Implicit Environment-based Coordination in Pervasive Computing

Amine Tafat
PAI Group
Informatics Department
Fribourg University,
Switzerland
Amine.Tafat-Bouzid@unifr.ch

Michele Courant
PAI Group
Informatics Department
Fribourg University,
Switzerland
Michele.Courant@unifr.ch

Béat Hirsbrunner
PAI Group
Informatics Department
Fribourg University,
Switzerland
Beat.Hirsbrunner@unifr.ch

ABSTRACT

By immersing computational systems into the physical world, pervasive computing brings us from traditional desktop computing interactions into a form closer to human interactions, which are characterized by their dependance on the environment. Furthermore, acting within the environment can contribute in many cases to a form of implicit interaction in which no explicit communication message is exchanged. However, this way of interacting cannot be achieved without a shared common knowledge with a well defined semantic. This appeals to explore a new approach of interaction mediated by the environment, closer to human interactions, and therefore, more suitable to achieve the intended silent computation of pervasive computing.

To address this issue, we present in this paper XCM; a generic coordination model for pervasive computing. XCM is organized around few abstract concepts (entity, environment, social law and port) and is expressed as an ontology. While the abstract concepts of XCM deal with the environmental representation and context-dependency, the ontological representation allows to achieve knowledge sharing and context reasoning.

Keywords

Pervasive Computing, Coordination Model, Environment Awareness, Implicit Interaction, Ontology.

1. INTRODUCTION

Pervasive environments can be considered as physical environments saturated with computing and communication, yet gracefully integrated with human users. In the future our daily environment will contain a network of more or less specialized computational devices that will interact with us and among themselves [3].

SAC '05, March 13-17, 2005, Santa Fe, New Mexico, USA

However, the tendency of merging physical world and computational systems, requires appropriate software infrastructure and development tools. In particular, from the coordination and integration perspective, the use of computers in non-desktop environments goes beyond traditional desktop interaction paradigm.

When observing interactions between humans it appears that communication is influenced by context and environment. The situation in which the communication takes place provides a common ground. This common ground generates implicit conventions, which influences and to some extent sets the rules for interaction and also provides a key to understand and decode the meaning of words and gestures [2]. As suggested in [10], there is a need to systematize environment-based coordination approaches.

Moreover, interaction in our daily environment is not always explicit. In many daily life situations usual actions are contextually used as messages for communicating without any direct and explicit communication. It seems that silent communication [9] is the most natural and more able way to achieve the intended vision of invisible and calm pervasive computing as formulated originally by Weiser [1] than the explicit interaction. On the other hand, we cannot have environment-based and implicit coordination without having a common base knowledge, with a well defined semantic, shared between coordinated partners (or coordinated entities), making sure that the action performed on one side is captured and correctly interpreted by the other side.

The presented work is therefore a coordination model dedicated to pervasive computing called XCM and built around few abstract concepts allowing to achieve environment-based and implicit interactions within a pervasive computing approach. Its contribution consists mainly of an explicit representation and management of the environment, and a very flexible approach of interaction. XCM is expressed as an ontology, using OWL Semantic Web Language [8]. The ontological representation of the model allows achievement of environmental and contextual knowledge sharing, semantic agreement, and context reasoning between interactive partners.

The present paper is organized as follows: the second section describes the concepts of the abstract model XCM; the third section describes the representation and implementation of XCM model as an ontology using Ontology Web Language (OWL); in Section 4, a conference scenario, as example illustrating XCM model, is discussed; and finally a section on

related work and conclusions.

2. X COORDINATION MODEL (XCM)

XCM is a coordination model intended to support the specificity of a pervasive application. Its essential characteristics are:

- Genericity, which is obtained through a high level of abstraction based on the notion of entity.
- A capacity to handle the dynamics of pervasive execution environments -either physical or virtual-, and the context-sensitivity of applications, thanks to the explicit notion of environment.
- A homogeneous management of the contextual dynamics of components by the unique formalism of social law attached to the notion of environment, and a mechanism of port allowing entities to interact in a very flexibly and powerful manner.

As a coordination model, XCM uses Ciancarini's approach [5], and the vision of coordination proposed by Malone [4]. In addition, our approach adds on a theoretical component inspired by " autopoiesis " i.e. the modelling of living systems elaborated by Varela and Maturana [7]. The interest of this heritage is double. First, it allows profiting from the specificity of the physical space for modelling mechanisms like the construction and the maintenance of organism frontiers. Second, it introduces a fundamental distinction between organisation (domain of control expression) and structure (domain of entity existence).

2.1 Entity and agent

In XCM,everything is an entity. An entity e_i is defined by its structure, which is expressed as a recursive composition of entities $e_{i1}...e_{in}$ -called components of e_i- and by its organisation. An entity, whose structure cannot be decomposed, is called atomic; it denotes a pre-constructed element of the system. On the other hand, the highest-level entity recursively containing all the other entities of the system, is called the universe of the system. The organisation of an entity e_i specifies the rules which are governing the assembling of components in the structure and their dynamics. It then characterises the domain of the interactions, which are applying to e_i. It is expressed as a set of rules called by extension the social laws of e_i.

2.2 Environment and social laws

At a given time of the system's existence, every entity e_i -except the universe- exists as a component of another entity e which is called its environment.
Thanks to its social laws, the environment e prescribes the structure and the dynamics of e_i. These ones determine in particular the interactions between e_i and e, as well as between e_i and the e_j -i.e. they rule out the assembling and disassembling of e_i with the other components of e-. These laws also govern the input of e_i into e, and the output of e_i from e. Considering the case of an antenna, for example, where its environment is its coverage area, and the entering/leaving of mobile devices into/from it is controlled by its social laws.
When its social laws confer to an entity the capacity to initiate operations modifying its own structure (internal autonomy) or its relations with its environment (external autonomy), this entity is commonly called an agent . An entity can be related to several environments, however, it can be active at the most in one environment. Apart from this environment, it can be at the most "virtually" or "sensorially" present in the other environments.
The notion of environment then encompasses within a single concept all the semantic diversity of the pervasive application components: a social semantic, inherited from coordination in general, and a physical semantic of entities, which becomes essential as soon as the entities are evolving onto devices subjected to the laws of the physical space.
By social semantics, we mean the capacity of an entity to belong to a social structure such as a group of entities it is interacting with. The model supports a multiple organisational linkage of an entity, for example, a player is linked to his football team and the company he is working for. By physical semantics, we namely mean that it is impossible for an agent to act in two environments at the same time, or to be "teleported" from one environment to another .
An entity can however remain " aware of " another environment than the one in which it is active by opening some specific communication channels in this environment, thus implementing a remote perception mechanism. Finally, the same way the "autopoietic dynamics" rules in Varela & Maturana include meta-control and self-control, social laws may also govern the structure and the organisation visibility along the entangled entity and environment hierarchies.

2.3 Ports

A port is a special type of entity dedicated to communication between entities. A port p has the specificity to be generally active while being coupled to an agent a_i, which is the port's master. The coupling between a_i and p is obtained through a special type of composition called interface, which is therefore specified by social laws (of p and a, and of their common environment). These ones define how the port is assembled to its master, for example they may define the modalities of using the port (in terms of communication protocol, of bandwidth, etc). For answering pervasive computing needs, we also distinguish removable and irremovable ports. For example, for a human agent, a mobile phone is a removable port, whereas an audio-prosthesis is irremovable. An agent may be coupled to several ports. It can acquire ports, and dissociate itself from ports dynamically. The agent-port assembling and disassembling procedures are triggered either explicitly, by an initiative of a, or implicitly by the entrance of a into a new environment, or by the environment dynamics, which may for example welcome new ports, which are automatically coupled to a.
The notion of port is then a fundamental mechanism, which confers to XCM the ability to coordinate context-sensitive entities. This context-awareness is the central characteristics of application components in pervasive computing.

3. XCM ONTOLOGY

Ontology is a term borrowed from philosophy that refers to the science of describing the kinds of entities in the world and how they are related. An ontology is a formal explicit description of concepts in a domain of discourse (or classes), properties of each class describing various features and attributes of the class, and restrictions on properties [8].
The Web Ontology Language OWL is a Semantic Web lan-

guage for defining and instantiating ontologies, designed as a standard of W3C standardisation organisation. It's main advantages are openness, extensibility and the support of ontology reuse. An OWL ontology may include descriptions of classes, properties, instances of classes, and relationships between these instances.

We introduce here a partial representation of XCM model necessary for demonstrating our conference scenario example in section4.
`Entity` class models the XCM entity concept. This abstract class defines a set of properties that are common to all entities, which consists of `HasStructure` and `HasOrganisation`. `Entity` classes have associated containment relationships. The relationships are defined by two related object properties called `HasComponents` and `IsComponentOf`.
`HasComponent` property describes that the first individual entity from class domain is composed by the second individual entity from range class, and inversely with `IsComponentOf` that describes the fact that entity from domain class is a component of the entity from range class. These two properties are defined as an inverse property of each other, as shown in the partial XCM ontology code presented in Figure1.
As atomic entity does not contain any other entity, we introduce an abstract class called `AtomicEntity` which inherits all properties from its superclass `Entity` while adding restrictions on the range of `HasComponents` property. In `AtomicEntity` class, the cardinality of `HasComponents` property is 0 indicating all instances of this class do not contain any other entity. On the other hand, `CompoundEntity` is introduced to represent a set of entities that contains at least one `Entity` class member. `CompoundEntity` inherits all properties from `Entity` class with restrictions on minimal cardinality of `HasComponents` property (Figure1).

If two instances of `Entity` class, e_i and e are related by `IsComponentOf` property instance, then e is an *environment* of e_i. Therefore, we define a new class called `Environment` as equivalent to `CompoundEntity`, since every compound entity represent an environment for its components entities (Figure 2). *Universe* is a special entity which is not contained by any other entity, or ,otherwise described, is contained by itself.

Regarding Social Laws, we define `HasLaw` property, expressing the fact that environment e has social law L attached to its organisation; `LawOf` property, defined as inversed `HasLaw` property, expressing that law L is related to environment e, which means that in order to interact together, the components of the environment e must act on conformity with the law L, when interacting within e environment (Figure 2).

As port is a particular entity always related to a specific entity, and for which it acts as a communication interface, two properties `HasOwner` and `HasPort` are introduced, with the second property as an inversal property of the first.
A port is defined as a restricted entity which has at least one entity owner, that means that at least one instance of `HasOwner` property relating the port to its entity owner exists. As an entity related to a port is called Agent, an Agent class is defined as subclass of *Entity* , with additional restrictions on the range of `HasPort` property. In `Agent` class,

```
<owl:Class rdf:ID="Entity">
 <owl:UnionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#AtomicEntity/>
    <owl:Class rdf:about="#CompoundEntity/>
 </owl:UnionOf>
...  </owl:Class>

<owl:ObjectProperty rdf:ID="HasComponents">
  <rdfs:domain rdf:resource="#Entity"/>
  <rdfs:range  rdf:resourcre="#Entity"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="IsComponentOf">
  <owl:inverseOf rdf:resource="#HasComponents"
  <rdfs:domain rdf:resource="#Entity"/>
  <rdfs:range  rdf:resourcre="#Entity"/>
</owl:ObjectProperty>

<owl:Class rdf:ID="AtomicEntity">
  <rdfs:subClassOf rdf:resource="#Entity"/>
   <rdfs:subClassOf> <owl:Restriction>
    <owl:OnProperty rdf:resource="#HasComponents"/>
    <owl:maxCardinality rdf:datatype=
         "&xsd;nonNegativeInteger">0
    </owl:maxCardinality>
   </owl:Restriction> </rdfs:subClassof>
 </owl:Class>
<owl:Class rdf:ID="CompoundEntity">
  <rdfs:subClassOf rdf:resource="#Entity"/>
  <rdfs:subClassOf> <owl:Restriction>
   <owl:OnProperty rdf:resource="#HasComponents"/>
   <owl:minCardinality rdf:datatype=
        "&xsd;nonNegativeInteger">1
   </owl:minCardinality>
  </owl:Restriction> </rdfs:subClassof>
</owl:Class>
```

Figure 1: Partial implementation of XCM model(1)

the cardinality of the property `HasPort` must be at least equal to 1 indicating all instances of this class have the ability to communicate with the external world with at least one port (Figure 3).

4. CASE STUDY: CONFERENCE SCENARIO

To illustrate the XCM model described above, a conference scenario is presented based on the original conference scenario described by Anind D. Key [14].
Three colleagues from the same research group have registered for a conference and are trying to attend different presentations. The conference has multiple sessions (four sessions for example) held in alternative rooms, with each session holding a set of presentations.
Each individual obtains a handheld device(PDA) when he gives his personal details as well a s his research interests at the registration desk. The PDA automatically displays a copy of the conference schedule highlighting the paper tracks which are of interest to each person. In our case, as the three colleagues belong to the same research group, they would have the same highlighted presentation sessions, which they would like to attend. To get a maximum outcome from the conference, each of the three individuals should attend a different presentation.
When each individual holding a PDA walks in a conference

```
<owl:Class rdf:ID="Environment">
 <owl:equivalentClass rdf:resource=
  "#CompoundEntity"/> </owl:class>
<Entity rdf:ID="Universe" />
<owl:Class rdf:ID="Law">
  <rdfs:subClassOf rdf:resource="#Entity"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="HasLaw">
  <rdfs:domain rdf:resource="#Entity">
  <rdfs:range rdf:resource="#Law">
</owl:objectProperty>
<owl:ObjectProperty rdf:ID="LawOf">
  <rdfs:domain rdf:resource="#Law">
  <rdfs:range rdf:resource="#Entity">
</owl:objectProperty>
```

Figure 2: Partial implementation of XCM model(2)

room, he is automatically picked up by the network and his name is added straight away to the attendents list.

The whole conference space including the four conference rooms represents the physical space, which is modelled within XCM as environmental compound entities ($ConferenceSpace$,$ConfRoom1$,...,$ConfRoom4$), each including its components.
The research group, to which belong the three colleagues, is represented as virtual environmental entity $ResearchGroup$. The colleagues are represented as three Agents: $RGMember1$, $RGMember2$, $RGMember3$. Each agent is represented within two environments: a physical environment which is the conference room in which the person is attending a presentation, and a virtual environment which is the research group to which the three colleagues belong (Figure 4).

The social law governing the virtual environment (the research team) is the fact of attending different presentations in order to maximise the outcome of the research group from the conference, which represents the interaction rule between individuals within this virtual environment. If an individual (eg.$RGMember1$ chooses to attend a presentation taking place in $ConfRoom1$ for example, the other individuals will have to choose another highlighted presentation related to their research interest held in another conference room. This interaction rule is formalised in XCM as shown in Figure 5.
The `ExclusivPresenceInConf` law restricts the `IsPresentIn` property (which is a subproperty of `HasComponent` property) with the `InverseFunctionalProperty` property. This means that two members of $ResearchGroup$ cannot be located in the same conference room. Thus, when $RGMember1$ joins $ConfRoom1$ in order to attend a presentation session, the above fact is generated and is added to the contextual information about environment:

```
<owl:Thing rdf:about="#RGMember1">
  <IsPresentIn rdf:resource="#ConfRoom1" />
</owl:Thing>
```

and this session is no more highligted in $RGMember2$'s and $RGMember3$'s PDAs, therefore, they can only choose from the remainning highlighted presentations held in $ConfRoom2$ or $ConfRoom3$ according to the `ExclusivPresenceInConf`

```
<owl:Class rdf:ID="Port">
 <rdfs:subClassOf rdf:resource="#Entity"/>
 <rdfs:subClassOf> <owl:restriction>
    <owl:OnProperty rdf:resource="#HasOwner">
    <owl:minCardinality rdf:datatype=
           "&xsd;nonNegativeInteger">1
    </owl:minCardinality>
 </owl:restriction> </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Agent">
 <rdfs:subClassOf rdf:resource="#Entity"/>
 <rdfs:subClassOf> <owl:Restriction>
   <owl:OnProperty rdf:resource="#HasPort"/>
   <owl:minCardinality rdf:datatype=
            "&xsd;nonNegativeInteger">1
   </owl:minCardinality>
 </owl:Restriction> </rdfs:subClassof>
</owl:Class>
<owl:ObjectProperty rdf:ID="HasPort">
 <rdfs:domain rdf:resource="#Entity"/>
 <rdfs:range  rdf:resourcre="#Port"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="HasOwner">
 <rdfs:domain rdf:resource="#Port"/>
 <rdfs:range rdf:resource="#Entity/>
</owl:ObjectProperty>
```

Figure 3: Partial owl implementation of XCM model(3)

law related to $ResearchGroup$ environment. Note that such interaction is an implicit interaction where the agents coordinate themselves by observing the behaviour of the others to rule out the possibility of being together in the same place.

This example shows how XCM integrates the implicit interaction within pervasive computing. The agents interact without exchanging an explicit message dedicated to the interaction. Their interaction occurs in an implicit way based on both the social law expressed within their environment and the information gathered from it.
The definition of the environment within XCM model as a basic concept has allowed the information collected from the environment to be integrated in the application and to be exploited in the coordination process. While traditional approaches model this form of interaction by exchanging communication messages either directly between agents or through a shared dataspace, we argue that this way of modelling the interaction does not corrolate with the needs of pervasive computing.

5. RELATED WORK AND CONCLUSION

Many interesting frameworks investigate pervasive computing research, such as ContextToolkit [14], Cooltown [13], Intelligent Room [12], OneWorld [11]. However, such systems failed to uncouple the computation concern from the coordination concern, which is the corner stone of coordination models, and which is essentially needed in pervasive computing where integration is the main concern.
EventHeap[6] is the first tuplespace coordination model intended to pervasive computing rooms; limited, however, to a special purpose of interactive workspaces. Defining solutions to specific coordination problems prevent to have necessary

```
<XCM:Environment rdf:ID="ConferenceSpace" >
   <XCM:HasComponent rdf:ID="ConfRoom1"/>
   ...
   <XCM:HasComponent rdf:ID="ConfRoom4"/>
</XCM:Environment>

<XCM:Environment rdf:ID="ConfRoom1" >
   <XCM:IsComponentOf rdf:ID="ConferenceSpace">
   <XCM:HasPort rdf:ID="SpotPointCR1">
</XCM:Environment>
...
<XCM:Environment rdf:ID="ConfRoom4" >
   <XCM:IsComponentOf rdf:ID="ConferenceSpace">
   <XCM:HasPort rdf:ID="SpotPointCR4">
</XCM:Environment>

<XCM:Environment rdf:ID="ResearchGroup" >
   <XCM:HasComponent rdf:ID="RGMember1"/>
   ...
   <XCM:HasComponent rdf:ID="RGMember3"/>
</XCM:Environment>

<XCM:Agent rdf:ID="RGMember1">
<XCM:HasPort rdf:ID="PDA1" />
<XCM:IsComponentOf rdf:ID="ResearchGroup"/>
</XCM:Agent>
...
<xcm:Agent rdf:ID="RGMember3">
<XCM:HasPort rdf:ID="PDA3" />
<XCM:IsComponentOf rdf:ID="ResearchGroup"/>
</XCM:Agent>
```

Figure 4: Partial description of conference scenario with XCM

abstraction to generalise the model to other environments. What is needed though, is to systematically take the environment into account, by including it to the model as a first order concept. The main advantage of this approach is that it permits the interaction mediated by the environment [9] which is more suitable to pervasive computing.

In this paper we have proposed a general coordination model for pervasive computing, expressed as an ontology based on few abstract concepts: entity, environment, social laws and port. Representing the environment as a first class concept in the model allows to go beyond traditional desktop computing interaction towards implicit environment-based interaction. This new approach of interaction, supported by XCM, is closer to human interactions and therefore more suitable to achieve silent integration of computation capabilities in our daily life, as expressed in pervasive computing.

6. REFERENCES

[1] M.Weiser The computer for the 21st century. Scientific American, 265(30):94,104, 1991.

[2] Albercht Schmidt. Ubiquitous Computing- Computing in Context, PHD thesis, Computing Department, Lancaster University, UK, 2002.

[3] Lyytinen, Kalle, and Yoo Youngjin. Issues and Challenges in Ubiquitous Computing. Communications of the ACM 45(12): 62-65, 2002.

[4] T.W. Malone and K. Crowston. The Interdisciplinary Study of Coordination. ACM Computing Surveys, 26(1):87-119, March 1994.


```
<XCM:Law rdf:ID="ExclusivPresenceInConf">
<XCM:LawOf resource:about="#ResearchGroup">
<owl:restriction>
 <owl:OnProperty rdf:resource="#IsPresentIn">
 <rdfs:domaine rdf:resource="#ResearchGroup"/>
 <rdfs:range rdf:resource="#ConferenceSpace"/>
 <rdf:type rdf:resource=
        "&owl;InverseFunctionalProperty"/>
 </owl:OnProperty>
</owl:restriction>
  ...
</XCM:Law>
<XCM:Environment rdf:about="#ResearchGroup">
  <XCM:HasLaw rdf:about="#ExclusivePresenceInConf">
</XCM:Environment>
```

Figure 5: Social Law descripition

[5] P.Ciancarini, F. Arbab and C. Hankin: Coordination languages for parallel programming. Parallel Computing, 24 (7):989-1004, 1998.

[6] Bradley Earl Johanson. Application Coordination Infrastructure for Ubiquitous Computing Rooms. PHD thesis, Stanford University 2003.

[7] F. Varela and H. Maturana. Autopoiesis and Cognition: The realization of the Living. Boston Studies in the Philosophy of Science in Cohen, Robert S., and Marx W. Wartofsky (eds.) , Vol. 42, Dordecht (Holland): D. Reidel Publishing Co., 1980.

[8] Sean Bechhofer, et al. OWL Web Ontology Language Reference, w3c recommendation, 10 February 2004. http://www.w3.org/TR/owl-features/.

[9] C. Castelfranchi. Silent agents: From observation to tacit communication. Workshop Agent Tracking: Modelling Other Agents from Observations, in AAMAS 2004, July 2004, New York, USA.

[10] A. Omicini, A. Ricci, M. Viroli, C. Castelfranchi, L. Tummolini, Coordination Artifacts: Environment-based Coordination for Intelligent Agents. in AAMAS 2004, July 2004, New York, USA.

[11] Grimm, R., et al. A System Architecture for Pervasive Computing. in 9th ACM SIGOPS European Workshop. 2000. Kolding, Denmark.

[12] Coen, M.H. A prototype intelligent environment. in First International Workshop CoBuild'98 Proceedings. Springer-Verlag, 1998.

[13] Kindberg T., et al. People, places, things: Web presence for the real world. in Third IEEE Workshop on Mobile Computing Systems and Applications. 2000. Los Alamitos CA USA.

[14] K.Dey, et al. The Conference Assistant: Combining Context Awareness with Wearable Computing. ISWC 1999: 21-28.