RAP – A Basic Context Awareness Model

Ioan Salomie, Tudor Cioara, Ionut Anghel, Mihaela Dinsoreanu
TechnicalUniversity of Cluj-Napoca
Ioan.Salomie@cs.utcluj.ro

Abstract

This paper addresses two fundamental research problems in the domain of context sensitive systems: the development of a generic context model that can be used to represent general purpose contexts in a computer interpretable way and the context model management. The context model is represented using a triple set consisting of context resources, actors and policies. The model is mapped onto real contexts by populating the sets with context specific elements. A context situation to which a context aware system must adapt is represented by a specific context model instance. To ease the context reasoning and adaptation processes, a core ontology is defined to represent the relationships between the context model concepts. The core ontology is extended with domain specific concepts as ontology sub-trees. For the context model management problem we propose an agent based solution using BDI agents.

1. Introduction

The context-sensitive systems domain is a dynamic direction of the pervasive computing research efforts. According to Mark Weiser [1], the domain of pervasive calculus and context sensitive systems aims at developing intelligent environments which encapsulate sensor type physical entities and computing components, through the union of the material world and the digital world. Following these ideas, living and working spaces, study halls, airports, vehicles and even entire cities will incorporate and integrate sensors, intelligent devices, fixed and mobile computational resources which offer / consume services. All intelligent devices or resources must be capable of learning and adapting their behavior dynamically according to the context in which they evolve.

The objective of this paper is to define a context model for accurately representing general purpose real contexts so that context awareness is achieved. The context model is based on representing actors, resources and policies from the real world and uses BDI (Believes Desires Intentions) agents for context management and processing.

2. Related work

The research efforts in this domain are concentrated on two directions: the study and development of generic models that can be used to represent real world contexts and the development of context management models.

In [2] the concept of *multi-faceted entity* is defined for modeling and representing the set of context properties. A *facet* belonging to this set is represented as the effective values of context properties at a particular moment to which the context sensitive application, has access. The disadvantage of this approach is given by the lack of semantic information encapsulated in the concept of facet.

An original approach to the context modeling research problem is the use of parametric state machines to represent a context aware system [3]. The context is modeled using context functions that can modify the context aware system state. The main disadvantage of this approach is given by the high complexity of the parametric state machine (as number of states and transitions) associated to a real system.

The use of the XML standard and ontology for context modeling and representation is a new direction described in the domain literature. Within the context model, the context properties are specified using XML tags and is instantiated with actual values of the context properties (captured by sensors) during execution [4]. The context properties are represented as ontological concepts and the developer is being responsible for ontological concepts modeling. The advantage of such an approach is given by the fact that some relations between the context properties, for example context dependencies, can be easily modeled using ontology.

In [5] O'Connor proposes the construction of a system situation space and representing the system execution context as a situations group in the space. A function can be built that takes values from the set of situations that form the context, with values in the action set that the system executes. Using learning

978-1-4244-2673-7/08/$25.00 ©2008 IEEE

algorithms, the system can deduce which action to execute when a new situation appears by placing it in a group in the situations space.

The researches in the direction of context management are concentrating upon the definition of methods for the instantiation of the context model with specific information from the real world, for keeping the consistency of context instances with the real context and for the administration of the interaction between the context and applications.

In [6] the context information is classified in three categories: *user context information, physical context information* and *computational context information,* and different models are proposed for capturing and managing the context information for each category.

The capturing models of the context specific data, which are ontologically guided, are proposed by [3], [4]. Using prior defined ontology, the captured context specific data could be used to infer or to learn new context information. In [7] the use of reusable components for updating context specific data is proposed. These components generically named *conextors* provide stable communication channels for capturing and controlling context specific data. In [8] Spanoudakis propose the development of behavioral models guided by the context, which can allow context aware applications to detect only those context data variations that lead to the modification of their behavior.

3. RAP – The Basic Context Model

The **basic context model** is defined as a triple $\mathbf{C} = \langle \mathbf{R}, \mathbf{A}, \mathbf{P} \rangle$ where: **R** is a set of context resources; **A** is a set of actors which interact with context resources; **P** is a set of real context related policies.

The context model is mapped onto different real contexts by populating the sets with real context specific elements. The mapping result is a **specific context model** $\mathbf{C_S} = \langle \mathbf{R_S}, \mathbf{A_S}, \mathbf{P_S} \rangle$ (see also figure 1). A specific context model accurately reflects the real context and should be permanently kept consistent with the real context.

The relationships between the context model elements are represented by using *is-a* type relations in a general purpose context ontology core. The specific context model concepts are represented as sub trees of the core ontology (see figure 2).

An **instance of the specific context model** $\mathbf{CI_a^t} = \langle \mathbf{R_a^t}, \mathbf{P_a^t} \rangle$ is used for describing the actor – context interaction and is defined as a specific context model projection onto a certain actor. An instance contains the set of resources with which an actor can interact, together with their values at a specific moment of time *t*. These values are represented as instances of the context specific ontology.

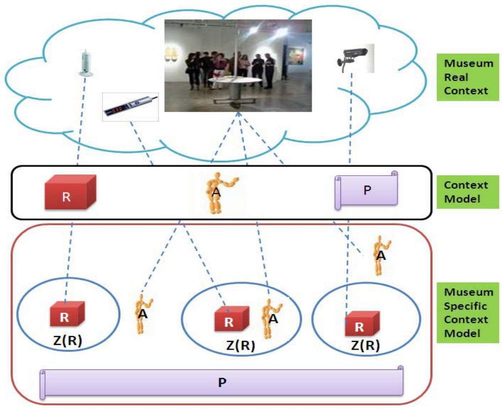

Figure 1. Context model elements

An actor interacts with the real context only through the specific context model. The following sections detail the context model main concepts.

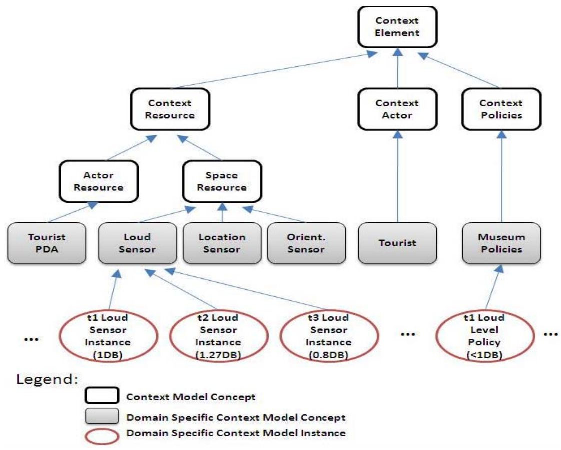

Figure 2. The context model ontology

3.1. Resources

A **context resource** is a physical / virtual entity which generates and / or processes context information. In a real context, we have identified *passive* and *active* resources. The **passive context resources** aim at capturing and storing context specific data while the **active context resources** can interact directly with the context and modify the context state.

The set of context resources (**R**) can be separated in two disjunctive subsets ($\mathbf{R} = \mathbf{R_A} \cup \mathbf{R_S}$): (i) the set of context resources attached to the physical space / environment in which actor interactions occur (R_S) and (ii) the set of context resources attached to the actors that provide actor interactions related information (R_A).

A context resource has a unique identity, can be annotated with semantic information and features *resource properties*, *resource services* and *resource influence zone*.

Resource Properties, K(r). It specifies the set of relevant context information that a resource can provide.

In our model, the context abstraction is represented by the set of all context properties in terms of relevant information provided by context resources:

$$\textbf{Context} = (C_S, K) = (C_S, K_S) \cup (C_S, K_A) \qquad (1)$$

where: (C_S, K) is an information system; **K** is the set of all context properties (generated by all context resources); K_S is the set of context properties generated by the context resources attached to the physical space; K_A is the set of context properties generated by the context resources attached to all actors interacting with the context model.

Resource Services, S(r). In the context model, the resource functionality is specified by services. The resource services are exposed by publishing them in a public registry such as UDDI. The actors will interact with a context resource through its attached services.

Resource Influence Zone, Z(r). It is the 3D physical space (3D spatial bubble) in which a resource captures / provides context information or in which the resource presence can be sensed or in other words, it becomes visible for an actor or for another resource. The resource influence zone is specified in the context model by using the resource position in the real space and the resource range taken from its specification.

The influence zone of a context resource attached to an actor (i.e. a resource from the R_A set) is the zero volume space.

$$r \in R_A \Rightarrow Z(r) = 0_v \qquad (2)$$

Relation (2) implies that a context resource attached to a certain actor offers relevant context information **only for the actor - context** interaction.

The influence zone for a context resource that is attached to the physical space (i.e. a resource from de R_S set) is a non-zero volume space.

$$r \in R_s \Rightarrow Z(r) \neq 0_v \qquad (3)$$

Relation (3) states that a context resource that is attached to the physical space can offer relevant information for **many actor - context** interactions.

The influence zone for a context resource attached to the physical space is used to determine if the given resource is or is not part of a context. Two resources r_1 and r_2 attached to the physical space are part of different contexts if the relation 4 is verified.

$$Z(r_1) \cap Z(r_2) = 0, \text{ and}$$
$$!E\ r_3 \in R_S \mid (r_1) \cap Z(r_3) \cap Z(r_2) \neq 0$$
(4)

The context influence zone $Z(C_S)$ for a given specific context C_S is defined as the physical space in which an actor can interact with at least one resource attached to the context space. $Z(C_S)$ represents the environment as a 3D spatial bubble in which the actor - context interactions occur. The context frontier $Fr(C_S)$, is defined as the closed spatial area that wraps the context zone.

3.2. Actors

An **actor** represents a physical or virtual entity that interacts directly with the context or uses the context resources to fulfill its needs. The actor is a context information generator, has a unique identity and can be annotated with semantic information.

An actor is characterized by: (i) the actor's specific resources, (ii) the context request, and (iii) the actor-context contract.

Actor Resources, R_{ai}, specify the set of actor associated resources such as position elements, RFID tags, "augmented reality" elements that have been associated with the actor before it enters a specific context (operating room, museum etc.).

Context Request, $Req(a_i)$, specifies actor preferences related to the context with which it interacts.

Context Contract, $Ctr(a_i, C_S)$, stipulates the actor's rights and responsibilities within a specific context.

3.3. Policies

A **policy** represents a set of rules that must be followed by actors or resources present in the context influence zone.

The process of evaluating the degree of respecting a policy by a resource or an actor is achieved by evaluating the complying degree of all policy rules. If a rule is broken, the associated policy is broken, an exception is thrown and the actor that committed the fault is eliminated from the context.

4. Context model management – an agent based solution

Our context model features complex processes for management, administration and adaptation. We have chosen the BDI type of agents as processing units for model management due to their reactivity, collaboration, inference and adaptability properties.

In our management model four types of generic agents are defined and formalized: *Context Model Administering Agents, Context Interpreting Agents, Request Processing Agents* and *Execution and Monitoring Agents*.

1. Context Model Administering Agent (CMAA) is the specific context model manager. Its main goal is the synchronization of the specific context model (C_S) with the real context.

Other goals of this agent refer to identifying/negotiating the entry stage of an actor in the context and adding/removing resources to/from the context. In a context entry negotiation process: (i) the entering actor or resource, present its requests and properties to the CMMA agent and (ii) the CMAA agent presents the context services, properties and policies. By using negotiating strategies, the CMAA agent adapts the actor or the resource to the context and generates an actor-context contract that will drive the actor/resource evolution within the specific context model.

2. Context Interpreting Agent (CIA), is used to semantically evaluate the information of a context instance CI_a^{t+1} when an actor makes a context related request. Its goal is to correctly represent the semantic value of the context instance into a space of semantic states. The semantic states space is a semantic hyper-space whose dimension equals the number of context resources from the specific context model. The CIA agent builds the semantic space by using reasoning and learning algorithms, the context policies and the context ontology. For a context instance, by positioning all resources values on the semantic axes, the semantic value of a context instance is obtained as a unique hyper-point in the hyper-space.

3. Request Processing Agent (RPA), is used for processing actor requests in a specific context model. This agent identifies and generates the action plans that must be executed for serving the incoming requests. The RPA agent uses the semantic states space to get the semantic value for a context instance. This value is used for searching into an action plan repository to identify the proper plan to be executed or for generating a new plan.

4. Execution and Monitoring Agent (EMA) processes the plans received from the RPA agent and executes every plan action using the available services attached to the specific context model resources. After mapping plan actions onto services, a plan orchestration is obtained which can be executed using transactional principles so that the specific context model is kept in a consistent state regarding to the real context. If an error occurs in the execution phase of the plan, compensatory actions will be taken for restoring the initial state (before execution) to the specific context model and implicitly to the real context.

5. Conclusions and future work

The paper proposes the use of a triple set consisting of context resources, actors and policies to represent real contexts in a computer interpretable way and the use of BDI specialised agents as a context model management solution.

For the future development, we intend to extend the context model with new concepts that will facilitate the definition and integration of autonomic computing features which are necessary due to the complexity of the context management / administration / adaptation process.

Acknowledgements

This work was supported by the Food Trace project within the framework of the "Research of Excellence" program initiated by the Romanian Ministry of Education and Research.

6. References

a

[1] Mark Weiser, *The Computer for the 21st Century*, Scientific American, 1991

[2] Anca Rarau, K. Pusztai. I.Salomie, *MultiFacet Item based Context-Aware Applications*, International Journal of Computing and Information Sciences 3(2): 10-18

[3] Irene Y.L. Chen, Stephen J.H. Yang, Jia Zhang, *Ubiquitous Provision of Context Aware Web Services*, IEEE International Conference on Services Computing (SCC'06)

[4] Ki-Chul Lee, Jung-Hoon Kim, Jee-Hyong Lee, *Implementation of Ontology Based Context-Awareness Framework for Ubiquitous Environment*, Int. Conference on Multimedia and Ubiquitous Engineering (MUE'07)

[5] Neil O'Connor, Raymond Cunningham, *Self-Adapting Context Definition*, First International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2007)

[6] Paolo Bellavista, Antonio Corradi, Rebecca Montanari, *Mobile Computing Middleware for Location and Context-Aware Internet Data Services*, ACM Transactions on Internet Technology, Vol. 6, No. 4, November 2006

[7] Damien Fournier, Sonia Ben Mokhtar, Nikolaos Georgantas, *Towards Ad hoc Contextual Services for Pervasive Computing*, Middleware for Service Oriented Computing 2006, Melbourne, Australia

[8] George Spanoudakis, Khaled Mahbub *A Platform for Context Aware Runtime Web Service Discovery*, 2007 IEEE International Conference on Web Services (ICWS 2007)