Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 67 (2015) 409 – 418

Procedia Computer Science

6th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Infoexclusion (DSAI 2015)

Use of sociology concepts as the basis of a model for improving accessibility in Smart Cities

Vitor Santos[a]; Carolina Santos[b]; Tiago Cardoso[c]

[a] , NOVA IMS -Nova Information Management School, Lisboa, Portugal

[b] National School of Public Health, Nova University of Lisbon, Lisboa, Portugal

[c] Faculty of Science and Technology, Monte da Caparica, Portugal

Abstract

The use of mobile devices and their use for Internet access, for georeferentiation and services consumption had a huge increase. Today, these devices ability to establish cooperation networks and to interact intelligently and cooperatively with the surrounding environment has growing importance.

In this paper, we propose a system, based on a model which uses the social paradigms in mobile ubiquitous interactions, capable to effectively connect and integrate these devices in order to improve the accessibility in the cities. The model is inspired in concepts from the theory of organizations and sociology, as the notions of "social role", "ownership" and "responsibility", to be incorporated in each device.

We present an example for applying the model in the context of accessibility in smart cities for the pedestrian traffic light automatic management.

Peer-review under responsibility of organizing committee of the 6th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion (DSAI 2015)

Keywords: context-aware computing, mobile computing, organization theory, knowledge representation;

1. Introduction

Mobile and ubiquitous computing may be characterized through the ubiquity of communications and devices with computational power, that become an integrating part of the physical space in which we live, as well as the various

Peer-review under responsibility of organizing committee of the 6th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion (DSAI 2015)
doi:10.1016/j.procs.2015.09.286

activities in our day-to-day lives [3]. To Schmidt, the way people interact with devices is paramount for their success [6].

In a time where mobile devices usage is widespread and its usage for the internet, georeferenced (Global Positioning System- GPS) and for the use of services is expanding, the ability of these devices to establish cooperation networks and interacting in an intelligent and collaborative way in the surrounding environment is growing. The idea of taking advantage of this context to improve the accessibility in the cities is interesting.

Mobility of devices comes mainly from the mobility of its carriers, by so originating a constant change of the informatics environment that surrounds the device. In the case of the cities there is a high probability of a citizen be a device mobile carrier. More so the availability of public and private wireless network access, as well as ad hoc connections to other devices, provides opportunities for using devices in the cities in integrated ways.

To understand and capture the contexts automatically, in what is commonly named context sensible computing, to participate and cooperate with different context member elements, to supply and use services and information, seems relevant.

Nowadays impaired people face frequently challenges in the assessment to public spaces, public and private infrastructures, transports, services. The main objective of this article is to propose a system, based on a model which uses the social paradigms in mobile ubiquitous interactions [9], capable to effectively connect and integrate these devices in order to improve the accessibility in the cities. This kind of system may increase people awareness about environmental risks, raise the accessibility to different kinds of services, prevent several ways of social discrimination and exclusion, so improving the overall quality of life of people with special needs. The model is inspired in concepts from the theory of organizations and sociology, as the notions of "social role", "ownership" and "responsibility", to be incorporated in each device.

The core idea is thinking on what role the device may perform in each system and their relation with other devices, so as to make it possible to be dynamically integrated in those systems, and the cooperation with other elements belonging to the system. The model for the use of social paradigms in mobile ubiquitous interactions main characteristics are explained in section 3.

We wish that the system be robust enough to be able to tolerate different kinds of devices, to tolerate performance flaws and to allow structural and ownership alterations to the roles, in a way it becomes resilient to environmental dynamics.

This document is organized as follows: section 2 presents an overview of mobile computing, ubiquitous computing and context sensible computing and section 3 a model which uses the social paradigms in mobile ubiquitous interactions. Section 4 presents the application of the model in the context of accessibility in smart cities. Finally, section 5 presents some conclusions and future work.

2. On mobile computing, ubiquous computing and context sensible computing

We commonly call mobile computing the use of small dimension computer devices and laptops on wireless networks, connected to public and private servers, to the internet or other devices. Among these computing devices are laptops, notebooks, tablet PCs, palmtops and personal assistants (PDAs).

Ubiquitous computing is a way to improve computer usage, making many computers physically available and making them effectively invisible to the user [7]. Ubiquitous computing has as main objective to make person-machine interaction invisible, be it, to integrate in a whole informatics and people's natural actions and behaviors [8]. By invisibility we mean to be able to interact with computing systems without realizing they are machines, rather as if one were talking with another person. In ubiquitous computing we assume that surrounding computing systems are proactive, and are connected, or are permanently trying to connect. This characteristic is often called "omnipresence".

Context sensitive computing (Context-Aware Computing) appeared as ubiquitous computing branch that studies the connection between environmental and informatics systems changes. Dey et al [2]. It is a recent investigative area with difficult implementation techniques challenges, and one that has caught the attention of investigators everywhere in the world.

In context sensible computing, the devices try to understand and capture automatically the surrounding contexts so as to provide a better interaction between the environment and the user, regarding hardware, software, and or communication [1].

3. A Model for the use of social paradigms in mobile ubiquitous interactions

In order to reach the dynamic integration of a device in a context sensitive computing system, with a pre-defined formal structure, it is necessary that the device has a minimal set of functionalities and a representation of different formal structures of the different context sensitive computing systems where it might fit.

The representation of the system's formal structure we propose in this article is based on the concepts of Role, Ownership and Responsibility that are liable to be reused in different computing contexts [5].

We call Role the particular connection of a device to the cooperative structure of a system that establishes, in that system, that determines a certain number of obligations and responsibilities to the device; Ownership will be the association of a device to a role to perform in the system; we call Responsibility task association to roles that bind role holders as responsible for the full task fulfilment, regardless of that fulfilment being assured by themselves or any other device in which the task execution is delegated.

Ownership, (role/roles that are performed by the device and those it interacts with in a given context, can be represented as shown in Chart 1.

Ownership is a relation between a device and a role that can be expressed under the form: Owner (X,P1) where X is the device and P1 is the role performed by the device. A device can own more than one role as long as it implements *per si* all the required functionalities for the correct performance of all roles. On the other hand, there may be more than one device that owns the same role.

CHART 1 – OWNERSHIP CHART

Ownership	
Role	***Device***
Device X	Role 1
Device Y	Role 2
Device Z	Role 3
Device X	Role 4

When a device owns a given role, a competency principle is admitted: the device implements *per si* all the functions that are required for the correct performance of the role(s). This means the device has the ability of executing all the necessary functions to fulfil the tasks it's responsible for in the role(s) it owns.

Competency to execute a task associated to a role can be defined in the following way:

1. Being Capac_Exec(X,F1) the ability to execute from X in order to execute the function F1
2. Being Execute(P1,T1) the Responsibility to Execute the task T1 attributed to role R1
3. Being Owner(X,P1) the relation of ownership of X to execute the function F1

So, if F1 is a function that pleases T1 we may conclude that the device X is competent to execute the task T1. The relation F1 pleases T1 can be expressed by: Pleases(F1,T1)

Formally, the competency to execute a task associated to a role is translated as:

If $Capac\_Exec(X,F1) \wedge Execute(P1,T1)$

$\wedge Pleases(F1,T1) \Rightarrow Competent\_Exec\_Task(X,T1)$~

The competency to execute all the tasks associated to a role- Principle of competency for the performance of a role is translated as:

$\forall_i \in n\ (Execution(P1, Ti) \Rightarrow Competent\_Exec\_Task(X,Ti)) \Rightarrow Competent\_Exec\_Papel(X,P1)$

In the Conception of the knowledge representation model there is the possibility of a device belonging simultaneously to more than one system, to be able to change roles and to perform more than one role within the same context. Therefore, in order for the model to support this possibility, we need to contextualize the ownership representation as illustrated in chart 2.

CHART 2 – EXTENDED OWNERSHIP CHART

Extended Ownership		
Role	***Device***	***Context***
Device X	Role 1	Hive Context
Device Y	Role 2	Hive Context
Device Z	Role 3	Hive Context
Device X	Role 4	Hive Context
Device Y	Role 1	Friends Context
Device Z	Role 6	Friends Context

Ownership is a relation between a device, a role and a context, expressed formally as: Owner(X,P1,Ca) where X is the device, P1 is the role performed by the device and Ca the context in which X owns the P1 role.

A device can own roles with the same name in more than one context. However, by considering a model that includes more than one context, the definition of the roles is contextualized, so the roles can have the same name in different contexts and have different aims, depending on its definition for each specific context.

The fact that different devices can be owners of the same role provides the model with the required hardiness to compensate performance flaws and to admit structural changes and role ownership.

In Figure 1, we present, schematically a simplified model of the representation structure of knowledge relative to a given context.

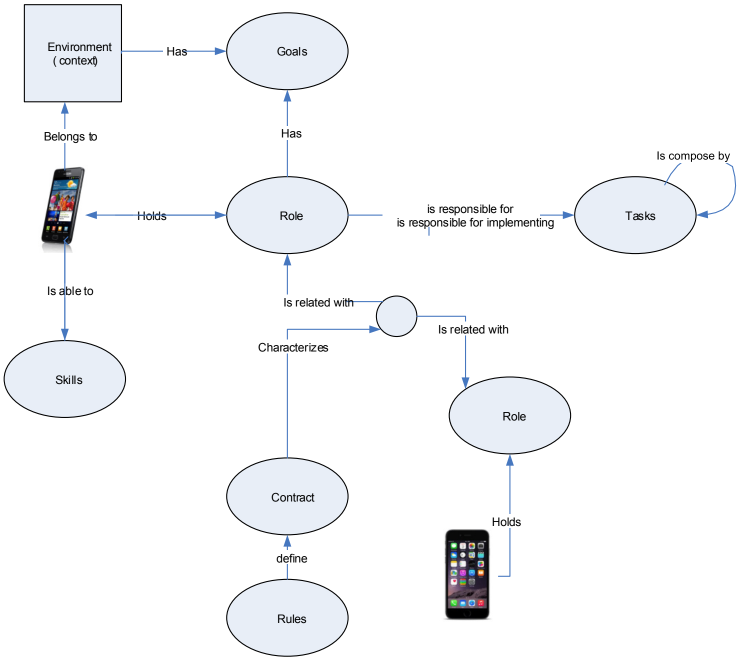

Figure 1- Simplified Model

The association between tasks and roles that exist in a given context, where Owners are responsible for their fulfilment, regardless of it being assured by themselves or others in which the task is delegated, is called Responsibility.

CHART 3 – RESPONSIBILITY CHART

Responsibility		
Role	***Task***	***Context***
Role 1	Task 4	Hive Context
Role 3	Task 3	Hive Context

Responsibility is a relation between a role and a task, in a context Ca, and may be formally defined as:

Responsibility(P1,T3,Ca)

In Chart 3, "Task 4" is of the responsibility of the device owner of "Role 1", but the latter may choose not to perform it directly. In order to do so, it delegates the task, which means it must obligatorily know which roles are responsible for "Task 4".

Another designation is "Execution Responsibility"- Tasks associated to roles existing in Context, for which the Owners are responsible to ensure the fulfillment in terms of execution. In the presented model, being responsible for the execution by a task doesn't consider this task to be composite.

CHART 4. RESPONSIBILITY OF EXECUTION CHART

Responsibility of Execution		
Role	***Task***	***Context***
Role 1	Task 1	Hive Context

Responsibility of Execution		
Role	***Task***	***Context***
Role 2	Task 4	Hive Context
Role 2	Task 2	Hive Context
Role 3	Task 2	Hive Context
Role 3	Task 3	Hive Context

Responsibility of Execution is a relation between a role and a task, in a context Ca, and is normally defined as: Execute(R1,T1,Ca).

In Chart 4, T2, T4 and T5 are atomic tasks, i.e. they have no defined Break Down:

Execute(R1,T1,Ca), Execute(R1,T2,Ca).

"Task 2" may be executed by the Owner of "Role 2" or the Owner of "Role 3".

A device is responsible for a given Task if it's responsible for its fulfillment, regardless that fulfillment is assured in terms of execution by itself or a "subcontract" of other devices.

A device is responsible for executing a given task if the task length is assured in terms of execution by the device. A particular aspect occurs when the same device performs a role where it is simultaneously responsible for the fulfillment and execution of a task. Responsibility(R1,T1,Ca) Λ Execute(R1,T1,Ca).

In this case, we must define which of the two relations is stronger and overlaps the other. The definition of this criteria must be programmed.

In the case where the strongest relation is Execute(R1,T1,Ca) then the task is executed and the Responsibility(R1,T1,Ca) is overlooked.

In the case the strongest relation is Responsibility(R1,T1,Ca), depending on the criteria used by the device in order to choose between executing or delegating, the relation Execute(R1,T1,Ca) may or not be used.

If the criteria is to delegate the task in a device that is responsible for the execution, then it is possible that the device will delegate the task onto another one or on itself, once it is also responsible for the execution. In this case the relation Execute(R1,Y1,Ca) may or not be used.

If the criteria is to check first whether the device is responsible for the execution and only delegate in case it isn't, then the relation Execute(R1,T1,Ca), if it exists, is used.

By trying to characterize the delegation, three important questions emerge [4]:

-What is the nature of the relations between which delegates and which accepts the relation;

-Through which types of communication can this delegation be made and how is it specified;

-Under which conditions is it possible to say the delegation was achieved successfully.

The answer to the first two questions comes from what is specified in the model itself. The answer to the third question is provided by using two mechanisms, used together or in separate. The first mechanism is based on message exchanges and between the device that delegates and the one the delegation is made onto.

Some tasks can be broken down into elemental tasks, as shown in Chart 5.

TABLE 5. BREAK DOWN CHART

Break down			
Task	***Subtask***	***Order***	***Context***
Task 1	Task 1.1	1	Hive Context
Task 1	Task 1.2	1	Hive Context
Task 3	Task 3.1	1	Hive Context
Task 3	Task 3.2	2	Hive Context
Task 3	Task 3.3	2	Hive Context
Task 3	Task 3.4	3	Hive Context

Tasks 3.2 and 3.3 can only be executed after the task 3.1 is finished. There is no particular execution sequence between them. However, task 3.4 can only be executed after tasks 3.2 and 3.3 are both concluded.

Breaking down tasks is the relation between a task and others that break it down. It may defined as follows:

Break down(T3,T3.1,1,Ca)
Break down(T3,T3.2,2,Ca)
Break down(T3,T3.3,2,Ca)
Break down(T3,T3.4,3,Ca)
….

To each relation between roles a certain Contract is determined (set of rules). A relation between two roles is always one-way (Ex: "Role1" to "Role 2"). The definition of bi-directional relations is achieved through two one-direction relations, in opposite ways.

A relation between roles is defined by the expression Relation(R1,P2,Contract,Ca) which means there is formally a relation in context Ca, between the roles R1 and R2, from R1 to R2. This formal relation is, in this model, called "Contract".

TABLE 6. RELATION CHART

Relation			
Role	***Role***	***Contract***	***Context***
Role 1	Role 2	Contract A	Hive Context
Role 2	Role 1	Contract C	Hive Context
Role 3	Role 2	Contract B	Hive Context
Role 4	Role 3	Contract C	Hive Context
Role 5	Role 3	Contract A	Hive Context

A contract is defined by a set of rules. Rules define the contract relative to the interactions between roles it is associated with, namely, in what concerns processing the various types of message. We will formally come up with:

Caract(ContractA,Rule1); Caract(ContractA,Rule2); Caract(ContractB,Rule1); ….

TABLE 7. RELATION CHART

Contracts	
Contract	***Rule***
Contract A	Rule 1
Contract A	Rule 2
Contract A	Rule 3
Contract B	Rule 1
Contract B	Rule 4

For each message sent from an emitter to a receiver, the latter makes an applicability test to the kind of message in question, the rules associated between emitter-receiver.

This check is achieved in two phases. In phase one, when check if the rule, according to the relation between the emitter and the receiver, is applicable in the given context.

The messages between devices have the following format: Msg(D1,D2,Tm,C)

Where D1 is the emitting device, D2 is the receiving device, T is the type of message and C is the contents.

The general condition for applying a Rule r in treating a message, as "Tm", send from X to Y, in context Ca, can be expressed this way:

msg(X,Y, Tm,C) Λ Owner(X,R1) Λ Owner(Y,R2) Λ Relation (R1,R2,) Λ Caract (ContractA, r)

If this check fails, we go on to the second phase, where we execute a check to the condition for applicability of the rule in a given context.

The general condition for the applicability of a Rule r in any given context that determines a given message treatment as “Tm” can be seen in the expression:

msg(X,Y, Tm,C) Λ Owner(X,R1) Λ Owner(Y,R2) Λ Relation (R1,R2, ContractA) Λ
Caract (ContractA, r)

4. Applying the model in the context of accessibility in smart cities

There are many possible areas to apply this model in the context of accessibility in smart cities. For example: access to public administrative services, interaction with traffic lights for pedestrians on the public highway, going to the hospital, etc.

Let’s consider, for example, the interaction of a blind person with traffic lights for pedestrians on the public highway (pedestrian traffic light X) that makes available their formal structure of computation, and holding a mobile device with computational skills.

In this case the representation structure of knowledge to be transferred to the mobile device may be as the one presented in charts 8 to 12:

CHART 8 - OWNERSHIP CHART

Extended Ownership		
Device	***Role***	***Context***
Pedestrian traffic light X	Manage the light switch times	Pedestrian traffic lights
Pedestrian traffic light X	Manage light intensity	Pedestrian traffic lights
Pedestrian traffic light X	Manage sound	Pedestrian traffic lights
Mobile Device blind person	Identify blind person	Pedestrian traffic lights
Mobile Device deaf person	Identify deaf person	Pedestrian traffic lights

CHART 9- EXECUTION RESPONSIBILITY CHART

Execution Responsibility		
Role	***Task***	***Context***
Manage light intensity	Change the intensity of light	Pedestrian traffic lights
Manage sound	Connect beeps	Pedestrian traffic lights
Manage the light switch times	Change the light switch times	Pedestrian traffic lights
Identify blind person	Submit blind ID	Pedestrian traffic lights
Identify deaf person	Submit deaf ID	Pedestrian traffic lights

CHART 10- BREAKING DOWN CHART

Breaking Down			
Task	***Subtasks***	***Order***	***Context***
Manage sound	Turn on sound -	1	Pedestrian traffic lights

Breaking Down			
Task	***Subtasks***	***Order***	***Context***
	green light		
Manage sound	Keep the sound for x seconds - green light	2	Pedestrian traffic lights
Manage sound	Accelerate the pace of sound in the last y seconds - green light	2	Pedestrian traffic lights
Manage sound	Turn off sound - green light	3	Pedestrian traffic lights
Manage sound	Turn on sound - green light	4	Pedestrian traffic lights

CHART 11- RELATION BETWEEN ROLES

Breaking Down			
Role	***Role***	***Contract***	***Context***
Identify blind person	Manage sound	B	Pedestrian traffic lights
Identify deaf person	Manage light intensity	D	Pedestrian traffic lights

CHART 12- CONTRACT CHARACTERIZATION CHART

Contract Characterization	
Contract	***Rule***
B	Rule 1- Receive and confirm blind request
B	Rule 2- Activate Pedestrian traffic lights sound system
B	Rule 3- Activate Pedestrian traffic lights sound system
D	Rule 4- Receive and confirm deaf request
D	Rule 5- ….

Apart the inclusion in each device of this knowledge representation structure we also need to include methods that implement contract-related rules. We show below, as example, a possible rule for Activate Pedestrian traffic lights sound system:

Rule 2- Activate Pedestrian traffic lights sound system

Description: To Activate Pedestrian traffic lights sound system in the context of the (Pedestrian traffic lights)

/ verify if the responsibility it has on the task is correct/

If type= 'Execute'* / If Execution Responsibility/ *

Validates if competent /if error: Answers origin with msg "I cannot do the task"; end algorithm

Or

Answers origin with msg type "Info" I can accept the task"

Check the Pedestrian traffic lights current state (green or red)

Activate Pedestrian traffic lights sound system

Chose the correct sound (for green or red status)

End If

5. Conclusion and future work

This paper describes a system, based on a model which uses the social paradigms in mobile ubiquitous interactions, capable to effectively connect and integrate these devices in order to improve the accessibility in the cities. The model, structured on concepts originated from organizations and sociology theories, brings with a minimal of functions and information to be incorporated in each device.

With the transcribed proposal in this article, we aim to contribute to the debate on accessibility in smart cities and on mobile and context-sensible computation, by proposing a system that allows us to integrate dynamically devices in computerized systems distributed in a pre-defined formal structure.

As future work, we aim to extend the system to the use of more devices and types of disabled people an effective development of a system that implements the suggested mode and the building of a website where it is possible, using this system, to register and obtain information on the valence of different devices and, to define and import formal structures on smart cities computing systems.

References

[1] Dey A. K. and G. D. Abowd, (1999) "Towards a Better Understanding of Context and Context-Awareness," College of Computing, Georgia Institute of Technology, Atlanta GA USA, Technical Report GIT-GVU-99-22

[2] Dey, A. K. (2001), 'Understanding and using context', Personal Ubiquitous Comput. 5(1), 4–7.

[3] Greenfield, A. (2006), Everyware : The Dawning Age of Ubiquitous Computing, New Riders Press. URL: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0321384016

[4] Norman, Timothy J.; Preece, A.; Chalmers, S.; Jennings, N. R.; Luck, M.; Dang, V. D.; Nguyen T. D.; Deora, V.; Shao J.; Gray W. A.; Fiddian, N. J.. (2004) Agent-based formation of virtual organisations. Knowledge-Based Systems, 17:103-111.

[5] Santos, V. (2008) "Criação dinâmica de empresas virtuais com estrutura formal pré-definida" – 8ª Conferência da Associação Portuguesa de Sistemas de Informação

[6] Schmidt, A. (2000) "Implicit Human Computer Interaction Through Context" Personal Technologies Volume 4(2&3), pp191-199

[7] Weiser, M. (1993), 'Some computer science issues in ubiquitous computing', Commun. ACM 36(7), 75–84

[8] Wikipedia (2015). "Computação ubíqua" Retirado em 18 de Março de 2015 de http://pt.wikipedia.org/wiki/Computação_ubíqua.

[9] Santos, V, (2015) "Use of Sociology Concepts as the Basis of a Model for Context-Aware Computing", International Journal of Web Portals