# Towards a Generic Contextual Elements Model to Support Context Management

Vaninha Vieira[1, 2], Patrick Brézillon[2], Ana Carolina Salgado[1] and Patrícia Tedesco[1]

[1] Center of Informatics, Federal University of Pernambuco, PO Box 7851, Recife, PE, Brasil
[vvs,acs,pcart]@cin.ufpe.br
[2] LIP6, University of Paris 6, 104 Avenue du Président Kennedy, 75016 Paris, France
[vieira,brezil]@poleia.lip6.fr

**Abstract.** Context management solutions propose the separation of context manipulation tasks from the application's business features. An important concern when managing contextual elements is how to structure and represent them. Context models in literature, generally, propose the formalization of specific contextual elements related to a domain and/or application. However, to be effective and reused by different systems, a context manager should abstract away domain or application particularities. In this paper, we propose COM (*Context-Oriented Model*), an approach to deal with the context modelling problem through the separation of generic context concepts from domain/application particular concepts. COM is a top-down modelling approach that can be used as the base context model to generic context managers. We believe that this approach can also serve to support developers of context sensitive systems in modelling the context usage into their applications.

## 1 Introduction

Context is what underlies the characterization of entities enabling to understand and differentiate situations. The term *Contextual Element* (CE) refers to pieces of data, information or knowledge that can be used to define the context [1]. Context management aims at providing solutions, such as models and services, to assist the acquisition, representation, processing, maintenance and manipulation of CEs, separating context-related tasks from the applications' business [1]. Generic context managers are of interest since they enable reuse and reduce the complexity associated in building context-sensitive systems, which are more complex than traditional systems exactly because of the additional context management needs.

An open topic in context management is to identify approaches and techniques to support the specification and representation of the contextual elements. Generic models are of interest since many different systems may benefit from them. However, existing proposals of context models, generally, are quite simplistic since they pre-define specific CEs related to concepts such as Person, Time, Location, Device and Activity [2;3], which can be, usually, identified by acquisition interfaces like sensors. We believe that the development of a generic context model should not start by specifying *a priori* which

CEs should be considered in a static and strict manner, since context is very dependent on the domain and application.

In this light, this paper proposes COM (*Context-Oriented Model*), a modelling approach for CE representation, based on the need to specify a generic context model for our domain-independent and reusable context manager, the CEManTIKA (*CE Management Through Incremental Knowledge Acquisition*) [1;4]. It separates the context modelling task in three steps: (i) formalize in a high level layer the context related concepts that will orient the context management; (ii) identify, for a given domain, the contextual elements that should be considered; and (iii) instantiate the contextual elements values according to an application usage.

The idea is to define generic concepts in the same way, for example, that object-oriented systems or aspect-oriented systems do. Generic concepts define the way the system should be modeled and thus allow the development of a mechanism that understands the systems that were built using that paradigm. Our approach proposes to bring this idea to the development of context-sensitive systems. We think that to achieve the generality and reuse of context-related solutions it is necessary to change the way developers think about their systems by explicitly identifying the context related functionalities. This model is our first step in this direction.

The rest of the paper is organized as follows: Section 2 presents our definition of context and a discussion about context modelling and related works; Section 3 introduces a motivating scenario to illustrate the model instantiation; Section 4 details the COM model and the context-related concepts of the high level layer; Section 5 exemplifies its use through the instantiation of the scenario illustrated in Section 3; finally, Section 6 points out some final considerations and further work.

## 2 Background

### 2.1 Our Definition of Context

Our work is based on two classical definitions of context. The first states that context is any information that can be used to characterize the situation of an entity (e.g. person, place, object, application) [5]. The second indicates that context is always related to a focus and that at a given focus the context is the aggregation of three types of knowledge: Contextual Knowledge (CK), External Knowledge (EK) and Proceduralized Context (PC) [6]. Focus means an objective or a step in a task, a problem solving, or a decision making. The focus enables to separate knowledge that is relevant or that is not relevant to determine the context. The CK is the known relevant part, while the EK is the unknown or irrelevant part in the context. The PC is the knowledge effectively used in the focus to support the task at hand; it is composed of a subset of the CK that is assembled, organized and instantiated to address the focus, along with the rationale that was used to achieve the instantiated CK.

We use the term *Contextual Element* (CE) to refer to pieces of data, information or knowledge that can be used to define the context [1]. This separation is necessary

because contextual knowledge comprises what is in the users' minds and thus is too abstract. To treat context computationally it is important to make this distinction between contextual data (e.g. location coordinates, identification and temperature), contextual information (e.g. weather is hot, nearby person) and contextual knowledge (e.g. understanding about the weather behavior in different regions).

### 2.2 Context Modelling and Related Works

An important issue in context-sensitive systems is the identification of the CEs that must be considered and how they should be represented to ease the acquisition and reasoning. With the advance of context-aware computing, there is an increasing need for developing formal and generic context models to facilitate context representation, sharing and semantic interoperability of heterogeneous systems [7]. A generic context model should take into account aspects such as the distinction between the context model and the application domain and ways to maintain a sharable context model that enables the communication between different systems.

There are several attempts to define and use context for computational purposes and different approaches are being experimented, such as key-value pairs [8], markup schemas [9], object-oriented models [10], ontologies [11] and topic maps [12]. Ontologies appear as a promising approach since they enable knowledge sharing between human and software agents, easy knowledge reuse between systems, and can be easily used by inference engines for reasoning. Topic maps also seem interesting because they can be used to organize large sets of information building a structured semantic link network over existing resources [12]. This network allows easy and selective navigation to the requested information. An interesting characteristic of topic maps is that topics can have relationships (associations) with each other and topics can play different roles in different associations. In our work we are using a combination of ontologies and topic maps to represent the CEs to achieve better results in terms of CE definition and navigation.

We observed that existing modelling proposals are based on pre-defined and limited sets of contextual information, in general related to a specific domain, or that can be acquired through sensors. They establish a static and pre-defined set of entities and model the context directly as static attributes associated to these entities. We believe that it is impractical to imagine a generic and reusable context model if we start by establishing and limiting the set of elements that is going to be modeled. Context is a dynamic and complex concept and the CEs are strongly dependent on the domain and application. So, it is not reasonable to imagine that a single system analyst or even worse the analyst of a generic context manager is able to specify and define *a priori* all CEs that are relevant to an application that not even exists.

The major difference of our proposal is that we do not intend to establish a fixed structure for the CEs related to a specific problem/application. We propose a meta-model through what the CEs will be modeled during the context-sensitive system analysis and development. Another approach for context representation, more similar to ours, was proposed by Bucur et al. [13]. They combine the generality of ontologies with the complexity inspired by object oriented models, modelling two ontologies: a domain ontology and another that is the description of the context attributes managed by the

system. The difference between our proposal and theirs is that we based our model in a definition well known and accepted in the context literature, which gives more theoretical foundation to support our modeled concepts. Also, we consider in the model the dynamicity of the context manipulation through the focus changing.

## 3 Motivating Scenario

To motivate and simplify the explanation of the model concepts we will consider the following scenario, that we will have in mind to exemplify the model concepts and its instantiation in a Mission Planning Support System.

Patricia is a researcher at a University located in the city of Recife in Brazil. Recently, she had a paper accepted to the *Context-07*, a conference that will take place at Roskilde University, in the city of Roskilde, Denmark. Roskilde is very near to Copenhagen, which for travelling matters is a city of reference indicated by the event organizers. The event will be held in the period of 20 to 24 august 2007. After receiving the good news about the paper, Patricia has to prepare her mission, taking care of things like the transport and hotel reservation. Another important point is that she has a PhD student under her direction doing a stage in a laboratory in Paris. So, to Patricia it is important to accommodate all these requirements (go to the conference, attend to interesting workshops and meet her student at Paris) to choose the best options for travelling and accommodation. Since it is her first time going to Denmark and she knows absolutely nothing about Roskilde, it will be really interesting to Patricia to have a system that supports her while preparing her mission by suggesting itineraries, hotels and travelling companies that are adequate to her needs. The system developer must identify the involved CEs and the CEs that are relevant to the different phases (focus) of a mission preparation.

## 4 The Context-Oriented Model (COM)

The COM model is divided in three layers (Fig. 1): the *upper layer*, which characterizes the generic context management related concepts and can be qualified as a "meta-model" since it is used for creating individual models; the *middle layer* that defines the domain-specific context-related concepts in accordance with the upper layer; and the *lower layer*, which represents the instantiation of the domain concepts according to a specific application, incrementally acquired during the system usage.

The idea behind the separation into these three layers is that if we define the context specific concepts in a high level domain-independent layer, than these concepts can be managed in a generic manner, without worrying about the domain particularities. So, the mechanics of the context manager will be applied over the upper layer concepts. A compatible context-sensitive system must model the application concepts as instantiation of the upper layer concepts. Existing context and domain ontologies, for example, can be used in the middle layer.

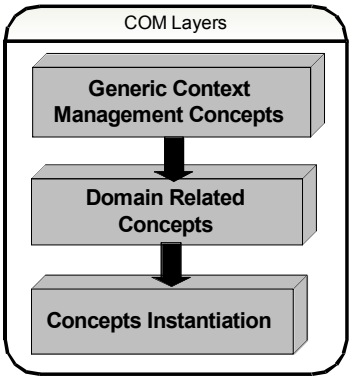


**Fig. 1.** Interaction between the three layers in the COM model

This paper focuses in discussing the upper layer concepts, which are detailed in the next subsections. The other layers will be discussed in Section 5 through the instantiation of the model according to the motivating scenario described in Section 3.

### 4.1 Concepts Specification

An illustration of the generic context management concepts and their properties and relationships are presented in Fig. 2. The model is centered on five main concepts: *Entity*, *Contextual Element*, *Focus*, *Rule* and *Action*; and three derived concepts: *CEF-Set*, *RF-Set* and *Proceduralized Context*. The main concepts must be specialized by the application/domain analyst while the derived concepts are built by the context manager based on the main concepts and guided by the focus.

#### 4.1.1 Entity

An entity is anything in the real world that is relevant to describe the domain. For example, in the scenario in Section 3, some entities are Person, Hotel, Mission, Location, TransportType, FlightCompany, TrainCompany and Event. The entities definition can be imported from existing sources such as a domain ontology. An Entity is defined through: a *name*; a *type* (e.g. an entity Missionary is of type Person, an entity Country is of type Location); and an *URI* that permit to link the entity to an external resource that contain further knowledge, such as its OWL description file. The Entity has an association *isCharacterizedBy* with the concept Contextual Element, meaning that one Entity is characterized by one or more CEs.

#### 4.1.2 Contextual Element (CE)

A Contextual Element, as defined before, may represent a contextual data, information or knowledge, and is used to characterize entities. Examples of CEs for the entity *Mission* include: *location*, *initialDate*, *endDate, duration, whoPays* and *officialReasons*; and for the entity *Hotel* include: *location*, *numberOfStars*, *category*, *minimunPrice* and *numberOfRoomsAvailable*.

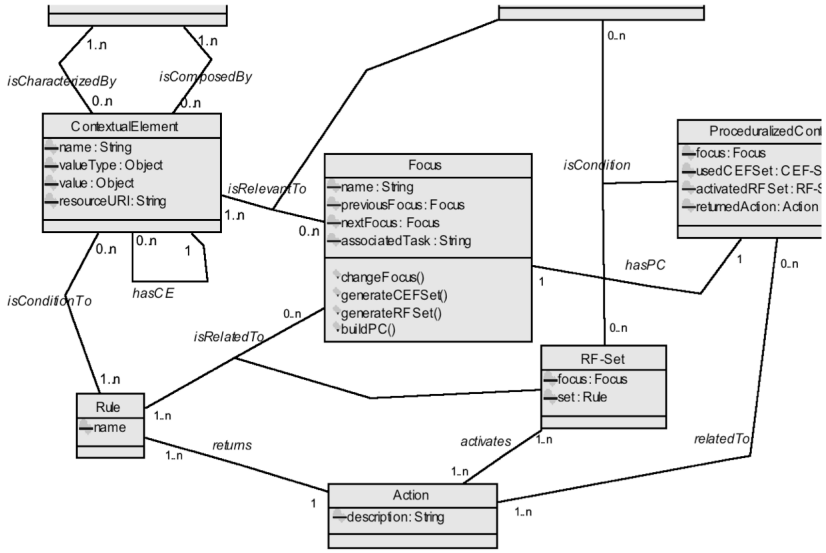


**Fig. 2.** Specification of the generic context management concepts using UML Notation

The attributes that identify a CE are *name*, *valueType*, *value* and *resourceURI*. The *valueType* indicates the expected value for the CE. For example, the CE *location* has as valueType an entity Location, the CE *numberOfStars* expects as value type an object of type Integer, while the valueType for the CE *category* is an entity Category, which contains the instances {economic, executive, first class}. The attribute *value* signifies the ascribed instance of the CE and will be filled by the context-sensitive application during its usage. The attribute *resourceURI* identifies a link to an external resource that describes the CE, such as an OWL or RDF file.

A CE may be associated to one or more CEs, creating a hierarchical structure between the contextual elements. For example, the CE *officialReasons* may have as possible values the set {*presentation*, *experimentation*, *meeting*}, meaning that the official reasons for a mission may be one of these. In its turn, *presentation* is itself a CE that has as value types the subset {s*eminar*, *conferencePaper*}. This relation between CEs reinforces the hierarchy and dependency.

The CE concept has two associations with the Entity concept. The first points out that an entity is characterized by one or more CEs, and the second implies that a CE may be composed of one or more entities. The latter is important in situations where a CE associated to an Entity X needs to make a reference to an Entity Y. For example, the CE *distance* associated to entity Hotel needs to make a reference to another entity, such as Mission, indicating the distance between a Hotel and a Mission location.

### 4.1.3 Rule and Action

The Rule concept was included in the model to explicitly represent the rules associated to the CEs, necessary to produce contextual information from contextual data and also to

support the building of the Proceduralized Context in the focus. The Rule is identified by a *name*, it has one or more conditions, which are represented by the association *isConditionTo* with the Contextual Element concept, and it has one returning action, represented by the association *returns* with the Action concept.

The Action is represented by a *description* and exists in function of a rule. We modeled the action separately from the rule to easy the modelling the context dependent actions that could be implemented by the systems. In this light, the rules are specified having in mind the possible and desired actions.

#### 4.1.4 Focus

The focus is a central and important concept in our model, since the context is always related to a focus. We define the focus as an objective to be achieved, such as a task in a problem solving or a step in a decision making. It is used to identify clear points of time and space that the context is all about. The focus allows the context manager to determine what CEs should be used and instantiated, since it determines the relevancy of a CE in a specific situation. In the scenario of Section 3, examples of focus are *Define the mission*, *Book Hotel*, *Book Transport* and *Make Payment*.

The focus is identified by a *name*. Since it is related to objectives to solve a problem or to execute a task, we modeled it as a sequence of foci, where each element has references to the *previousFocus* and the *nextFocus*. Another attribute is the *associatedTask* that may be a problem, a decision making or a task, and can contain a textual description or an URI referencing an external resource that describes the task.

The associations for the concept focus are: *isRelevantTo* with the concept Contextual Element, showing that the CE is relevant to that focus. Similarly the concept Rule has an association *isRelatedTo* showing that the rule is related to the focus. The *hasPC* indicates that a focus has a related Proceduralized Context.

Context is a dynamic construction that evolves with the focus. As the focus changes the set of CEs that must be considered change accordingly [6]. In this way the Focus concept is the one who guides the generation of the derived concepts (CEF-Set, RF-Set and Proceduralized Context). This is done by the methods associated to the Focus: *changeFocus*, shows that the focus changed pointing out that it is necessary to review the current context; *generateCEFSet*, which marks the building of the CEF-Set according to the current focus and its associated CEs; *generateRFSet*, to build the RF-Set according to the relevant Rules for the focus; and *buildPC*, which indicates the procedures to construct the PC for the focus.

#### 4.1.5 CEF-Set

To identify and manipulate the CEF-Set (*Contextual Elements in the Focus Set*) we use the principles of the mathematical theory of sets. A set is a collection of abstract objects. So, a CEF-Set is a collection of CEs that are relevant and must be considered in a Focus. It is dynamically generated and will be continually rebuild when a new focus arrives. So, as stated in the definition below (*CEF-Set (f)*) an CEF-Set related to a focus *f* is comprised by a set of tuples (*c, v*) such that *c* is a Contextual Element, *v* is a value for *c*, being another Contextual Element or a literal (e.g. string, integer or date), and *c* has an association of relevancy with the focus *f*.

```
CEF-Set (f) = {(c, v) | isContextualElement(c) AND
isValueOf(v, c) AND (isContextualElement(v) OR isLiteral(v))
AND isRelevant (c, f) }
```

For example, in the scenario of Section 3, the entity Hotel is characterized by CEs such as: *location*, *architecturalStyle*, *yearOfFoundation*, *executiveManagerName*, *numberOfStars* and others. In the focus *Book Hotel* the CEF-Set will contain only the CEs *location* and *numberOfStars*, since the other CEs are totally irrelevant for the problem of recommending hotels for Patricia's mission.

#### 4.1.6 RF-Set

The RF-Set (*Rules in the Focus Set*) has a similar functionality as the CEF-Set, but it determines the relevant *rules* that should be activated in the focus. The Rule concept models all rules in the system related to the defined CEs. However, in a focus only a subset of all rules must be, in fact, activated. Thus, the definition *RF-Set (f)*, below, indicates that an RF-Set in a focus *f* is formed by a set of tuples (*r, a*) such that *r* is a Rule, *a* is an Action, *a* is a result of the execution of the rule *r*, and *r* has an association is related to the focus *f*.

```
RF-Set (f) = {(r,a) | isRule(r) AND isAction(a) AND
isResultOf(a, r) AND isRelatedTo(r, f)}
```

#### 4.1.7 Proceduralized Context (PC)

The PC contains the CE that should effectively be used to support the current focus, and the rationale that has enabled the context manager to identify these CEs. It is constructed based on the contextual elements in the CEF-Set and the selected set of inference rules (the RF-Set). The PC is composed of an explanation related to the processing of the instantiation of the inference rules with the elements in the CEF-Set as conditions and the returned actions.

```
PC (f) = {[(c, v),(r, a)] | isInCEFSet(f,(c,v)) AND
isInRFSet(f,(r,a)) AND isConditionOf(c,r) AND return(r,a)}
```

The definition *PC (f),* above, states that the PC in a focus *f* is formed by a set of tuples [(*c, v*), (*r, a*)] such that (*c,v*) belongs to the CEF-Set, (*r*, *a*) exists in the RF-Set, *c* is a condition for *r*, and the execution of *r* with the tuple (*c*, *v*) return the action *a*.

#### 4.1.8 Final Considerations about the COM Model

This section presented the general context related concepts defined for COM. We based our model in the same idea existent when we think about an object oriented model or a relational based model. These modelling techniques define the main structure that enables the building of a computer system that fits each paradigm. If we are going to develop an object oriented system our world must be represented according to classes, properties attributes and objects (instances). When modelling a relational system, the

world should fit in the concepts of tables, rows and columns. So, for us, when thinking about modelling a context-based system we must think in terms of entities, contextual elements, focus, rules and actions. And the usage of context is done based on focus and proceduralized contexts.

An important concept in the set theory is the *Universal Set*, which means the set of all elements and subsets we are interested in. In our model it is what we mean by the domain being considered. Another concept of the set theory is the Complement of a Set, which contains all the remained elements in the Universal Set not considered in a set. The context model proposed by Brézillon and Pomerol [6] discussed in Section 2.1, in which this work is based on, includes the concept of External Knowledge (EK). In our model the EK in a focus may be achieved by applying the complement of a set over the CEF-Set. So, all elements that are a Contextual Element and that is not on the CEF-Set for a given focus is considered External Knowledge.

## 5 Example of Use – Instantiation of the Model

Fig. 3 shows an overview of the instantiation of the main concepts according to the scenario presented in Section 3, illustrating the interaction between the three layers presented in Fig. 1. In the example, three entities were defined: *Person*, *Misson* and *Hotel*. The entity Person has the CE *location*. The entity Mission has the CEs *location*, *duration* and *whoPays* (that uses the entity Person to indicate that the value for the whoPays for a mission will depend on who the missionary is). The entity Hotel has as CEs *location* and *distance* (who uses the entity Mission to indicate that the distance of the Hotel is relative to the location of the mission).

The foci *BookHotel* and *MakePayment* indicates different phases related to the mission preparation. The focus Book Hotel uses the CEs *location, duration* and *distance*. The CE *whoPays* is not relevant to BookHotel since the focus is related to finding hotels for a missionary taking into account elements such as: the distance between the hotel and the mission location and the mission duration (to verify the availability of rooms). The focus Make Payment indicates how the missionary should conduct the payment of the mission. For example, if *whoPays* for the mission is a travel agency that has a contract with the missionary department, then the payment should be done by a payment order emitted by the department secretary to the agency.

The main concepts are used to produce the derived concepts that will enable the context manager to build the PC related to a focus. The role of the derived concepts in the PC building is illustrated in Fig. 4 from the execution of selected rules (RF-Set) over relevant CEs (CEF-Set) according to the example shown in Fig. 3.

The CEF-Set in the focus BookHotel contains the instantiated CEs identified as relevant: *location*, *duration* and *distance*. The RF-Set contains the relevant rules for the focus: *EconomicHotelRule, IsClientRule, IsNearRule* and *HasRoomRule*. The execution of these rules may produce new CEs that will increase the original CEF-Set. For example, the rule *IsClientRule* instantiated a CE *isClient* for the entity-instance *Person:Vaninha* related to the entity-instance *Hotel:Ibsens*. For the sake of clarity we separate in the CEF-Set illustrated in Fig. 4 the previously known CE (upper part of the CEF-Set) from the new inferred CE (lower part).

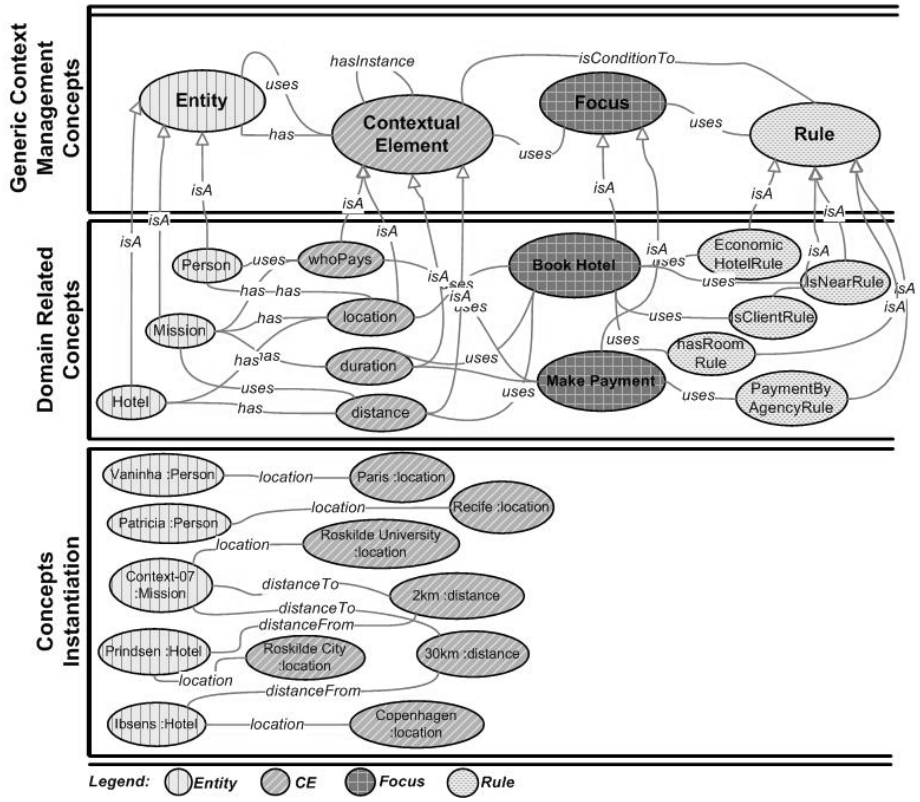


**Fig. 3.** Illustration of the COM Model layers interaction and the main concepts instantiation

The PC contains the set of the rules executed with the respective instantiated CEs and the final returned actions. For example, the entities *Prindsen* and *Ibsens* passed in the rule *EconomicHotelRule*, while only *Prindsen* passed for the *IsNearRule,* but the *IsClientRule* identified that the *Vaninha* is already a client of the *Ibsens*, and both hotels were returned by the *HasRoomRule* application. The recommended hotels list includes the *Prindsen* for the missionaries *Vaninha* and *Patricia*, however the *Ibsens* was also indicated to *Vaninha* since she was classified as a client of this hotel. With all this rationale contained in the PC, the context-sensitive system may, for example, show the list of recommended hotels for the two missionaries, with the explanation of how it arrived to that list. Each missionary can then apply their own criteria and select the hotel they finally want to book.

This is a small example of how the COM model works, that illustrates how it can be instantiated for a specific domain and application, shows the relation between the different layers and discusses the generation of derived concepts from main concepts. The objective of our work is to implement the CEManTIKA context manager in ways to enable an easy instantiation of the main concepts by domain/application specialists, and to implement the mechanics that permit the generation of the derived concepts.

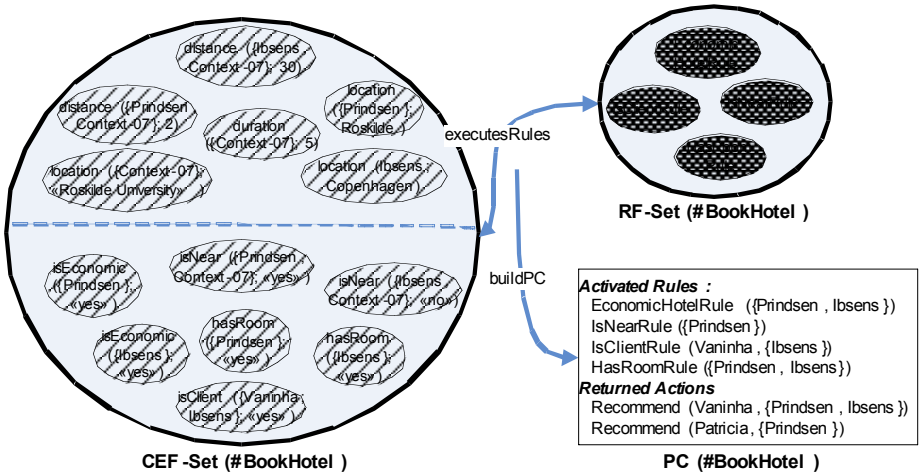


**Fig. 4.** Derived concepts of COM model used to build the PC in the focus BookHotel.

## 6 Conclusions and Further Work

This paper presented COM (*Context-Oriented Model*) an approach for context representation that proposes the separation of the context management concepts from domain and application concepts. We assumed the hypothesis that it is impossible to imagine a context model that is at the same time specific and generic, since context is extremely domain and application-dependent. Thus, we propose that developers of context-sensitive systems rethink the way of modelling their systems including the context management phases in the system building process and considering the context related concepts when specifying the system functionalities.

Currently, we are working on the implementation of the proposed model using ontologies and topic maps. For us, the approach of topic maps seems ideal since all concepts (in the upper, middle or lower level) can be represented as topics and freely linked with one another through associations. This enables a richer approach and ease the knowledge incremental acquisition providing a flexibility in the contextual elements representation without a rigid hierarchical format. Ontologies enable the formal specification of the concepts, such as entities and contextual elements, and ease the reuse of existing solutions. We believe that this work is a first step in a new approach to model and develop context-sensitive systems as well as the integration between these systems and context managers.

**Acknowledgments.** Authors want to thank CNPq and CAPES for their financial support, and the first author thanks UFBA for its support.

## References


1. Vieira, V., Tedesco, P., Salgado, A. C., Brézillon, P. "Investigating the Specifics of Contextual Elements Management: The CEManTIKA Approach". In: CONTEXT 2007, LNAI 4635, Roskilde, Denmark (2007), pp. 493-506.
2. Bulcão Neto, R. F., Pimentel, M. G. C. "Toward a Domain-Independent Semantic Model for Context-Aware Computing". In: Proceedings of the 3rdIW3C2 Latin American Web Congress, IEEE Computer Society, Buenos Aires, Argentina, (2005), pp. 61-70.
3. Belotti, R., Decurtins, C., Grossniklaus, M., Norrie, M. C., Palinginis, A. "Modelling Context for Information Environments". In: Workshop on Ubiquitous Mobile Information and Collaboration Systems (UMICS), CAiSE 2004, Riga, Latvia (2004).
4. Vieira, V. "The CEManTIKA Project Homepage" (2007) In: http://www.cin.ufpe.br/~vvs/cemantika/, Accessed in 03/2007.
5. Dey, A. K., Abowd, G. D. "Towards a Better Understanding of Context and Context-Awareness". In: Proceedings of the CHI 2000 Workshop on The What, Who, Where, When, and How of Context-Awareness, The Hague, Netherlands (2000).
6. Brézillon, P., Pomerol, J.-C. "Contextual Knowledge Sharing and Cooperation in Intelligent Assistant Systems", Le Travail Humain, PUF, Paris, v. 62, n. 3 (1999), pp. 223-246.
7. Gu, T., Pung, H. K., Zhang, D. Q. "A Service-Oriented Middleware for Building Context-Aware Services", Elsevier Journal of Network and Computer Applications (JNCA), v. 28, n. 1 (2005), pp. 1-18.
8. Zimmermann, A., Lorenz, A., Specht, M. "Applications of a Context-Management System". In: Proceedings of the CONTEXT-2005, Paris, France (2005), pp. 556-569.
9. Indulska, J., Robinsona, R., Rakotonirainy, A., Henricksen, K. "Experiences in Using CC/PP in Context-Aware Systems". In: Proceedings of the 4th International Conference on Mobile Data Management (MDM2003), v. LNCS 2574, Melbourne/Australia (2003), pp. 247-261.
10. Henricksen, K., Indulska, J. "Developing Context-Aware Pervasive Computing Applications: Models and Approach", Pervasive and Mobile Computing Journal (2005).
11. Vieira, V., Tedesco, P., Salgado, A. C. "Towards an Ontology for Context Representation in Groupware". In: Proceedings of the 11th International Workshop on Groupware (CRIWG'05), Porto de Galinhas, Brasil (2005), pp. 367-375.
12. Power, R. "Topic Maps for Context Management". In: Proceedings of the 1st International Symposium on Information and Communication Technologies - Workshop on Adaptive Systems for Ubiquitous Computing, Dublin, Ireland (2003), pp. 199-204.
13. Bucur, O., Beaune, P., Boissier, O. "Representing Context in an Agent Architecture for Context-Based Decision Making". In: Proc. of the International Workshop In Context Representation and Reasoning, CRR-05, Paris, France (2005).