A Generic Context Model with Autonomic Features

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

Abstract

This paper addresses two fundamental research problems in the domain of pervasive autonomic systems: the development of a generic context model that can be used to represent general purpose contexts in a system interpretable way and the autonomic context model management. The context model is generically 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 pervasive system must adapt is represented by a specific context model instance. To ease the context reasoning and adaptation processes, the context model concepts and relationships are represented using a core ontology. The increasing complexity of the pervasive systems, and the difficulties of their management, administration and adaptation have headed us towards the necessity of integrating autonomic computing paradigms in the context model management process.

1. Introduction and related work

The pervasive systems continuously monitor / capture and interpret the environment related information in order to assure high context awareness. All intelligent devices or resources part of a pervasive system must be capable of learning and adapting their behavior dynamically according to the context in which they evolve.

Due to the complexity and continuous evolutions of the environment where the pervasive systems are integrated and executed, their management has become extremely difficult. This headed us towards using the autonomic computing paradigms (self–configuring, self–healing, self–optimizing and self-protecting) for the development and integration of self-* enhanced components into the pervasive systems.

The *objective of this paper* is to define a context model with autonomic features for accurately representing general purpose real contexts, targeting the development of pervasive autonomic systems.

The researches efforts in the pervasive autonomic systems domain are concentrated on two directions: the development of generic context models that can be used to represent the environment and the development of autonomic systems that self-adapt to context changes.

In the **real world context modeling** research direction many approaches have been proposed. In [1] the concept of *multi-faceted entity* is defined for modeling the set of context properties. A *facet* is represented as the effective values of context properties, at a particular moment, to which the context sensitive application has access. The main drawback of this approach is the lack of semantic information encapsulated in the facet concept. As a result, inferring new context related knowledge is difficult.

An original approach to the context modeling problem is the use of parametric state machines to represent a context aware system [2]. The context is modeled using context functions that modify the context aware system's state. The complexity of a real system's associated parametric state machine, in terms of number of states and transitions, is the main disadvantage of this approach.

The use of XML together with ontologies is a new direction for context representation. The context properties are represented as ontological concepts in design time, and instantiated with actual values, captured by sensors, during execution [3] [4]. This way, the relations between context's properties are easily modeled using ontologies. The main disadvantage is the high degree of inflexibility determined by the human factor intervention in context representation.

In [5] O'Connor proposes the construction of a system situation space where system execution context is represented as group in this space. A function can be defined taking values in the context set of situations, with values in the system's action set. Using learning algorithms, the system may infer the action to be executed when a new situation appears by placing it in a situation space group.

Regarding the **context aware self-managing systems,** most of the researches reported in the literature are focusing on the self-adaptation problem. In this respect, research efforts are made to create new models and algorithms that allow computational systems to execute specific actions according to the context or situation at hand. The objective is to associate a certain degree of intelligence to the computational systems for context adaptation.

In [6], the authors propose a context adaptive platform based on the closed loop control principle. The novelty element of this proposal consists of defining and using the concept of application-context description to represent system knowledge about the context. This description is frequently updated and used for the system control allowing the system to reconfigure and take adapting decisions.

[7], [8] and [9] propose a context adaptation model based on defining system behavior in a certain situation using a set of context adapting rules. A rule consists of a context condition and an associated action. The main disadvantage of these approaches is given by the fact that new rules can not be learn or inferred at run time.

To conclude, we can state that none of the research approaches provides a unitary and complete solution for the development of pervasive autonomic systems. In this paper we try to overcome this deficiency by: (i) defining a basic context model that can be used to accurately represent general purpose real contexts, (ii) enhancing the basic model with new concepts that allow specifying the self-* autonomic properties, (iii) defining the self-configuring and self-healing autonomic properties for the context model.

The rest of the paper is organized as follows: in Section 2, the context model and its main elements are presented; in Section 3 we detail the context model management processes; Section 4 introduces the grounding concepts used to enhance the context model with autonomic features; Sections 5 and 6 show how the self-configuring and self-healing properties are added to the context model, while Section 7 concludes the paper and shows the future work.

2. The basic context model

Let's consider a pervasive system used to guide the tourists into a museum. The museum is an intelligent space where the visitors are identified by RFID readers, while their location and orientation is determined using a network of sensors. The tourists can interact with the pervasive system if they have a wireless capable PDA on which an application can be downloaded and executed. In the museum, the visitors must follow a set of rules such as the minimum distance to the artifacts, the loud limits, etc.

By studying and analyzing similar real world relevant scenarios we define the **basic context model** as a triple:

$$C =< R, A, P > \qquad (1)$$

where: R is a set of context resources, A is a set of actors which interact with context resources and P is a set of real context related policies. In our model the **context abstraction** is represented by the set of all context properties in terms of the relevant information provided by context resources.

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** that is defined as follows:

$$C_S =< R_S, A_S, P_S > \qquad (2)$$

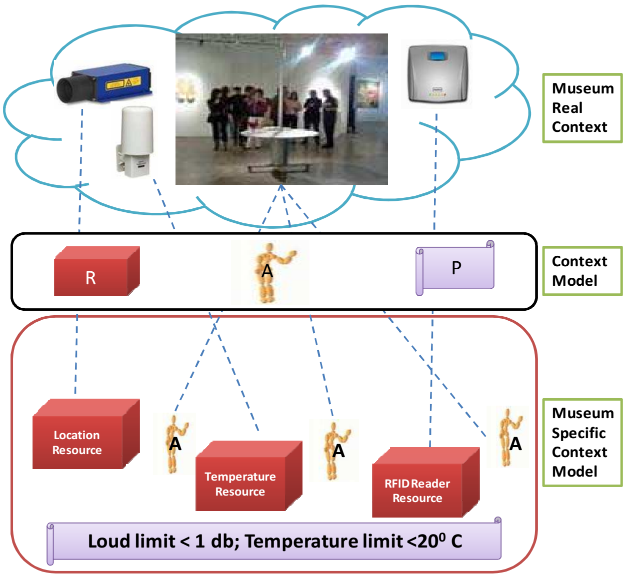

Figure 1. Context model elements

Using the above presented scenario, we have identified the following context specific elements (see Fig. 1) that populate the context model: (i) the context resource set R_S containing the tourist attached resources such as PDA or RFID tags and the intelligent museum resources such as location sensors; (ii) the set of context actors A_S containing the tourists and the executable context aware applications; (iii) the real context related policies P_S containing the constraints used to drive the tourist - museum interaction such as the minimum distance to the artifacts or loud limits.

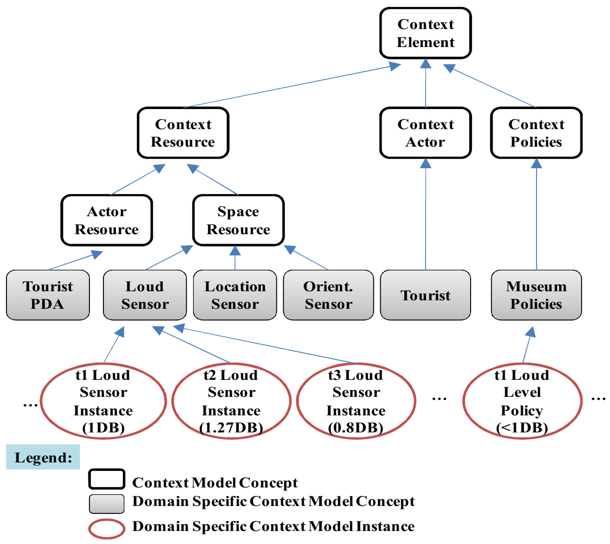

Figure 2. The context model ontology

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

A **context situation** to which a pervasive system must adapt is represented by a **specific context model instance**. An instance contains the set of resources with which an actor can interact, together with their values at a specific moment of time *t* and it is defined as a specific context model projection onto a certain actor:

$$CI_a^t = < R_a^t, P_a^t > \quad (3)$$

An actor interacts with the real context only through the specific context model. An actor or a resource is part of the context if and only if it is part of the specific context model.

The following sections detail the context model main concepts.

2.1. Resources

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

The set of context resources R can be separated in two disjunctive subsets: (i) the set of context resources attached to the physical space / environment R_S in which actor-context interactions occur and (ii) the set of context resources attached to the actors R_A that provide information related to actor-context interactions:

$$R = R_A \cup R_S \quad (4)$$

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

Resource Properties, $K(r)$ **-** specifies the set of relevant context information provided by the resource. As an example, $K(PDA) = \{Bluetooh, Wireless\}$.

Using the resource properties we can express the **context abstraction** as

$$(C_S, K) = (C_S, K_A) \cup (C_S, K_S) \quad (5)$$

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, $S(r)$ - specifies the resource functionality as a set of services (for example a service that locates / updates an object). The resource services are exposed by publishing them in a public registry such as UDDI. The actors interact with a context resource through its attached services.

Resource Influence Zone, $Z(r)$ **-** represents the 3D physical space in which a resource captures / provides context information or in which the resource presence can be sensed (in other words, it becomes visible for an actor or for another resource).

The influence zone of a context resource attached to an actor is the zero volume space: $\forall r \in R_A \Rightarrow Z(r) = 0_V$.

The influence zone for a context resource that is attached to the physical space is a non-zero volume space: $\forall r \in RS \Rightarrow Z(r) \neq 0_V$.

A physical space resource is considered an immobile resource so the influence zone is specified by using the resource position in the real space and the resource range.

2.2. 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) its specific resources, (ii) its context related requirements and (iii) the actor-context contract.

Actor Resources, R_a - specifies the set of actor associated resources such as position elements, RFID tags and / or augmented reality elements.

Context Related Requirements, $\mathrm{Re}q(a)$ - specifies actor context related preferences.

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

2.3. Policies

A policy represents a set of rules that must be followed by the actors or resources located in the context influence zone. The process of evaluating the policy complying degree is calculated by considering the complying degree of all policy rules. If a rule is broken an exception is thrown determining the elimination from the context of the actor or the resource that committed the fault.

3. The context model management

Due to the context model complex management, administration and adaptation processes, we have chosen the BDI agents as processing units for their reactivity, collaboration and adaptability features and inference capabilities. Four types of generic agents are defined for our context management model: *Context Model Administering Agents, Context Interpreting Agents, Request Processing Agents* and *Execution and Monitoring Agents*.

Context Model Administering Agent (CMAA) is the specific context model manager. Its main goal is the synchronization of the specific context model with the real context. This agent is also responsible for identifying and negotiating processes related to actor joining context as well as for adding / removing resources to / from the context. During the actor joining context negotiation process: (i) the entering actor presents its requests and properties to the CMMA agent and (ii) the agent informs the actor about context services, properties and policies. By using negotiation strategies, the CMAA agent adapts the joining actor or resource to the context and generates an actor-context contract that will drive the actor / resource evolution within the specific context model.

Context Interpreting Agent (CIA), semantically evaluates the information of a context instance when an actor makes a context related request. Its goal is to represent the semantic value of the context instance into the semantic states space. The semantic states space is a semantic hyper-space whose dimensions equal the number of context resources. The CIA agent builds the semantic space by using reasoning / learning algorithms and the context ontology. The semantic value of a context instance is obtained as a unique hyper-point in the hyper-space by positioning all resource values on the semantic space axes.

Request Processing Agent (RPA), is used for processing the actor requests. 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.

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. If an error occurs in the plan execution phase, compensatory actions will be taken for restoring the specific context model initial state and implicitly to the real context.

4. Towards an autonomic context model

To enhance the basic context model with autonomic capabilities we have introduced three new concepts: *isotropic context space*, *context granule* and *context model entropy*. Each of these concepts is discussed below.

4.1. Isotropic context space

A context sub-space (part of the whole context space) is **isotropic** if and only if the set of sub-space attached resources R_S is invariant to the movements of all actors in the context sub-space. In other words, in an isotropic context sub-space, the R_S set is the same for all the actors that are physically located in sub-space influence zone. It should be noted that if $Card(R_S) = 1$, the context sub-space is isotropic. From now on, the context sub-space will be also considered and referred as a context space.

Given a non-isotropic context space, the variation degree of the space isotropy Δ_{IZ} is defined as the variation of the R_S set, while the actor moves in the context space.

4.2. Context granule

Usually, a context space is non-isotropic but it can be split in a set of disjunctive isotropic context space volumes. We define the **Context Granule (GC)** as the maximum volume of a context space where the space isotropy degree variation is zero: $\Delta_{IZ\,GC-GC} = \emptyset$.

In a given moment of time, an actor can be physically located in a single context granule. As a result, Δ_{IZ} is non-zero only when an actor moves between context granules.

When the actor moves between two context granules GC1 and GC2, the space isotropy degree variation is determined as:

$$\Delta_{IZ\,GC1-GC2} = \{R_{GC1} \setminus R_{GC2}\} \cup \{R_{GC2} \setminus R_{GC1}\} \quad (6)$$

If $\Delta_{IZ\,GC1-GC2} = \emptyset \Rightarrow R_{GC1} = R_{GC2}$, the actor remains in the same context granule.

4.3. Specific context model entropy

$E(C_S)$ is defined as the specific context model entropy (the level of disorder) reflecting the degree of fulfilling the context policies ($E(C_S) = 0$, all context policies are respected).

If Rf, is a function over the policy rules that evaluates whether a certain rule is broken or not and Pf measures the policy fulfilling degree then the entropy is defined as:

$$E : C_S \rightarrow \mathrm{Z}$$

$$E(C_S) = \sum_{i>0} \sum_{j>0} Pf(Rf(rule_{ij})) \quad (7)$$

The entropy $E(C_S)$ is used to globally determine the autonomic capabilities of the specific context model:

$$E(C_S) = 0 \Rightarrow C_S \text{ is in an autonomic state}$$

$$E(C_S) > 0 \Rightarrow C_S \text{ is in a non-autonomic state}$$

A specific context model features autonomic behavior if the autonomy invariant (8) is always true.

$$E^t(C_S) * E^{t+1}(C_S) = 0 \quad (8)$$

5. Adding self-configuring capabilities to the context model

Using the concepts defined in the Section 4 we formally describe the self-* autonomic paradigms as an enhancement of the context model presented in Section 3.

The property of self-configuring or context adaptation is obtained by detecting and configuring those context resources / actors that determine real context variations. When a variation is detected, the CMAA agent performs a negotiation stage that has as the outcome the configuration of a new resource / actor according to the context policies. The context model self-configuring process always ends by creating a new specific context model adapted to the new real context.

The self-configuring property of the context model is enabled only if the autonomy invariant (8) holds for $C_S =< R, A, P_{Self-Configuring} >$.

The problem that arises is to evaluate the real context variation Δ_C and determine when the self-configuring process must be started.

From (5) we have seen that the context abstraction is given by the set of all context properties in terms of the relevant information provided by its resources. The context variation (9) is obtained by adding the variation generated by the physical space and variation generated by actors:

$$\Delta_C = \Delta_{CS} \cup \Delta_{CA} \quad (9)$$

5.1. Evaluating the physical space variation

The physical space context variation is generated by the space isotropy degree variation and by attaching / detaching a space context resource. The space isotropy degree variation is generated by the actor mobility as a result of its migration from a context granule to another context granule. On the other hand, attaching / detaching a space resource generates a real context variation and therefore, specific context model adaptation is necessary. In this case, the specific context model self-configures itself by adding / eliminating the resource. The physical space variation is calculated as follows:

$$\Delta_{CS} = \Delta_{IZ} \cup \{R_S{}^{t+1} \setminus R_S{}^{t}\} \quad (10)$$

The space isotropy variation degree Δ_{IZ} is zero only if the context actors are moving inside a context granule.

The specific context model self-configuring process starts only when $Card(R_S{}^{t+1} \setminus R_S{}^{t}) \geq 1$ which means that a new resource has been added / removed from the context. When $Card(\Delta_{IZ}) \geq 1$, a new context specific instance must be generated.

5.2. Evaluating the context variation generated by actors

The context variation generated by an actor is given by the context resources attached to the actor (i.e. the resources used in the actor-context interaction process). In a given context, an actor is characterized by a large number of actor-context interaction patterns. Only two of these patterns determine the modification of the specific context model: (i) the actor enters the context and (ii) the actor leaves the context.

If $A^t = \{a_k \mid k > 0\}$ is the set of all actors physically located in the context influence zone at a given moment *t* and R_A is the set of context resources attached to the context actors then:

$$\Delta_{CA} = \{A^{t+1} \setminus A^t\} \cup \{R_A{}^{t+1} \setminus R_A{}^{t}\} \quad (11)$$

The specific context model self-configuring process starts only when $Card(A^{t+1} \setminus A^t) \geq 1$ and has as a result the specific context model modification by adding / removing an actor and its resources form the specific context model.

6. Adding self-healing capabilities to the context model

The context model has the self-healing property enabled if only if the autonomy invariant (8) holds for $C_S =< R, A, P_{Self-Healing} >$.

In our context model, the self-healing property is enforced by the Model Administering Agent (CMAA), the Context Interpreting Agent (CIA) and the Execution and Monitoring Agent (EMA).

The **CMAA** agent continuously monitors the real context for detecting the broken context policies and executing compensating actions. It also reconfigures the resource or the actor that breaks the identified context policy. If the reconfiguration stage fails, the resource / actor is removed from the context.

The **EMA** agent implements the self-healing property by monitoring the action plan execution and taking compensation actions in order to keep the context in a consistent state when the actions plan execution fails.

The **CIA** agent implements self-healing properties in order to achieve (i) the semantic space construction and (ii) the evaluation of the context instances for determining and placing the associated semantically value in the semantic space. The semantic space and the semantic zones are constructed by the CIA agent using context policies, context ontology and reasoning / learning algorithms. The semantic values attached to the context instances that will determine the execution of the same actions form groups in the semantic space (see Fig. 3 for a semantic space example).

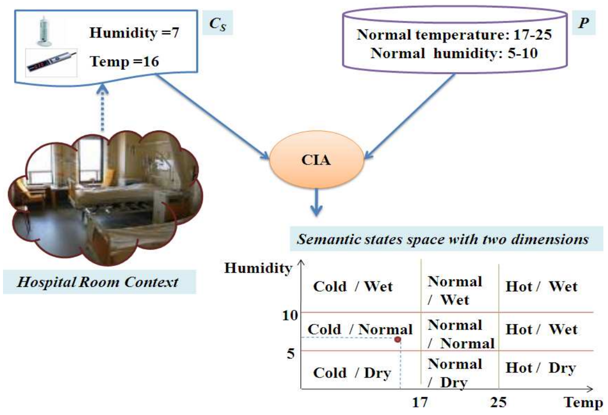

Figure 3. Representing a semantic space for a two resource context

The self-healing property is obtained by periodically evaluating the groups of the semantic space, targeting to avoid the forming of isolated semantically values.

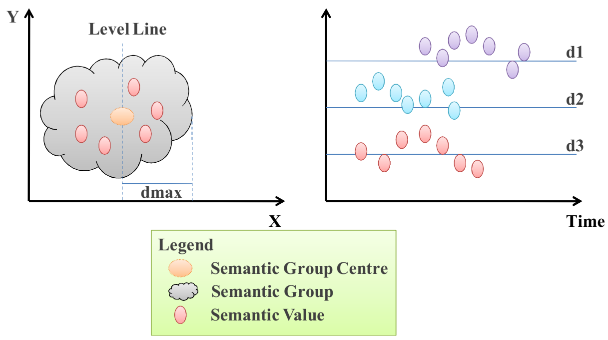

Figure 4. The semantic values polarization in time

In a semantic space, the time footmark of the semantic values belonging to a group follows a polarization line (level line) that is parallel with the time axis (Fig. 4). The self-healing property is implemented by permanently checking the semantically values polarization degree and identify the values that don't follow a level line. For these values the context state must be verified by the EMA agent for finding inconsistencies and for executing compensating actions, even if the plan execution does not fail.

7. Conclusion

This paper proposes the use of a triple set consisting of context resources, actors and policies to represent real contexts in a computer interpretable way. The model is mapped onto real contexts by populating the sets with context specific elements. To ease the context reasoning and adaptation processes a core ontology is defined to represent the relationships between the context model concepts. Specialized BDI agents are defined to deal with the complex context model management processes.

Using new concepts like isotropic context space, the context granule and the context model entropy, the self-configuring (context adaptation) and self-healing paradigms are defined for pervasive systems.

For future development, we intend to extend the context model with new concepts that will facilitate the definition and integration of new autonomic computing features.

8. References

[1] Anca Rarau, K. Pusztai. I.Salomie, "MultiFacet Item based Context-Aware Applications", *International Journal of Computing and Information Sciences*, 2006 pp. 10-18

[2] Irene Y.L. Chen, Stephen J.H. Yang, Jia Zhang, "Ubiquitous Provision of Context Aware Web Services", *Proc. of the IEEE International Conference on Services Computing*, 2006

[3] Ki-Chul Lee, Jung-Hoon Kim, Jee-Hyong Lee, "Implementation of Ontology Based Context-Awareness Framework for Ubiquitous Environment", *Proc. of the Int. Conference on Multimedia and Ubiquitous Engineering*, 2007

[4] Feruzan Ay, "Context Modeling and Reasoning using Ontologies", *University of Technology Berlin*, 2007

[5] Neil O'Connor, Raymond Cunningham, Vinny Cahill, "Self-Adapting Context Definition", *Proc. of the First International Conference on Self-Adaptive and Self-Organizing Systems*, 2007

[6] Marcel Cremene, Michel Riveill, "Autonomic Adaptation based on Service-Context Adequacy Determination", *Electronic Notes in Theoretical Computer Science, Elsevier*, 2007

[7] Brecht Desmet, Jorge Vallejos, "Layered design approach for context-aware systems", *Proc. Of the 1st Int Workshop on Variability Modelling of Software-Intensive Syst.*, Ireland, 2007

[8] Luiz Silva Santos, Fano Ramparany, Patricia Dockhorn, "A Service Architecture for Context Awareness and Reaction Provisioning", *Proc of the IEEE Congress on Services*, 2007

[9] Frederick Seyler, Chantal Taconet, Guy Bernard, "Context Aware Orchestration Meta-Model", *Proc. of the 3rd Int Conference on Autonomic and Autonomous Systems*, 2007