Ubiquitous Intelligent Object: Modeling and Applications

Miaomiao Wang[1,2], Jiannong Cao[2], Joanna Izabela Siebert[2],Vaskar Raychoudhury[2], Jing Li[1]
[1] *Department of Computer Science,*
The University of Science and Technology of China, Hefei, Anhui, China
[2] *Internets and Mobile Computing Lab, Department of Computing,*
The Hong Kong Polytechnic University, Kowloon, Hong Kong
{csmmwang, csjcao, csjsiebert, csvray}@comp.polyu.edu.hk

Abstract

Recently, research on ubiquitous computing has attracted a lot of research. Many existing works focus on the design and development of Ubiquitous Intelligent Environments (UIEs). Essentially, UIE is composed of collections of Ubiquitous Intelligent Objects (UIOs) and their interactions. Conceptualize and abstract the UIOs as well as exploring more UIO-oriented applications are essential to the future development of UIEs. However, so far, there has been no work on the formal description and modeling of UIOs. Addressing this deficiency, in this paper, we study the modeling of UIOs. We identify the model design principles and develop a model for representing the abstract data types and operations, as well as the relationships between the UIOs. We also discuss the challenging issues in implementing the model and present a ubiquitous searching application as an example to show how the proposed modeling techniques can be used.

Keywords: Ubiquitous Computing, Object Model, Knowledge sharing

1. Introduction

The widespread deployment of integrated sensing, computing, and communication systems is transforming the physical world into a ubiquitous computing platform. Sensing tags, memory, computing and communication capabilities are immersed into our living environments, appearing on motion detectors, door locks, light bulbs, alarms, cellular phones, vehicles, and possibly in person's wallets or even key rings [1]. So, recently research on the *Ubiquitous Intelligent environments* (UIEs), such as the *Smart Room* and *Intelligent Market Hall*, has attracted a lot of attention. Essentially, UIE is composed of collections of interactive Ubiquitous Intelligent Objects (UIOs). Modeling the UIOs is an important and fundamental problem for large scale Ubiquitous Intelligent sharing [2] in UIE. But to the best of our knowledge, so far there has been no research on the formal description and abstraction of UIOs.

Conceptualize and abstract the UIOs are essential to originate new design approaches and develop programming tools, as well as to explore more exciting applications, in the future development of UIE. However, UIOs modeling and implementation are not a trivial task, because the characteristics of objects in both the physical world and cyberspace should be taken into consideration. In this paper, we explore the UIOs at both the conceptual level and the implementation level. We first identify the model design principles and then develop UIO models for representing the abstract data types and operations of UIOs with the consideration of privacy issues as well as the complex relationship amongst them. We also discuss the inadequateness of existing implementation strategies and the challenges of improving them. We study the ubiquitous searching application [3] as an example to show how the proposed modeling techniques can be used. The example shows how UIOs can trigger and help the development of exciting new applications.

The rest of the paper is organized as follows: Section 2 briefly reviews the related work. Section 3 proposes the concept of UIO, discusses the model design principles, and describes the UIO model. It also discusses the model implementation issues. In Section 4, we describe the ubiquitous searching application. Finally, in Section 5, we conclude this paper by discussing the future works.

2. Related Work

To the best of our knowledge, this paper initiates the study of modeling UIOs. In this section, we briefly review the existing works with similar motivations or contributing to the modeling or implementation methods used in this paper.

A class of related works that have been studied for many years is about modeling the context information of ubiquitous computing environments [4, 5]. They focus on abstracting the information from the high level applications' points of view. In contrast, our work

0-7695-3007-9/07 $25.00 © 2007 IEEE
DOI 10.1109/SKG.2007.153

in this paper addresses the issues in modeling the intelligent objects that compose the UIE.

The second kind of work is about integration of ubiquitous devices and data. An example is SensorWeb [6, 7, 8], whose main objective is resource sharing. SensorWeb intends to make various types of devices, including web-resident sensors, instruments, image devices, and repositories of senor data, discoverable, accessible, and controllable via the WWW. In doing this, SensorWeb provides a model for the sensor information. Compared with SensorWeb, our work distinguishes the concepts of "intelligent devices" and "intelligent objects", separating them into the concept level and the implantation level, so as to make the model more generic. In addition, we focus more on the automatic information sharing between the UIOs.

Another class of related work is the traditional object-oriented modeling and agent modeling approaches [9, 10]. However, we investigate more issues unique to the UIE such as the temporal and spatial relationships in the physical world.

3. Ubiquitous Intelligent Object

3.1 Concept of UIO and Modeling Principles

We define ubiquitous intelligent objects (UIO) as the physical objects in the real world that are equipped with the ubiquitous computing devices and have *intelligent abilities*.

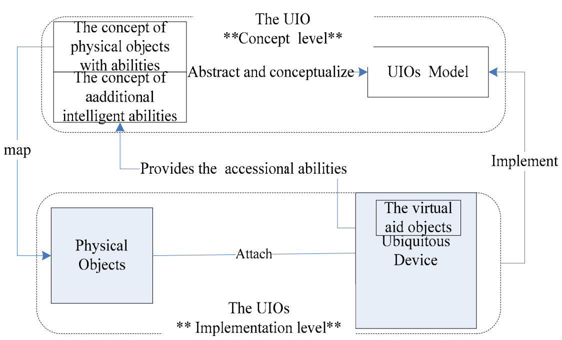

Figure 1 Relationship between the basic concepts

Figure 1 describes the conceptualization of UIO abstracted from both the concept level and the implementation level. At the conceptual level, a UIO model is defined to represent the essential features of UIO, which are extracted from the physical object itself with its inherent abilities and additional intelligent abilities provided by the associated ubiquitous devices. At the implementation level, *physical objects* are the objects in the real world, such as plants, animals, and people. Not all the physical objects can be abstracted as UIOs, unless it has some specified *intelligent abilities.* The *intelligent abilities* can come from the physical objects themselves, and can also be provided by the *ubiquitous devices* with *the virtual aid objects.* We call the *intelligent abilities* provided by the *Ubiquitous devices* as *additional intelligent abilities. The Ubiquitous devices* that can be used to implement the *UIOs model* include electronic labels/tags, RFIDs, MEMS devices, tiny sensors, etc. The *virtual aid objects* can be a snippet of program or data that help represent the physical object and provide the additional intelligent abilities.

To explain the concept, let us look at an example. Jack's dog is a physical object with its inherent abilities such as move and bark. When it is attached with some ubiquitous devices, such as a RFID tag and a temperature senor, it is provided with additional intelligent abilities, such as being identified and reporting surrounding temperature, and it becomes an UIO, called a UIO-Dog. The program and information stored on the devices are called the virtual aid objects.

Before presenting our UIO model, we first identify the model design principles by discussing what features and information of UIO should be modeled. First, we can look at the "U" part of the UIO concept from two aspects: the comprehensive features and the diversiform features. The UIOs model should be generic enough to abstract the objects in UIE, and at the same time flexible enough to embody their special features. Second, for the "I" part of the UIO concept, the intelligent abilities of UIO are the combination of the abilities of the physical objects and the abilities provided by the associated devices with the virtual aid objects. The UIOs model should model the necessary abilities at different levels. Third, the "O" part of the UIO concept suggests that the UIO model should share same basic requirements as traditional object modeling, such as the life cycle and the static or dynamic relationships. Of course as special kinds of objects, UIOs should extend the corresponding object concept.

To better understand the above principles, let us see an example. Suppose O-Dog is a virtual object in traditional object-oriented computing that represent Jack's dog by abstracting its characteristics and behavior. Comparing the UIO-Dog with the O-Dog, we can find out the following differences. First, the life cycle of O-Dog begins when it is created, and ends by some explicitly or implicitly invoked mechanisms (e.g. garbage collecting mechanism), while the UIO-Dog' life cycle is affected by more factors. It begins when the dog is attached with the devices so as to possess specified intelligent abilities. If the physical dog disappears (e.g. dies) or the necessary intelligent abilities are removed (e.g. caused by the fault of the intelligent device or the virtual aid objects) the UIO-Dogs' life cycle ends. Second, the relationships between O-Dog and other virtual objects are predefined while the relationships between of

UIO-Dog and other UIOs reflect the social characteristics of the UIO-dog and are dynamic. It is because the UIO-dog may move within and among different UIEs, and change its relationship with other UIOs. Third, the O-dog gets the information from only other virtual objects but the UIO-dog gets the information from not only other UIOs, but also the physical environments and the virtual objects in the traditional computing environments. Finally, if the O-Dog also has a field that contains the information about the dog or the surrounding environment, e.g. the temperature, the corresponding value needs be updated manually, while for UIO-Dog, the corresponding value is automatically obtained from the sensing devices directly. So the information about UIO-Dog can be made real-time and free from being outdated as it might be in the case of manual input. In summary, UIOs is a special kind of objects, and their particular characteristics should be carefully considered during the modeling process.

The concept of UIO has far-reaching impact. It even extends the concept of the Internet. From the networking point of view, UIOs can form a network by themselves, and can be always or intermittently connected. They can also connect to the existing networks. Based on this inter-connection infrastructure we can define the UIO intelligence sharing framework, referred to as *Internet of Things.* As shown in Figure 2, in the framework, *internet of cyberspace* refers to the internetworking for sharing the intelligence embodied as virtual objects, e.g. music, documents and programs providing the computing services as in today's Internet. On the other hand, UIOs can share their intelligence in the *internet of physical space* directly where their information can be processed and consumed on-spot, in a real-time and distributed manner without being bothered to transfer information into and from the internet of cyberspace. Furthermore, UIOs can also share the intelligence with the virtual objects, based on the interaction between the virtual aid objects residing in it and in the virtual objects.

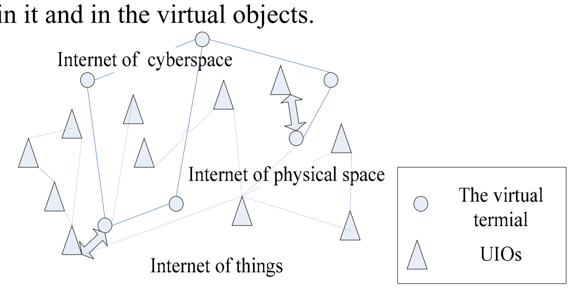

Figure 2 The internet of things

3.2 UIOs Model

Based on the above discussion, now we can propose the UIO modeling. We will use UML for the description of the models. Figure 3 describes the high level UIO model. Internet of things is composed by the internet of cyberspace, UIOs, and the UIOs relationship. UIOs and UIOs relationship can be atomic or composite. (Note that we model the relationships between the virtual objects in the cyberspace and UIOs under the UIOs relationship).

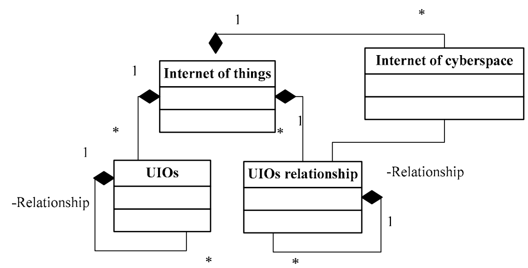

Figure 3 The high level UIOs model

Figure 4 describes the basic UIO model, consisting of the abstract data type and the abstract operation of UIOs. Both components are constrained by the UIO abilities, which are defined, as shown in Figure 5, at different levels.

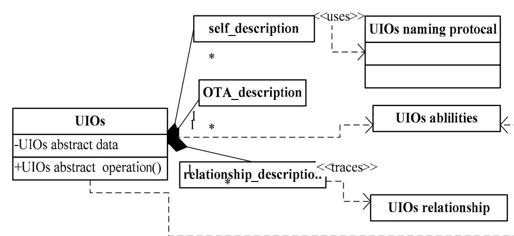

Figure 4 The basic UIOs model

The only necessary ability of a UIO is "being identified", meaning the UIO can be distinguished from other UIOs. As mentioned before, the *UIO abilities* is an integrated concept. For example, the sensing-ability of a UIO representing a person as the physical object can come both from that person and the sensing devices associated with the person.

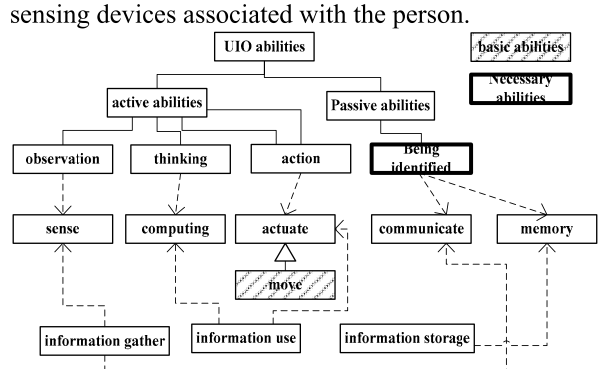

Figure 5 The UIOs abilities model

Now let us go back to Figure 4 and define the abstract data of UIOs. The abstract data are the description of the UIOs and contain three parts: *Self_description, OTA_description,* and *Relationship_description*.

The Self_description describes the features of the UIO, which specifies what or who the UIO is, and what kinds of abilities it has. This part of description is predefined and usually quite stable. Self_description can be defined as a tuple of the following format.

[UIO_ID, attributes-list (name, value, privacy)]

UIO_ID is related to the abilities of "being identified". It is globally unique and defined by the *UIO naming protocol*, which specifies the organization of UIOs using some naming strategies. Every attribute in the attribute list has a name, value, and privacy class. The privacy class can be public_RW, restricted_W_public_R, or restricted_R.

OTA_description is related to the UIOs abilities of observing, thinking, and acting. It defines two types of data. The first type of data is about the information that UIOs can obtain. The obtained information specifies what UIOs "observe" and "think". It can be the raw data captured from the sensors in a UIO or the processed results from the UIO. The second type of data is about the information that UIOs can affect by performing a specific action. The OTA_description can be defined as a tuple of the following format.

[Information_type, Information/Action_values, Report/Action_interval, Privacy]

Here, the privacy class can be public_R or restricted_R. Report_interval indicates the interval for value reporting. Action_interval indicates the interval between consecutive actions. The setting of the Information/action_values depends on the information type and the status of the UIO. For example, suppose a UIO is equipped with a thermometer sensor perceiving the temperature information. If the sampling rate of the temperature information is low, the corresponding Report_interval can be long. In addition, the status of the UIO will affect the Report_interval. An example is a UIO equipped with a GPS sensor. When the UIO is moving faster, the value of Report interval should be shorter to refresh the information timely.

Relationship_description describes the relationships between the UIOs themselves, and between UIOs and other other virtual/physical objects. It can be defined as a tuple of the following format:

[Out_set (Relationship_type, Objects_refernces), In_set (Relationship_type, Object_refernces)]

Out_set contains all the objects that the UIO are aware of, while In_set contains all the objects that "know" this UIO. The *Objects_references* represent the object that has definite *Relationship_type* with the subjective UIO. Relationship_description traces the UIOs relationship.

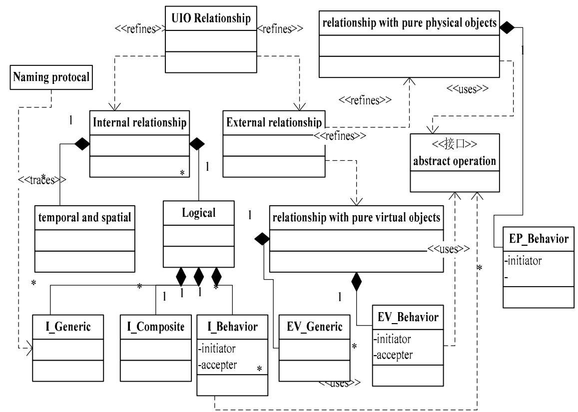

Figure 6 The UIOs relationship model

Figure 6 describes the UIOs relationship model. We refine the UIOs relationships into two sub-categories: external relationships and internal relationships. An internal relationship refers to the relationship between the UIOs themselves. It can be a temporal (e.g. "before" or "after") or spatial relationship (e.g. "near" or "up to", etc), as well as a logical relationship including *I_Generic*, *I_Composite* and *I_Behavior*. *I_Generic* and *I_Composite* are static while *I_Behavior* is dynamic. The *I_Generic* relationship affects the *UIO naming protocol*. For example, if two kinds of UIO-plants are of the same species, i.e., they have the *I_Generic* relationship, and then their UIO-IDs will have at least one common field. *I_Composite* is related to the composite UIOs. For example, the relationship between a UIO-Pen-Cap and a UIO-Pen is a composite relationship. *I_Behavior* suggests the dynamic relationship between two UIOs such as "be used by" or "play with".

The external relationships describe the relationships between UIOs and other physical objects that are not modeled as UIOs, (hereafter referred to as non-UIO physical objects) and between UIOs and the virtual objects in the cyberspace. Because the non-UIO physical objects do not have the abilities of "being identified", the relationship between them and UIOs exists only if UIOs perform some operations on them. This kind of relationship is called *EP_Behavior*. The relationship between a virtual object and an UIO is called *EV_Generic* if the virtual object represents the abstract type of the UIO. On the other hand, the relationship between a virtual object and an UIO is called *EV_Behavior* if there is any interaction between the virtual object and the UIO. For example, the relationship between an abstract O-Dog and a UIO-dog can be *EV_Generic* because O-dog represents the abstract type of the UIO-dog, while the relationship between an O-dog that is sending information to the UIO-dog is a *EV_Behavior* relationship Finally, recall that in the *high level UIOs model* the UIOs relationship can be composite relationship, which is embodied as a collection of relationships here.

From Figure 6, we can see that the behavior related relationships are defined in terms of the abstract operations of UIOs. So, next, let us investigate the *UIO abstract operation model* which is another essential part of the *basic UIO model* (See Figure 4).

Figure 7 describes the UIO abstract operation model which uses the UIOs abilities or compound abilities to provide the operations (services) through the UIOs operation interface. The UIO operation binds to *UIOs operation scene*. A UIO operation scene is composed of several UIOs and UIOs relationships. The *UIO operation* provides a specified set of interfaces when the UIOs play specified roles in specified *UIOs operation scene*. Every *UIO role* has a set of rights, which can be refined into two types: the security rights and the primary rights. The concrete operations of UIOs depend on both the right and the UIO interface set. Take the UIO-Dog as an example. Suppose it can play two roles: pet and spy. When it plays as a pet, it should not provide the interface of getting sounding information for the sake of privacy, but when it plays as a spy, the above interface is needed.

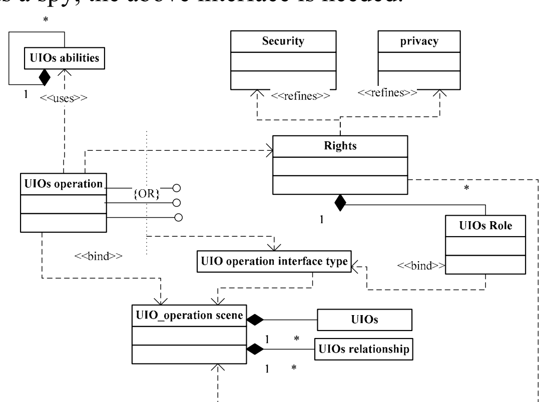

Figure 7 The UIOs abstract operation model

3.3 Implementation Issues and challenges

Recall that the ubiquitous devices are used to implement the UIOs. The devices may be electronic labels/tags, RFIDs, MEMS devices, tiny sensors, etc (See Figure 8), with virtual aid objects residing in them. However to bring the UIOs from the concept level to the implementation level still faces many challenges.

The first challenge comes from the requirement of integration of devices with heterogeneous hardware and software. As seen in Table 1, each kind of ubiquitous device can provide a set of additional intelligent abilities of UIOs, and each of them has some advantages and disadvantages in terms of intelligence support. How to integrate one or more devices together with a physical object, to get a monolithic UIOs entity should be carefully considered. The possible solutions require appropriate integration techniques from multi-layers, such as from the underlying network infrastructure and the embedded operating systems.

Figure 8 Devices to implement UIOs

The second challenge comes from the requirement of high usability. First, the ubiquitous devices are always resource constrained. How to make use of the resources as much as possible to provide adequate ubiquitous intelligence to the UIOs is an important issue. Second, it is necessary to improve appropriate mechanisms to reduce the harmful effects that may be brought by the supporting devices of UIOs to the human being.

Table 1.The UIOs abilities provided by devices

Devices and platform	Intelligent Abilities	Implementation Characteristic
RFID	Be identified	Simple and small Constrained single hop Communication
Wireless Sensor (Motes)	Be identified Observation Think	Multi-hop communication Constrained power Middle size
PDA or other kinds of emended devices	Be identified Think	Big Powerful Expensive

Another challenge comes from the requirement of high dependability. UIOs are more privacy and security sensitive than the traditional virtual objects. Implementing the privacy protocols of the UIOs model requires supporting techniques from both the hardware and software levels.

4. UIOs Oriented Application

The UIO models provide us not only the views to understand UIOs but also new approaches to developing ubiquitous computing applications. We can explore new kinds of ubiquitous applications based on the UIOs model. In this Section, we present an on-going project in developing an application called *ubiquitous searching*. The key idea of *ubiquitous searching* is to extend the searching from the cyber world to the physical world, acquiring, organizing, and browsing the desired information about the objects in the physical world, and navigating from one object to others through their contextual links, just like what we do in the web searching using *Google* or *Yahoo.*

Let us look at a scenario of ubiquitous searching. Jack starts the search using the keywords "dog, black", as well as his personal information such as "Jack, password". He will then obtain all the information (with restrictive privacy) available in the UIE about black dogs, such as "belonging to whom" "location", "near to what", etc., ranked by the pre-defined order using some ranking algorithm. Jack can then select a particular dog for further information. He may find that the dog is near a cat and, if he is interested in the cat's information, he can simply click on the link for that "cat" object.

To develop the ubiquitous searching application, we can start with building the model, using the modeling techniques proposed in the previous section. Due to limit in space, here we only summarize how the modeling techniques can help design and realize the application.

The *UIO abstract data model* defines the fundamental generic information structure of searching objectives as well as that of the search results. The *UIO relationships description* helps reserve the "links" among UIOs and the "links" among UIOs and virtual objects, providing the necessary information about "navigating from one UIOs to others" during the browsing process. The *UIO relationships model* helps capture and organize the searched results. For example, in the ranking algorithm, we can define a ranking strategy based on the "spatial relationship" to list the results by the distance between the UIO under search and the UIOs found in the result list. The *UIO abstract operation* model helps protect the privacy and provide the security of searched UIOs during the information gathering process of *ubiquitous searching.*

Currently, the ubiquitous search project is under development. We have achieved some primary results including the design of ubiquitous searching framework (USF), the design and performance evaluation of two key related algorithms, as well as a proof-of-concept prototype [3].

5. Conclusions and Future Works

This paper serves as the first step of conceptualizing and modeling UIOs, as well as exploring future UIO-oriented ubiquitous applications. We have provided insights into the principles of UIOs modeling, and described a reference framework of modeling UIOs. We have also discussed the implementation challenges, and presented a UIOs oriented application to show the impacts of UIOs, with the hope to inspire more research on the topic.

What we have done is just the beginning toward the modeling, design and implementation of UIOs. In our future work, we will enhance the current work in several aspects. We will further improve the proposed models by addressing more issues, such as semantics of composition (decomposition) of UIOs, and the collective behaviors of UIOs based on the social networking theories [11]. We will also develop the exiting UIO-oriented application prototype into a test-bed. Finally, based on the model, we will build application development tools.

Acknowledgement

This work is supported by Hong Kong Polytechnic University under the ICRG grant G-YE57 and the large equipment fund G.61.27.D01B

Reference

[1] D. Estrin, R. Govindan, J. Heidemann (Editors):"*Embedding the Internet*", Communications of the ACM, Volume 43, Number 5. May 2000.

[2] Runhe Huang, *"Intelligence Pool and Sharing"* , Computer and Information Sci., Hosei Univ. Japan

[3] Miaomiao Wang, Jiannong Cao, Yan Sun, "*Toward Ubiquitous Searching",* The 13th International Conference on Parallel and Distributed Systems (ICPADS'07) (to be appeared)

[4] Karen Henricksen, Jadwiga Indulska, and Andry Rakotonirainy, "*Modeling Contex Information in Pervasive Computing Systems"*,Pervasive 2002, LNCS 2414,pp.167-180, 2002.

[5] Gabriella Castelli, Alberto Rosi, Marco Mamei, Franco Zambonelli, "*A Simple Model and Infrastructure for Context-aware Browsing of World*", in the proceeding of Pervasive computing, (Percom07),2007.

[6] Nickerson BG, Sun Z, Arp JP (2005) "*A Sensor Web Language for Mesh Architectures*". 3rd Annual Communication Networks and Services Research Conference, May 16-18, 2005, Halifax, Canada.

[7] Tao V, Liang SHL, Croitoru A, Haider Z, Wang C, *"GeoSWIFT: Open Geospatial Sensing Services for Sensor Web*". In: Stefanidis A, Nittel S (eds), CRC Press, pp.267-274. 2004.

[8] Reichardt, Sensor Web Enablement: "*An OGC White Paper*". Open Geospatial Consortium (OCG), Inc.

[9] Ann Wollrath, Roger Riggs, and Jim Waldo, *"A Distributed Object Model for the Java System"*, Proc. of the USENIX 1996 Conf. on Object-Oriented Technologies, Toronto, Canada, June 1996.

[10] Jean-Jacques Laffont, David Martimort, *"The theory of incentives: the principal-agent model"*, 2002.

[11] Elizabeth F.Churchill, Christine A. Halverson, "*Social Networks and Social Networking*", IEEE internet computing.