00:77, 2004.

[7] P. Maret, M. Hammond, and J. Calmet. Virtual knowledge communities for corporate knowledge issues. In *ESAW*, pages 33–44, 2004.

[8] J. Patel, W. T. L. Teacy, N. R. Jennings, M. Luck, S. Chalmers, N. Oren, T. J. Norman, A. D. Preece, P. M. D. Gray, G. Shercliff, P. J. Stockreisser, J. Shao, W. A. Gray, N. J. Fiddian, and S. Thompson. Conoise-g: agent-based virtual organisations. In *AAMAS*, pages 1459–1460, 2006.

[9] S. Pinyapong and T. Kato. Query processing algorithms for time, place, purpose and personal profile sensitive mobile recommendation. In *CW '04: Proceedings of the 2004 International Conference on*

Cyberworlds (CW'04), pages 423–430, Washington, DC, USA, 2004. IEEE Computer Society.

[10] D. Salber and G. Abowd. The design and use of a generic context server. Technical report, 1998. Also available as `http://www-static.cc.gatech.edu/fce/contexttoolkit/pubs/pui98.pdf`.

[11] T. Schoberth and G. Schrott. Virtual communities. *Wirtschaftsinformatik*, 43(5):517–519, 2001.

[12] R. Sharma, V. Pavlovic, and T. Huang. Toward multimodal human-computer interface. *Proceedings of the IEEE*, 86(5):853–869, May 1998.

[13] C. Sugimoto, M. Tsuji, G. Lopez, H. Hosaka, K. Sasaki, T. Hirota, and S. Tatsuta. Development of a behavior recognition system using wireless wearable information devices. pages 1–5, 2006.