The Conceptual Model of Context for Mobile Commerce Applications

Benou Poulcheria[1], Vassilakis Costas[1],

[1]University of Peloponnese,

Terma Karaiskaki 22100 Tripoli, Greece

{pbenou@ethnodata.gr, costas@uop.gr}

Abstract. Mobile commerce applications adhering to anytime and anywhere paradigm, required to be flexible. They should be able to adapt their interface, services and content towards a certain context. Several proposals for definition of context have been already proposed originating from various areas related to mobile commerce. However, an integrated, formal and methodological approach for the determination and representation of context, adjusted to special characteristics of mobile commerce applications, has not been insofar presented. This is the challenge we address in this paper, through a conceptual model that includes: i) a clear and formal definition of context, ii) the depiction of its specific characteristics as metadata, iii) a methodology for its determination and iv) the presentation of an extension of class diagrams of UML for its representation, all of them tailored to the special nature of mobile commerce applications.

Keywords: mobile commerce, context, context-awareness, adaptation, UML

1 Introduction

The increased popularity of new mobile and embedded computing devices and the advent of wireless communication technologies have broadened the application areas of computing. Computing is expanding from stationary desktop PC environments to locations like cars, shopping malls, entertainment areas and even corridors. Small devices, such as Personal Digital Assistants (PDAs) and mobile/smart phones of varying capabilities allow the anywhere and anytime access to information and conduct of electronic transactions. The new technological environment enables e-commerce to widen the spectrum of its applications and users to a new form of commerce known as *mobile electronic commerce*. According to [3], "*Mobile commerce* or *m-commerce* is defined as any activity related to a commercial transaction (or a potential one) –a transaction that includes a monetary value- and is conducted via wireless and mobile communication networks and uses wireless and mobile devices as user interface.

The process of designing and developing mobile commerce applications is inherently more complex and demanding, as compared to traditional applications, due to the fact that they are executed in diverse environments, as opposed to "traditional applications" which are typically executed on the relatively stable desktop PC. Within these environments, there are greatly varying characteristics regarding (a) the properties of the individual devices (memory capacity, battery lifetime, processing power, input/output and communication capabilities), (b) the properties of the networking infrastructure (latency, bandwidth, disconnections, cost), and (c) the properties of the natural surroundings (noise level, brightness, temperature). These aspects should be taken into account while designing the m-commerce applications, since – for instance- image download and display may be omitted to save battery and/or communication costs; smaller size images can be used if the available communication bandwidth is limited or the screen size is small (to avoid scrolling); limited input capabilities also dictate the need for less typing through the keyboard. Moreover, user mobility leads to the need for extending the use of these applications both temporally and spatially, while at the same time users may interact with mobile commerce applications while concurrently engaging in other activities (e.g. driving). Hence, the full attention of the user cannot be assumed and alternative communication modes may need to be explored (e.g. auditory instead of visual) [21][22].

M-commerce applications are furthermore addressed to an audience with greatly varying qualities regarding their personal characteristics, preferences, computer literacy and skills, needs and desires. Lastly, the merchandise (tangible or intangible) traded within an m-commerce transaction is of focal interest, since the added value of an m-commerce transaction lies in the ability to promote and trade the merchandise within the "anytime/anywhere" framework. These particularities, known under the general term *context,* demand from an m-commerce application adaptability, in terms of *user interface*, *functionality* and *content*, so as to maximize user satisfaction; these demands naturally reflect on the design and development methodology of m-commerce applications.

To facilitate the processes of m-commerce applications design and development, a solid perception of concepts and structures related to context capturing and management is required. This perception should be common to all involved user groups (customers, system analysts, service providers etc), and must also be free of vagueness. In this paper, we present a conceptual model which focuses on providing clear and precise modelling for context-related issues, in an implementation detail-independent fashion. The requirements for this model are derived from studying individual context models as well as comparative analyses of context models (e.g. [14]), as well as from the study and evaluation of both real and potential usage scenarios [12] and include:

(i) clear and formal semantics for context information

(ii) documentation of the context information's special characteristics

(iii) an effective methodology for determining the context information which is of value to each particular case and

(iv) well-founded tool for its representation, based on the broadly accepted class diagram notation.

The remainder of this paper is organized as follows: Section 2 presents a brief overview of context generation and exploitation approaches. Section 3 formally defines the concept of context information and discusses its special characteristics. Section 4 proposes a methodology for the determination of context information and an extension of UML diagrams for its representation. Section 4 also includes a scenario analysis exemplifying and clarifying the steps of the methodology, showing the relation of each step to the formal definition of context. Section 5 discusses related work and makes a comparison with our proposed conceptual model, while section 6 concludes the paper.

2 Context Generation and Exploitation

Many of the parameters that comprise the operational environment of mobile commerce applications, including context, have been studied in recent research, mostly in the areas of *pervasive and ubiquitous computing,* as well as in the areas of *mobile computing* and *personalization*.

Regarding the area of *pervasive and ubiquitous computing*, the vision of Mark Weiser for transparent interweaving of technology and user environment [15] has led to studying the parameters that comprise the operational environment (i.e. *the context*) of an application, aiming to enhance it with intelligence to the benefit of the user. Schilit [13] was the first to introduce the concept of *context* as “location, identities of the people around the user, time of day, season, temperature, etc”. Later, more researchers followed Schilit’s approach and defined context through a series of enumerations [4][11]. Dey [6] adopted a more global view and presented a generic definition according to which, ’*context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves’*. He also defined an application to be *context aware*, *’if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user’s task’*.

The approach of defining context through a series of enumerations suffers from the drawback that the context is modelled too rigidly and specifically and thus parameters not listed in enumerations may be disregarded. Moreover, in the cases that enumerations suggest implementation details, these may be inappropriate for the specific situation. The alternative approach of defining context through a generic definition that can potentially accommodate any information as a context element, indeed provides the freedom to choose the context elements suitable for the specific situation. This definition needs, however, to be complemented with means allowing the identification and modelling of *all* relevant context parameters, otherwise context specification may be incomplete and the application’s potential for adaptation will be diminished. Indeed, most current context-aware systems employ a small portion of the available context, comprising typically of elements related to the user’s identity and location [14].

In the area of *mobile computing,* efforts mainly target the development of adaptive systems having *multichannel delivery* and *network adaptation* as their main adaptation focus; the context was explored in alignment to this focus. Multichannel delivery attempts to provide services, through web applications, that can be readily used on various diverse mobile devices with different hardware characteristics. *Network adaptation*, on the other hand, mainly addresses two issues: the first is *communication autonomy*, i.e. the adaptation of the application to sudden disconnections, either initiated by the user (in an attempt to avoid disturbance, reduce cost or power consumption, etc) or imposed by the infrastructure (inadequate battery, signal loss, etc). The second issue targeted by network adaptation is the adjustment of the services to changes in the network bandwidth (e.g. switch from WiFi to GPRS); bandwidth changes and limitations often dictate the need to change the transmitted data content (lossy vs. lossless compression mechanisms, change of multimedia resolution or quality) or alter the underlying protocol [2].

Finally, the aspect of *personalization* has been investigated in various areas of application development, including that of adaptive user interfaces [7], which pursue the tailoring of the behaviour of a system's interaction to match the skills, tasks and preferences of its users. Personalization also examines issues of content adaptation, through research on *information filtering and recommender systems* [1]; systems employing these technologies aim to present to the user the information considered most appropriate for his/her current information needs. In the area of personalization, the meaning of context revolves around the user and his preferences.

All the aforementioned research areas address –from different standpoints- issues related to context for mobile commerce applications, which are under our consideration, in order to be able to realize the adaptation of their interface, content and/or functionality. However, a complete methodological approach for identifying and representing context, specifically tailored for the nature of mobile commerce applications, has not been suggested insofar. The need for a methodology specifically targeting the context for m-commerce applications, stems from a number of reasons including:

1. m-commerce applications aim –by their very nature- to the realization of sales. Therefore, factors of the context specific to the domain of sales [36], [37], should be taken into account in the methodology. Such a factor is the introduction of innovative services to increase market share, using context information to seize sales opportunities [38] (e.g. in the event of sudden, heavy rain, suggest nearby stores for purchasing an umbrella). Another factor is, for instance, the time limitations that may be present for completing a transaction (e.g. a ticket purchase should be completed before the customer arrives at the train station or local offers information should be timely delivered to a driver before s/he moves away from the offering stores). Maintaining loyalty to the application and/or the merchandise are also factors that are mostly addressed in the context of m-commerce.
2. m-commerce applications target a very wide range of end-users and access devices, with no opportunity whatsoever to train the users (as can be the case with, for instance, a context-aware computer-assisted learning environment that is used in a school [39]) or exclude access devices[1]. Failing to provide a suitable interface, levelled to each user's needs [36], or excluding a device, will result to reducing the number of sales that will be realized, therefore the application's sole goal will not be achieved.
3. The m-commerce application domain is a highly competitive one, while other application domains that involve context may be far less competitive. For instance, a context-aware computer-assisted learning environment that is used in a school is considered as "given" and cannot be changed, while in the m-commerce environment a lot of alternatives exist and switching to a competitive application is very easy (typing a different URL or using a different bookmark).

[1] Device exclusion is seldom explicit, but performing poor adaptation, by e.g. using very large images in devices with small bandwidth and limited display sizes will lead the users to navigate away from the application, possibly switching to another competitive application that is available.

4. Finally, cost effectiveness is a major concern while designing an m-commerce application. Information regarding the target group of the services, the cost of collecting and exploiting the context, and the added value that will be provided by the introduction of certain context factors has to be reviewed, by both technology experts –to determine feasibility- and business analysts alike –to assess cost effectiveness.

In the following two sections, we will present a formal definition of the context for mobile commerce applications and a methodology to elicit and represent the context parameters required for a mobile commerce application. The proposed methodology can be employed in the phases of mobile commerce application requirements analysis, design and implementation.

3 Definition of Context for Mobile Commerce Applications

3.1 The Concept of Context Information

After analyzing the concepts of context in the domain of m-commerce applications, we define *context* as *the set of all possible conditions and states that surround an electronic commerce operation*, whereas we define *context information* as *the set of data elements comprising the operation context*. Context is therefore an abstract model, which -through a series of design and implementation activities- will be mapped into concrete *context information* elements; the latter will be finally utilized to support the adaptive services.

In the stages of m-commerce application analysis, design and development, we will mainly address *context information*; for the requirements of these stages, context information may be extended as follows:

"Context information of an m-commerce application is every piece of information which may be used to characterize a state of an entity, which can be considered to be relevant to the interaction of the user with the particular application. The entity state may be either static or dynamically changing, while the relevance of the entity to the user-application interaction can be derived from the potential to exploit the information describing the entity state to optimize this interaction, so as to maximize the commercial value of the application."

With the term "entity", we refer both to the term "entity" and the term "relationship" of the Entity Relationship Model (ER-Model). For clarity purposes, we will call the "entities" of the ER-Model as "base entities" and the "relationships" of the ER-Model as "associative entities" [17].

The definition of context, as given above, attempts to limit the scope of Dey's definition, according to which *context is any information that can be used to characterize the situation of an entity*, by confining this information to the portion *that can be considered to be relevant to the interaction of the user with the particular application*. We consider this to be important, since it allows application analysts and designers to reduce the scope of context coverage only to the part that they intend or can exploit, which leads to a more manageable analysis and design phase. Moreover, the definition above tries to capture two different aspects of context: the aspect of context *as humans understand it* ("entities can be considered to be relevant to the interaction of the user with the particular application"), and the aspect of context *as understood in the software engineering domain* ("every piece of information which may be used to characterize a state of an entity"). The methodology presented in section 4 includes steps for capturing the former aspect of the context (*as humans understand it*) and transforming it to the latter (context *as understood in the software engineering* *domain*), which can be subsequently exploited in the m-commerce application design and implementation.

In order to facilitate the identification and classification of entities whose state should be considered for an m-commerce application context, we introduce a high-level grouping of the base entities corresponding to the various context domains that should be considered. Thus, we define as *context domains* the following: *user domain, computing domain, environment domain* and *application-specific domain* (see figure 1). Context domains are also called *entity groups*, because they group together entities pertaining to the same actor. The *user domain* includes information relevant to the user. The *computing domain* includes information regarding the computing (and communication) infrastructure. The *environment domain* encompasses information regarding the real-world aspects of the user and computing surroundings, such as location, time, weather etc. The *application-specific domain* contains information that is conceptually related to the particular application. We can observe that the first three groups are common to all m-commerce applications' categories, whereas the fourth (*application-specific* domain) is specialized for different application classes or even at application level.

The base entities of these four context domains may be interrelated; relationships may be established either among base entities within the same context domain, or among base entities belonging to different domains; these relationships, as we have already mentioned, are called "associative entities'. Relationships between entities of different domains - namely the associative entities -, are depicted in figure 1 as lines labeled "Relation u-e", "Relation u-c", "Relation e-c", "Relation a-c", "Relation a-u", "Relation a-e". Associative entities, derived by relationships among entities of the same domain, are naturally classified within the domain to which both associated entities belong to. For the classification of associative entities derived by interrelating entities of different domains, the analyst should consider the semantics of the relationship -if some of the interrelated entities are deemed more important, or even properties of the development philosophy (e.g. user-centric model vs. process-based models). For example, we may consider the associative entity “User access devices” (a relationship between the base entity “user” of the user domain and the base entity “device” of the computing domain), which is more naturally classified into the user domain, since the properties of this relationship are more user-oriented. The arrows’ directions in figure 1 show the domain in which the produced associative entities could be classified.

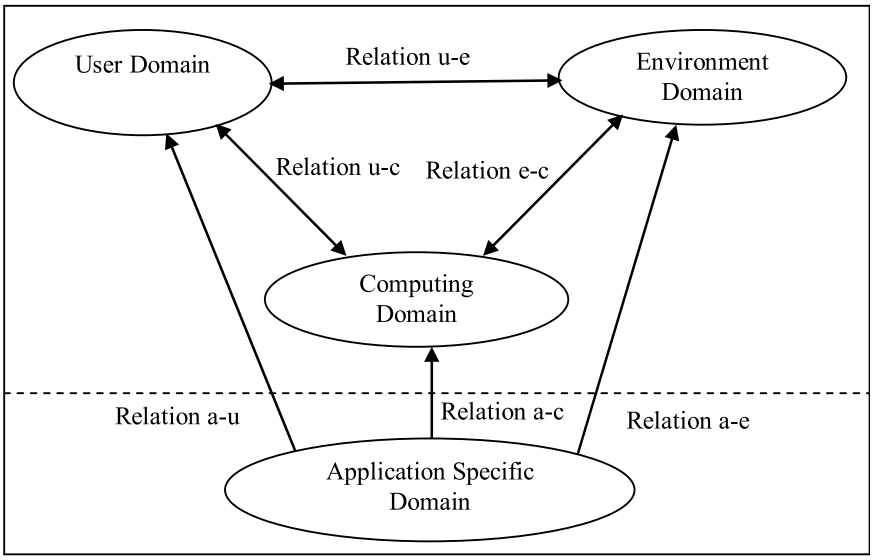


Fig 1. Types of Context Information

3.2 The Formal Definition of Context Information

Before dealing with the representation of context information, we will present a series of definitions that formalize this representation and further elaborate on the definition of context information that we have introduced in section 3.1.

Definition 1:. An entity O_i is defined as a tangible or intangible real-world entity, such as a device, a place, a CD, an electronic product such as an mp3 file or a customer order.

Definition 2: The *context domain* is a high-level abstraction which partitions entities into the following categories: {*user, computing, environment, application-specific}*.

Definition 3: An entity O_i, may be modelled using a number of properties that describe aspects of the object O_i and a number of relationships, which describe how the entity relates to other entities. The set that includes all attributes (i.e. both properties and relationships) for entity O_i will be denoted as $A_i = \{a_{i,1}, a_{i,2}, \ldots, a_{i,m}\}$. Note that relationships may model the "part-of" semantics.

Definition 4: The *state* of an object O_i during a particular transaction *t* will be denoted as $S_i(t)$ and is derived by assigning a concrete value to each attribute of A_i. Each value may be atomic, record-typed, array-typed or any combination of the above. Orthogonally to their types, attribute values may be *sensed* (i.e. be gathered from physical or logical sensors), *explicitly provided* (i.e. the user enters the value) or *derived* (i.e. other values are processed to compute the value of the particular attribute). This is effectively the *context information* of object O_i during transaction *t*.

Definition 5: The context of a transaction *t* will be denoted as *C(t)* and is defined as the collection of all states of objects O_i which can be perceived as relevant to the user, the executed application or their interaction during the transaction *t*. Formally, $C(t) = \cup k S_k(t)$ for all objects O_k that are considered relevant.

According to the definitions above, context information for a particular transaction can be denoted as a set of quadruples *(object context domain, object, attribute*, *value*). The element *object* denotes the object to which the particular piece of context information corresponds, e.g. a particular stock share (which includes information regarding "share description", "share daily prices", "shareholders registry," etc.), location (which includes information regarding the longitude and latitude, but may include more "high-level" information such as "office", "home"), etc. The element *object context domain* specifies the context domain to which the *object* belongs; the element *attribute* identifies the particular attribute that is measured; and the *value* element gives the exact value for the attribute within the specific transaction.

Note that the level of abstraction considered in the selection of an entity is dependent on the requirements of the application domain and the choices of the systems' analysts. For instance, a PDA may be modelled as a single entity O_{pda}, having attributes representing its keyboard, screen, etc, or these parts may be modelled as separate entities and be connected to the O_{pda} entity through relationships of type "part-of" (cf. "FIPA Device Ontology Specification" [18]). These modelling choices do not affect the generality of the modelling method, since the goal, of capturing all the required context state information, can be accomplished independently of whether this information has been represented as a value of a property within a linked entity or as a component of a structure-valued attribute of a single entity.

3.3 The Metadata of Context Information

Context information is greatly heterogeneous regarding the characteristics of its elements – e.g. quality and persistence [10], a fact that stems both from the variety of sources the information element values are gathered from and from the nature of the elements (e.g. some are time-varying while others are permanent). *Sensed context information* (i.e. context information, the values of which are gathered through sensors) is typically dynamic, in the sense that it changes often and it is not uncommon to be imprecise owing to sensing errors, sensor failures or network disconnections, or to delays introduced by distribution [8]. On the other hand, *explicitly provided context information* may be dynamic or static – in the sense that it changes less often - and is prone to human errors [9]. Finally, *derived context information* may be imprecise due to imperfect inputs or deficient derivation mechanisms [9]. Therefore, to reflect these aspects of the context information elements, which may be exploited by their consumers, we define a set of attributes for the elements, which are effectively the *context information metadata*. These attributes are the following:

- *Source attribute:* this refers to the provider/sensing service of the value, e.g. user, physical sensor, application service.
- *Timestamp*: this reflects the instance that the value was produced; a granularity of second is expected to be adequate for most applications. This piece of meta-information is of more value when dynamic context information is considered.
- *Confidence*: this attribute reflects the degree of belief that the value is correct. When dynamic context information is considered, *confidence* may be considered to be a "probability of correctness", and can be determined after a series of trials (e.g. testing the accuracy of the sensing device or evaluating the relevance of an advertised product with the actual purchases of user, in order to assess the accuracy of user preferences). For static context information, the *confidence* attribute may reflect the certainty of its provider that the provided value is correct (credibility)[2].

[2] Some researchers distinguish between *accuracy*, which is an objective measure of the "probability of correctness" and *confidence*, which is a subjective measure of the same quantity. Since objectivity cannot be easily defined and quantified, we will adopt a single measure for this aspect.

- *Frequency*: this attribute expresses the periodicity of recording for the specific value. It can be used both for static or dynamic context information.
- *Validity Period*: this attribute states the time duration for which the information is valid, i.e. it may be used without obtaining an up-to-date value.
- *Metric*: this expresses the unit/scale used to measure the context information element value. This is necessary when different units/scales are used by different providers to measure the value, in order to allow value interpretation and meaningful comparisons. We have opted to include the metric as part of the metadata, rather than as part of the value (as for instance is suggested in [23]), in order to facilitate arithmetic operations and comparisons on the values; for instance, if the value of temperature is "9" and the accompanying metadata indicates that temperature is expressed in Celsius degrees, then it is straightforward e.g. to compare the value to a threshold of "15", which may be included in a service's adaptivity condition. If the metric were included in the value, to obtain a meaningful comparison between "9^{o}C" and "15^{o}C" would necessitate a parsing step and therefore encumber the process.

Note that it is not mandatory to assign all the aforementioned meta-information attributes to each context information element. The attributes assigned to each context information element will be determined by examining the means for acquiring its value, whether the value is dynamic or not, and the aspects of the value's dynamic nature. Metadata may be exploited by applications that make use of the context information, towards the end of assessing the quality of the context information and to select that data that satisfies certain requirements.

4 The Representation of Context Information

4.1 Diagrammatic Notation for Context Information

Mobile context-aware applications need to determine their present situation of usage; as already stated, this is achieved by modelling the context information and obtaining the pertinent values (and the associated meta-information), which will finally be exploited for the provision of adaptive services.

A diagrammatic notation of the context-related concepts and their interrelationships, enhanced with the appropriate annotations that will quantify and clarify certain concept aspects, is expected to be very helpful in the stages of application requirements analysis, design and development, since it may serve as a "common language" between roles involved in these stages (end users; analysts; designers; and programmers). In the area of general-purpose application development, Unified Modelling Language (UML) has been established as a standard and commonly accepted means for defining and modelling concepts and components of a software system in an implementation-independent fashion, thus allowing for the required flexibility at the implementation stage. UML tools can be appropriately extended to include the specific aspects of context. In particular, *use case diagrams* – which describe the tasks within a system and the actors involved in them - can be extended to include the different context states/situations under which the tasks are carried out. Furthermore, UML *class diagrams* can formalize the way context entities, as well as their attributes and interrelationships, are captured, documented and designed at a conceptual level. Class diagrams can be extended to include the necessary annotations for accommodating the particularities of context information, in a fashion that is easy to understand and apply and that is fully consistent to UML's object-oriented philosophy. The context therefore -through a series of design activities adhering to UML paradigm- will be mapped into concrete *context information* elements, which will be subsequently exploited for performing adaptation within the services (see figure 2). As shown in figure 2, the Context of a task which is performed by a user, consists of the state of the objects Oi, Ok, Om, etc which play a role in the situation under which that task is performed. The state of these objects will be transformed in Context Elements (which will be used in adaptation process) through a modelling process that makes use of the extended Use Case and Class diagrams of UML,.

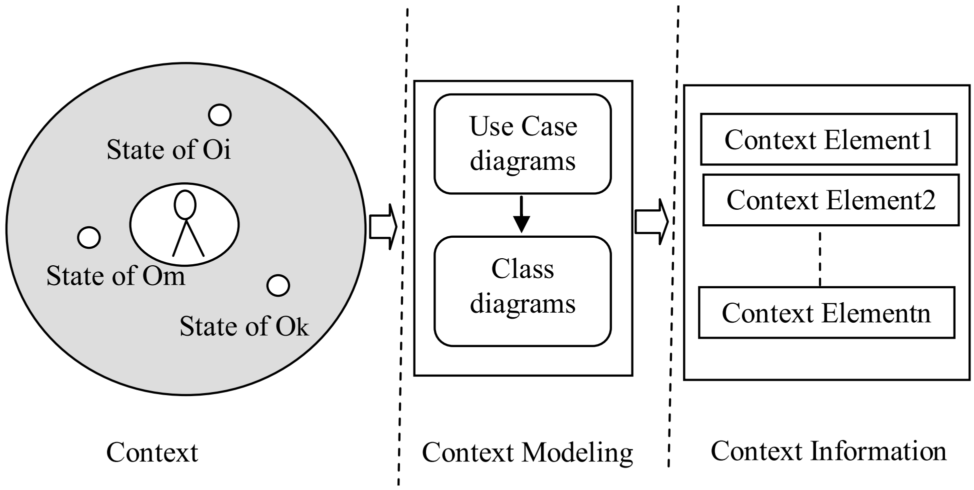


Fig2. The identification of context elements

4.2 Context-Aware Use Case Diagrams

According to [5], an initial, high-level and abstract description of the system to be developed will help to draw conclusions on fundamental system characteristics. UML use cases have been successfully used to this end in the context of typical software system development. The goal-driven nature of UML use cases leads to a user-oriented requirements model, through a hierarchical structure of goals and sub-goals focused on user intentions. On the goals and sub-goals documentation (both diagrammatic and textual), the operations (tasks) that are carried out in order to fulfill the goal/sub-goal are described. This description, however, does not depict the different conditions and states under which the operations (tasks) are performed (e.g. weather conditions, change of value conditions, traffic situations), which constitute the Context of the operations (tasks). However, the use case diagrams, after being appropriately extended, may support the capturing of each basic operation's Context (task's Context). This could be done through a description of different scenarios/situations, in the framework of which each basic operation (task) will be performed.

Our ultimate object is to identify the entities that are involved in the aforementioned scenarios/situations, in order to model their characteristics so that they can be used in order to provide adapted services. Therefore, while describing the different scenarios/situations under which an operation (task) will be performed, the *real world objects* that will play a role in these scenarios/situations will be identified, taking into account the dimension/aspect of *user mobility*. These objects correspond to the *entities that are potentially relevant* to the operation under consideration. The "degree of relevance" of an entity to the user-application interaction, as described in the definition of context information, will be specified after identifying the attributes of the entity that can be measured and quantified and determining if these attributes can be exploited by the application so as to provide adaptive services.

Naturally, the relevant real-world objects will vary according to the m-commerce application category; however, some specific objects can be considered to be quite probable to be relevant to most m-commerce applications. Within the user domain, the most probable real-world objects are those answering the questions "*who* is the user" (identity, preference, mood), "*where* is the user" (location), "*when* is the user" (time) and "*what* does the user do" (activity, intention). Within the *computing domain*, the most prominent entities are the *access device*, the *communication network* and the *communication channel* (e.g. screen display, audio feedback, vibration etc). Within the *application-specific domain* the main objects corresponding to the application's main focus should be examined. For instance in a mobile advertising application, the (advertised) *product* is a strong candidate. In a mobile banking application the *banking account* and *payment* objects would be considered. Similarly, in a mobile information provisioning application, the information item itself (e.g. "weather", "football match", "theatrical play") is of high importance. The object list derived from the process described above, can be enhanced with any other object whose state can be exploited to enhance service adaptivity.

The process described informally above can be more formally described as a series of steps as follows:

- **Step 1:** Each individual operation/task T, is described textually and documented through the "usual" use cases diagrams, illustrating the system's main goals, which are related to the product's commercial value. Note that in this step, the core system functionality is captured without taking into account the context, e.g. "the user is searching for parking space" (task T), so the (initial) service which will support the task will give "a list of parking places".
- **Step 2:** In this step, the different scenarios/situations under which the task T is performed, must be defined. Business analysts and technology experts will assist in this step, identifying cases that context exploitation can lead to services with added business value, and exploring the potential offered by the technology to enhance user services. So, in this step the scenarios/situations and their corresponding potential adapted/enhanced services will be defined. A scenario/situation has the form "on a rainy day, user X, equipped with a WiFi- and Bluetooth-capable mobile phone, is driving in the centre of city Z searching for parking space".
- **Step 3.** Critical factors for each scenario/situation are identified during this step. These include (a) the objects involved in *how* the information/service will be

delivered to the user, and (b) the actors that are (implicitly or explicitly) present in the scenario and will possibly affect the user's decision regarding the purchase of product. For instance, the scenario/situation listed in step 2, includes the specification "rainy day" which is related to the actor "weather conditions" and can lead the user to choose an indoor parking space (as opposed to outdoor parking). The objects listed in the scenario/situation text are a starting point for this process. However, some of these objects may be considered irrelevant (or infeasible to exploit) and be disregarded later on (step 8), whereas other objects that are not apparent in the text may be added to the critical factor list (step 7).

- **Step 4:** Scenarios/situations and their corresponding probable adapted/enhanced services (identified in step 2) are elaborated on, to determine the details on *how* the application will be affected by the context elements in each of them. Within this step, the final adaptive services, which are related to the application's commercial value and eventually will be offered by the system, will be concluded after answering the following questions:
  - How will the core information required by each task be shaped under the specific conditions of each scenario? *(information adaptation)*
  - Which additional information and/or functionality should be offered to the user under the specific conditions of each scenario? *(functionality adaptation)*
  - How will the information/functionality be delivered to the user? (*presentation adaptation)*
- **Step 5:** In this step, the attributes of the critical factors of scenarios/situations should be determined. Additionally, the relations of these attributes with the scenarios/situations should be captured and documented. To facilitate and formalize this process, critical factors -together with their identified attributes of interest- and

scenarios/situations –together with their corresponding adapted services- are listed in a table. The columns of this table correspond to the scenarios/situations (with their adapted services) and rows correspond to critical factors (context objects) and their attributes. The (i, j) cell of the table is checked (marked with an X) if the attribute listed in the i-th row plays a role in the scenario/situation listed in the j-th column (see table 1). Therefore, each scenario/situation can be determined using a logical expression e.g. (($attr_1$=val_1) and ($attr_2$=true) and … and ($attr_n$=val_n)).

Note that table 1 does not intend to capture the full complexity of mobile commerce application contexts on its own right; its purpose is to correlate the critical factors (context factors) identified in step 3 with the scenarios/situations identified in step 2 and their corresponding probable adaptive/enhanced services identified in step 2 and step 4. Therefore, table 1 serves the purpose of documenting *which context factors are relevant to each scenario/situation*. At this stage, the critical factors represent real-world objects that comprise the context, but for which further elaboration and modeling is required so as to provide the entities of definition 3. A software modelling process, such as UML [19] or Booch methodology [20], will be subsequently employed to determine the entities' respective properties and relationships and shape the software artefacts for the entities' representation and manipulation.

Table 1: Context-Situation-Adapted Services matrix

Scenarios/Situations-Adapted/Enhanced Services Critical Factors (Context variables)	Scenario1/ Situation 1 – Adapted/ Enhanced Service 1	Scenario2/ Situation 2 - Adapted/ Enhanced Service 2	Scenario3/ Situation 3- Adapted/ Enhanced Service 3	…….
Environment				
Temperature		x		
Wind	x			
Sunshine		x	x	
Location	x			
Device				
Channel	x			
Screen size		x		
Memory			x	
Parking Place				
Kind of parking	x			
Distance from city center		x		
Capacity	x		x	

- **Step 6:** Within this step, we revisit the list of critical factor attributes, to recognize attribute combinations that can be effectively modelled as new scenarios/situations which can increase the application's commercial value. For example, we can consider combining the weather conditions with the user location and intention, such as "the user wants to go to a specific shopping mall (intention/location) on a day with heavy rain (weather conditions), thus the parking space s/he seeks should be *very close* to the user's target location". The new scenarios/situations will be documented, and an iteration of the methodology will start from step 3. This is necessary since each new scenario/situation identified in this step needs to be analyzed and documented. Additionally, new critical factors, which may be recognized in each new scenario/situation, should be considered in combination with (a) each other and (b) critical factors already documented for other scenarios/situations.

- **Step 7:** Steps 1-6 are iteratively performed to identify more scenarios/situations and/or critical factors, which potentially could affect the application's commercial value, that were not initially considered. In the example of the parking space for instance, *time* may emerge as a critical factor, after taking into account that the price charged by parking garage proprietors varies according to the time of the day and thus the user should be informed regarding these variations. In this manner, factors can be incrementally taken into account and details can emerge progressively as suggested in the stepwise refinement approach [16].
- **Step 8:** Technology experts, developers and business analysts review the critical factor attributes, to determine the degree to which the automated collection of their values is feasible. This examination includes *both* the technological potential to determine the values of the attributes of interest *and* the cost implied for performing the automated collection. Additionally, among the issues that should be examined is the cost of incorporating the necessary sensors, developing or leasing software services, how much the user will be charged *per service use*; all these factors may affect the population of the target group. As a result of this review, certain adapted/enhanced services may be dropped or the degree of automation may be reduced (e.g. some context's attribute values are provided by the user instead of being sensed or derived).

- **Step 9:** In this step a final iteration over all previous steps is performed. Table 1 is carefully reviewed, to identify innovative services that were not included in the original specifications, which will provide a further competitive advantage to the application and reinforce the user's loyalty to the particular application.

After the application of the aforementioned methodology and by using the information presented in Table 1, the extended UML use case diagrams shall be composed. In these diagrams –under a separate section entitled "Context-Related Issues"– the critical factors comprising the contest of an operation/task will be documented. Additionally, for each scenario/situation within the context of which the specific operation/task is carried out, the corresponding critical factors as well as the recommended adaptive/enhanced services will be documented. An example of an extended UML use case diagram is presented in Figure 4. The critical factors will subsequently undergo further processing –as described in paragraph 4.3– in order to produce the entities of the context as they have been defined in definition 3. The documented situations along with their respective critical factors and adaptive/enhanced services will be used for the design of the initial service's adaptivity process, which, however, is beyond the scope of this paper.

4.3 Context-aware Class Diagrams

The context factors comprising the context, together with their attributes (as these have been determined during steps 3 and 5 of the previous subsection), will undergo a typical software engineering modelling process (e.g. UML [19] or Booch methodology [20]), to finally determine the entities' (cf. definition 1) respective properties and relationships (cf. definition 3). These entities and their interrelationships will be illustrated as enhanced UML class diagrams, in which the enhancement refers to the inclusion of the special characteristics of context information and more specifically:

- the *dynamicity of the value of each attribute*, given that context information is distinguished in *static* and *dynamic*, depending on how often it changes.
- the *acquisition method* of each attribute value (sensed, explicitly provided, derived).
- the *metadata information* accompanying each attribute.
- the *need to record past values* for the information.
- the possibility that *the type of a related entity may change*. This is important in adaptive services, since such a change may make new context items or sensing methods available or may
- terminate the availability of context items/sensing methods, which in turn may trigger changes to the user interface, processing or available data. For instance, a change in the user's location from an instance of "office" to an instance of "shopping mall" may lead to the withdrawal of the user interface item "Read corporate memos" and establish the item "Get offers".

An example of an enhanced class diagram is illustrated in fig. 3. The notations used in this diagram are described in the following paragraphs.

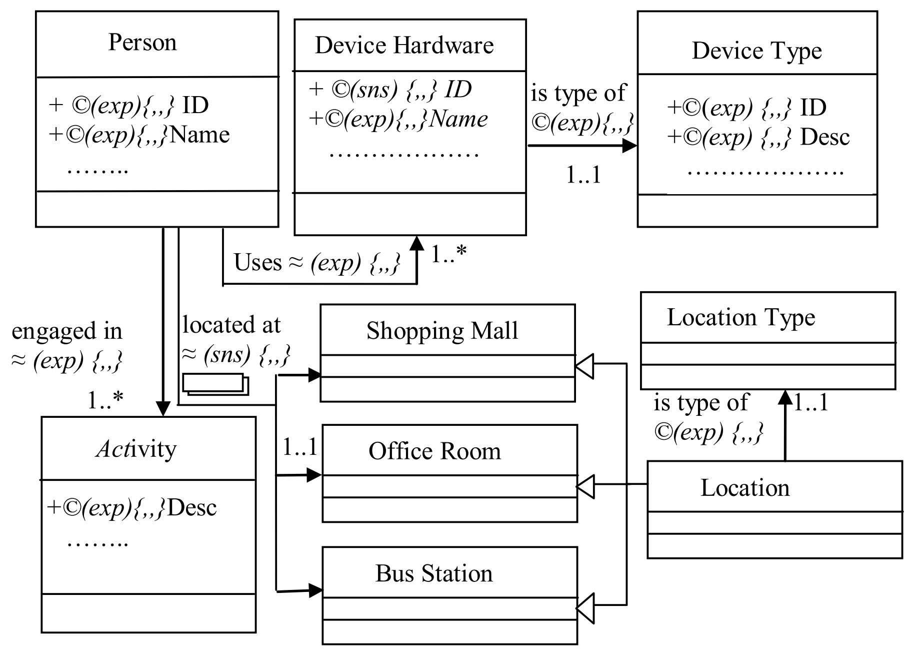
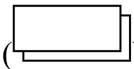
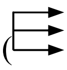


Fig3. Context-aware class diagrams

Consistently to the UML class diagrams, entities are represented using rectangles with three areas. The top area contains the class name, the middle area lists its properties and the bottom area lists the basic operation it provides. Its relationships, (associations), with other entities are denoted using arrows (→), which are labeled with the relationship cardinality (e.g. 0..1, 1..1, 1..*) [19]. The special-type *generalization links* (—▷) denote the parent/child class relationships. Each attribute or relationship may be labeled with additional marks that denote the special nature of context information as follows:

- *Information dynamicity* is denoted using:
  - the symbol © for static information
  - the symbol ≈ for dynamic information
- The *acquisition method* is denoted using:

    - (sns) : for sensed information
    - (exp): for explicitly provided information
    - (drv): for derived information

- The *metadata* associated to the context information are denoted as a series of values, with each value corresponding to a piece of metadata, e.g. {source, timestamp, confidence, frequency, validity period, metric}
- the *need to record past values* for the information is illustrated through a double rectangle ().
- the *possibility that the type of a related entity may change* is denoted using an arrow splitting to multiple ends (), one end for each possible type.

4.4 Example of a Scenario Analysis

In order to demonstrate the use of the methodology for context determination and representation, as well as the relationship of the methodology with the definitions of context and context information given above, we will describe how the methodology is applied in the case of designing a context-aware mobile shopping application supporting the following scenario:

> *"We consider a user, who is a consumer of organic foods, who is located in some area –e.g. the center of a city or suburb X- and is looking for a specific product"*

The mobile shopping application should at least offer information regarding the organic products offered by shops –within the user's vicinity- that are "subscribed" to its database.

Step 1: We use standard software engineering practices (e.g. [19]) and classic use case diagrams to define the operation under consideration without any reference to the context, i.e. the "reduced" operation "user U is searching for an organic product P". This is the task T and the supporting "initial service" will provide "a list of shops that offer product P along with the product's corresponding prices"; details of this procedure are beyond the scope of this paper and are not given.

Step 2: We elaborate on the context C of user U while s/he is executing task T, i.e. the context C(t) of definition 5. The context in question, as humans perceive it, is a set of situations that surround operation T (cf. the definition of context in section 3.1); therefore, to capture this aspect of the context, we begin describing different situations/scenarios under which the user may execute task T. At the same time, we identify in which respects that task T – as well as the initial service offered by application - should differentiate itself under each situation/scenario. Effectively, this step allows us to gain insight to the application's adaptivity under certain situations, and the adaptivity conditions will indicate the context factors that should be taken into account (step 3). This is inline with the definition of context listed in section 3.1 which states that "*…while the **relevance** of the entity to the user-application interaction can be derived from the potential **to exploit** the information describing the entity state to optimize this interaction, so as to maximize the commercial value of the application.*"

In the following, we will list some of these situations and the associated actions, carried out for each of them.

1. ***Situation 1:*** *"The user, who is using a particular access device, performs task T, i.e. s/he is looking for a specific organic product".* This is considered to be a key one, since it is effectively a factor in all situations and, moreover, it is the minimum information that is supplied by the access device to the application. Technology experts and business analysts will suggest variations for the initial service ("a list of shops that offer the product P along with the product's corresponding prices"), taking into account the characteristics of the device and the

available information for user U (which may also be the user group that U is classified in). So the application could also offer added-value services, such as "Display, additionally, the ingredients of the searching product", since an organic food consumer can be deemed to require this information.

Subsequently, we broaden the context for the particular task, through the elicitation of more context aspects.

2. ***Situation 2.*** (While the user) "*is in motion (e.g. walking, driving), s/he performs task T"*. In this case, the added value for the task could be *"the information for the product could be conveyed in an audible manner"*.
3. ***Situation 3:*** (The user) *"during a very hot or a hot or a rainy day is performing task T"*. In this case, the added value for the task could be *"offers from nearest stores should be placed first on the list"*.
4. ***Situation 4:*** (The user) *"at a specific hour of the day/day of the week is performing task T"*. In this case, the added value for the task could be *"the stores list should be filtered to retain only entries for stores that are open during this time"*; alternatively closed stores could be retained in the list but be presented in a different area of the user interface.
5. ***Situation 5:*** (The user) *"is at a specific spot and is performing task T"*. In this case, the added value for the task could be to place first in the list the offers from the nearby stores.
6. ***Situation 6:*** (The user) *"is within a specific shopping mall and is performing task T"*. In this case, the added value for the task could be to show only offers from shops of the specific shopping mall (other offers could be hidden or be accessible through a specific area of the user interface, similarly to situation 4). Note that this

situation is different than the one in situation 5, since the added value is based on a higher-level interpretation of the location.

Step 3: During this step, we identify the critical factors for each situation/scenario. The situation is described through an abstraction that includes the real-world entities (cf. definition 1) relevant to the situation, which effectively are the critical factors for supporting the situation.. The critical factors are identified in this step at entity level, while determination of individual entity characteristics is deferred until step 5, since the list of entities may be enriched in step 4. Regarding the situations identified in step 2, the critical factors that can be identified in this step are as follows:

1. ***Situation 1: "****The user, who is using a particular access device, performs task T, i.e. s/he is looking for a specific organic product".* The critical factors that can be identified through this description are the "User", the "Device" and the "Organic Product".
2. ***Situation 2.*** (While the user) "*is in motion (e.g. walking, driving), s/he performs task T"*. In this case, the additional critical factor is the "User's State" regarding his/her mobility.
3. ***Situation 3:*** (The user) *"during a very hot or a very cold or a rainy or a snowy day, is performing task T"*. In this case, the additional critical factors are the "Environment Temperature" and the "Rain" and the "Snow".
4. ***Situation 4:*** (The user) *"at a specific hour of the day/day of the week, is performing task T"*. In this case, the additional critical factor is the "Time"; especially the attributes current hour, date, and day of the week.
5. ***Situation 5:*** (The user) *"is at a specific spot and is performing task T"*. In this case, the additional critical factor is the "Location of the User" (longitude and latitude).

6. ***Situation 6:*** (The user) *"is within a specific shopping mall and is performing task T"*. In this case, the additional critical factor is "User's Location" (interpretation of the longitude and latitude at a higher level).

Step 4: Within this step, we further elaborate on the variations of the initial service and the context-enhanced/value added services that have been identified during step 2, to determine how information will be conveyed. This step may produce additional value-added services, probably as combinations of situations identified in step 2. For instance, in situation 4, an additional column may be added to accommodate the closing hour of the shops, and/or introduction of color-coding for indicating how much time is left until the closing time of each shop (e.g. red for shops that will close in 15 minutes; orange for shops that will close in one hour; green for those that will remain open for more than one hour). Device characteristics may also be considered to determine which is the best way to perform to convey the information (e.g. if the device display is too narrow, color coding is preferable than the additional column). The enhancements described above, bring up the "Shop" and possibly the "Device" as additional critical factors for this situation.

Step 5: Within this step we document the situations along with their corresponding adapted/enhanced services and the associated critical factors (real-world entities) together with their characteristics. Recall that in step 3 we examined the different situations under which a task may be carried out and we identified the real-world entities (critical factors) that are associated with each situation. In step 5, we attempt to identify the individual entity characteristics that are of interest for delivering the services and the added values described in the previous steps. These characteristics effectively model the *state* of each real-world entity, Subsequently, these characteristics, through the appropriate modelling process (cf. subsection 4.3), will finally form the "*state of an entity*" mentioned in the definition of context information (cf. subsection 3.1), or equivalently the "*state* $S_i(t)$" of definition 4.

Note that the characteristics within this step are not necessarily simple data elements (e.g. attributes with atomic values [24]), but may be more complex, such as composite values, repeating values or relationships with other entities (part-of relation). Table 2 depicts the entities involved in selected situations, and the characteristics of interest for these entities. Note that table 2 includes a situation (situation of step 6) that is not listed above yet, it will be identified in step 6, below.

Step 6: Within this step, context characteristics (i.e. context variables) are combined to explore the potential of identifying new scenarios/situations. For example, the month part of the date (which indicates the season of the year) may be exploited to formulate the following situation scenario/situation: (the user) *"during a particular season of the year performs task T"*. The added value in this case could be, for instance, to display seasonal products (e.g. seasonal fruit and vegetables, organic suntan lotions etc.). Since a new scenario/situation has been identified, an iteration of methodology steps from step 3 onwards is performed, to bring out new characteristics of the critical factors or even new critical factors. In our example, the new characteristics that emerge and will be identified upon the iteration of step 3, are the "season of the year" and the "seasonal product indication" (part of the application-specific context).

Table 2: Context-Situation-Adapted Services matrix

Situation-Adapted / Enhanced Service Critical Factor-Context variables	*Situation 1 of step 2-Adapted Service*: "User with specific device"- "show additionally the product's ingredients"	*Situation of step 4-Adapted Service:* "a specific hour" - "display open stores using colour - coding"	*Situation of Step 6--Adapted Service:* " a specific season " - "show seasonal products"	*Situation 3 of step 2--Adapted Service: "a very hot day"-* "show only offers from the nearest shops"
User				
ID	x	x	x	x
Name				
Device				
Channel	x	x	x	x
Screen size	x	x	x	x
Environment				
Temperature				x
Sunshine/Rain				x
Date		x	x	
Organic Product				
Ingredients	x			
Product's Shop		x		
Related Products			x	

Step 7: Steps 2-6 are iterated, in order to identify new scenarios/situations and new critical factors –or their characteristics- which were not identified over the initial iteration. In our example, no new scenarios/situations or critical factors are identified during this iteration.

Step 8: Within this step, the feasibility, cost and degree of automation for collecting each context variable is assessed. For instance, in order to represent the situation 3 of step 2 in terms of context variables, we need to know the current temperature. This can be accomplished if temperature sensors are embedded in the device, or the device can collect this information from sensors in its vicinity. Since such devices are rare, business analysts may suggest dropping the specific value-added service since it is not cost-effective. Technology experts, however, may propose that for GPS-equipped devices it is possible to have the device communicate to the application server the current coordinates (longitude/latitude), and then the server may query a weather service (e.g. [25]) to retrieve the current weather conditions at the user's location.

Step 9: Within this step, table 1 is carefully reviewed and an iteration over steps 1-8 is performed, to identify possible innovative services that will constitute a competitive advantage for the application and will reinforce the user's loyalty to the application. For example, in situation 6 "the user performs task T while s/he is inside a shopping mall", the added value is "to show only offers from shops of the specific shopping mall". This service might be enhanced with additional functionality, for instance to display special offers from shops within the same mall, and/or suggest an additional nearby store that sells the particular product at a significantly lower price.

Subsequently, context analysis performed in steps 2-9, is integrated into the use cases produced in step 1, to create the context-aware use cases. More specifically, we will incorporate i) a section listing the relevant critical factors and their characteristics and ii) a section listing for each situation, the following items:

- a verbal description of the situation,
- the associated critical factors and
- the related adapted/enhanced services

An excerpt of these sections for the example presented above is depicted in fig. 4.

Fig4. Excerpt of sections extending the initial use cases to incorporate context

Task T: A User is looking for a specific organic product

Initial Service: Show a list of shops that offer the specific product, along with the product's corresponding price

CONTEXT-RELATED ISSUES

SECTION 1: CRITICAL FACTORS (FOR ALL SITUATIONS)

Critical factor: User

Characteristics: ID, Name, …

Critical factor: Device

Characteristics: Channel, Screen size, Memory

SECTION 2: SITUATIONS-CRITICAL FACTORS-ADAPTED SERVICES

Situation 1

Description: The user, who is using a particular access device, performs task T (looking for a specific organic product)

Critical Factors: User, Device, Organic Product

Adapted/Enhanced Services: Display the ingredients of the product; use fewer/smaller images

Situation 2

Description: (While the user) is in motion (e.g. walking, driving), s/he performs task T

Critical Factors: User's State

Adapted/Enhanced Services: The information for the product could be conveyed in an audible manner

When this procedure is concluded, all real-world entities and their characteristics will have been documented. Through the modelling process described in subsection 4.3, the software entities and their attributes will be defined (cf. definition 3). The set of (attribute, value) pairs for all attributes of an entity, will effectively be the particular entity's context information $S_i(T)$ in the context of task T (cf. definition 4). The context of task T is the union of the states of all involved entities, or, more formally, $C(T) = \cup_k S_k(t)$. The outcome of the modelling process will be documented through context-aware class diagrams, as described in subsection 4.3.

5 Related Work

Most context representation models that have been presented insofar [14], suggest that an entity-based approach should be adopted for context representation, with the exceptions of the *key-value* models and *logic-based* models. Entity-based approaches include Markup Scheme Models, Object Oriented Models, Graphical Models and Ontology-based Models. All these models finally represent the context's real-world entities as a set of interrelated entities, using the "data artefacts" prescribed by each adopted approach. These models, with the exception of Graphical Models, have both the advantages and the limitations implied by the pertinent technology, and therefore may represent specific context aspects with greater or less success.

Graphical models, on the other hand, represent context concepts in a graphical environment, making use of special notations. A characteristic example is the object-role model (ORM) extension proposed by Henricksen [9]. In this model, context information is distinguished to *static* and *dynamic*, with dynamic context information being further subdivided to *sensed*, *derived* and *profiled*. The term *profiled* is used for information that has been entered by the user and rarely changes. At this point, we have substituted the term *profiled* by the term *explicitly provided*, to specifically include the information provided by the user at runtime. Furthermore, *static* information is not necessarily explicitly provided (as implied by the absence of subcategories in Henricksen's classification), but can be also sensed, derived and explicitly provided (profiled). Therefore, we adopt the classification *sensed*, *derived* and *explicitly provided* for all elements of context information. Furthermore, Henricksen's model introduces the concept of *simple associations* for capturing the entities' attributes and the concept of *composite associations* to capture the relationships between different entities. Composite associations are further subdivided to i) *collection associations*, used for relationships that are not mutually exclusive (linked with "and" operator) ii) *alternative associations*, used for relationships that *are* mutually exclusive (linked with "or" operator), as well as for alternative representations of the same information and iii) *temporal associations* for capturing historical data for attributes. Finally, the object-role model (ORM) is mapped to a relational schema, to facilitate implementation on top of a conventional RDBMS. The main drawback of these representations, however, is that the graphical notations employed for representing context, require excessive space (each entity, attribute or metadata element may be assigned its own placeholder on the diagram), and therefore they are only suitable for applications with a limited number of context elements.

Many recent works adopt ontologies for context representation, such as Tao Gu, et.al [26], Panu Korpipää [27], to align with the developments in the area of semantic web, which is nowadays widely used in the m-commerce domain. [26] is oriented towards use by *intelligent environment* applications, i.e. environments that are *sensitive and responsive to the presence of people*. The central entities in these systems are the *location* of the user and the *ordinary, usual activities* s/he executes while s/he is present at the specific location. These environments typically use a limited amount of the context (typically comprising of user identity, location and activity, as well as computing and communication infrastructure), and only the "location" element of the context is further elaborated on. Moreover, intelligent environment applications that make use of the context, usually have a broad target –such as "assisting the user in a range of customary activities that s/he is often engaged in, during his/her presence at a specific location"-, rather than a focused target, - such as mobile ticketing, mobile retailing, mobile broking, etc.

The work presented in [27] targets mobile applications; however, the concepts of context are limited, since the greatest portion of the proposed context management system is executed on the mobile device and, therefore, the context factors considered are reduced to save on hardware resources. The ontology described in [27] uses the concepts of *location*, *time*, *user*, *device* and *environmental factors* (sound, light, temperature, humidity), without, however, elaborating on inter-concept relationships or the application specific-context, which is of particular importance for m-commerce applications. Finally, this work includes an API, implemented on the Symbian platform; the API is designed for procedural technologies, hence some modifications will be needed to make the API more suitable for other implementation environments (e.g. object-oriented platforms).

The Semantic Web Architecture for Context-Awareness, presented in [28], [29], [30], [32], [33], is an important work addressing the issue of context organization and provisioning using web technologies. In this architecture, context is represented using ontologies and is organized around the central concept of the user as *context resources*, corresponding to web services. A very important concept in this work is that of *service invocation rules*, through which context attributes are mapped to external services. The work presented in [29] includes also the aspect of *dynamic knowledge*, captured through *rules*, i.e. a series of conditions and associated actions, with an action firing when the respective conditions are satisfied; this provision, however, is mostly related to adaptivity, rather than to context representation itself.

In the area of m-commerce, the role and importance of context has been stressed [34], and approaches for its capturing, representation and exploitation in different areas of m-commerce have been proposed, e.g. mobile advertising [35] and mobile information provision [31].

Our proposal, similarly to the works reviewed above, follows the entity-based approach to context representation, since through this approach the context (of an operation) can be specified by focusing on the context of entities that comprise the context. Thus, the complexity of the context as a whole is finally reduced to the level of its constituent entities, and thus the overall problem becomes more manageable. The approach proposed in this paper extends the works reviewed above in the following aspects:

a) it provides a theoretical foundation for the context, the context information and the relationship between them in the domain of m-commerce applications.
b) it gives a methodology for specifying context information for m-commerce applications, through a series of steps that lead from the broad notion of *task context* to specific *context information elements*. These elements are effectively properties of the real-world entities that are relevant to the context of a task. Moreover, the *part-of* relationships in a real-world entity representations (e.g. the FIPA ontology [18]), can be exploited to further decompose entities and reduce the complexity, since context analysis is thus performed on "simpler" entities, and the results are synthesized to obtain the whole of the required context.
c) the methodology includes provisions for aspects of context that are particular in the m-commerce domain, such as the commercial value, the business opportunities, the

cost-effectiveness and so forth. It also considers the application-specific domain, which is of high importance for m-commerce.

d) it introduces a diagrammatical tool for context representation. This tool consistently extends UML diagrams, which are widely accepted and are particularly suitable for use in an object-oriented approach for context capturing, while they also facilitate the development with object-oriented technologies. Additionally this tool -as opposed to graphical models- doesn't require excessive space and therefore it is suitable for applications with a large number of context elements.

e) The context capturing methodology does not bind to a particular implementation technology (e.g. web services or procedural programming models), allowing thus more degrees of freedom for implementers to choose the most appropriate means for realizing the services.

Table 3 summarizes the differences and overlaps between existing approaches and the proposed methodology for identification and specification of Context Information.

Table 3. Differences and overlaps between existing approaches and proposed methodology

Methodology for identifying and specifying Context Information	*Proposed Approach*	*Graphical Models Approach (e.g. [9])*	*Ontological Representations ([26][27])*	*Semantic Web Architecture for Context-Awareness ([28], [29], [30], [32], [33)]*
Includes a theoretical foundation of Context	Yes	No; adopts Dey's definition.	No	No
Methodology for identifying and specifying Context Information	Yes	No	No	No; mainly targeted to service adaptivity rather than context specification and representation.
Methodology which reveals context's aspects relative to m-commerce (application-specific context)	Yes	No	No. A few ontologies for particular applications are available.	No; mainly targeted to service adaptivity rather than context specification and representation.
Diagrammatic Representation of context	Yes	Yes	Can be done through ontology. visualizations (c.f. [40]), but in a generic manner, not tailored to the needs of context	Can be done through ontology visualizations (c.f. [40]), but in a generic manner, not tailored to the needs of context.
Suitability of diagrammatic tool for representation of a wide range of context factors	Yes	Limited, due to over-consumption of space.	Depends on the ontology visualization and its capability to scale to effectively present large ontologies ([40]).	Depends on the ontology visualization and its capability to scale to effectively present large ontologies ([40]).
Representation of Context in a manner independent of particular implementation	Yes, context is represented using UML-compliant methods	Mainly targeted to ORM and implementation in relational databases.	Mainly targeted to procedural environments.	Mainly targeted to the Service Oriented Architecture.
Representation of Context suitable for direct use from object-oriented technologies	Yes, context is represented using UML-compliant methods	Partially, due to the orientation to relational databases.	Ontologies can be used with object-oriented technologies, the approach however is mainly targeted to procedural environments.	Ontologies can be used with object-oriented technologies.

6 Conclusion

The exploitation of the context within which a mobile commerce application runs is a key to its successful operation, but also further complicates the process of developing such an application. In order to formalize and facilitate this process, it is necessary to promote a common understanding of the concepts related to the context, partucularly when questions such as “what context parameters should be taken into account for the system under development?”, “which sensors and derivation algorithms are required?”, “which context will trigger which services?” and “how will the services adapt, according to the context?” need to be answered.

The development of a formal framework for the representation of the context-related concepts, the introduction of a methodology that will allow the identification of the relevant context parameters and an extension of UML diagrams to serve as a tool for documenting context requirements are presented in this paper, aiming to facilitate the development of context-aware mobile commerce applications. Our proposal can offer elevated exploitation of context information and consequently facilitate the development of innovative services with added commercial value and higher usability. The explicit documentation of context-related parameters, such as sensing services, derivation methods and representation structures, can offer better reusability for these components, since projects involving similar development artifacts can be easily identified. Finally, the association of metadata to the values comprising the context information will increase their value, as applications will have a better knowledge regarding qualitative parameters of these information elements. Metadata standardization can also lead to having multiple providers offering appropriately tagged context information and this, in turn, can result to decreased costs in using and developing context-aware applications, as well as minimization of rollout times [10].

7 References

1. Avery, C., Zeckhauser, R.: Recommender Systems for Evaluating Computer Messages. Communications of the ACM , vol. 40, no. 3, pp. 88-89, (1997).
2. Badrinath, B., Fox, A., Kleinrock, L., Popek, G., Reiher, P., Satyanarayanan, M.: A conceptual framework for network and client adaptation. IEEE Mobile Networks and Applications (MONET), vol. 5, no. 4, pp. 221-231, (2000).
3. Benou, P., Bitos, V.: Developing Mobile Commerce Applications. Journal of Electronic Commerce in Organizations, vol. 6, no.1, pp. 63-78, (2008) .
4. Brown, J., Bovey, J., Chen, X.: Context-aware applications: From the laboratory to the marketplace. *IEEE Personal Communication, vol.* 4, no. 5, pp. 58-64, (1997).
5. Cockburn, A.: Structuring Use Cases with Goals. Journal of Object Oriented programming, vol. 10, no.5, pp. 35-40, (1997).
6. Dey, A., Abowd, G.: Towards a Better Understanding of Context and Context-Awareness. Technical Report 99-22, Georgia Institute of Technology, (1999).
7. Good, M., Whiteside, J., Wixon, D., Jones, S.: Building a User-Derived Interface. Communications of the ACM (CACM), vol. 27, no. 10, pp. 1032-1043, (1984).
8. Gray, P., Salber, D.: Modelling and using sensed context in the design of interactive applications. In: 8th IFIP Conference on engineering for Human-Computer Interaction. Toronto, (2001).
9. Henricksen, K., Indulska, J., Rakotonirainy, A.: Modeling Context Information in Pervasive Computing Systems. In: 1st International Conference on Pervasive Computing (Pervasive), F. Mattern, M. Naghsineh (eds). LNCS, vol. 2414, pp. 167–180. Springer Verlag, (2002).

10. Honle, N., Kappeler, P., Nicklas, D., Schwarz, T., Grossmann, M.: Benefits of Integrating Meta Data into a Context Model. In: 3rd IEEE International Conference on Pervasive Computing and Communications Workshops, pp. 25-29, (2005).

11. Pascoe, J.: Adding Generic Contextual Capabilities to Wearable Computers. In: 2nd International Symposium on Wearable Computers, pp. 92-99, (1998).

12. Reichle, R., Wagner, M., Khan, M., Geihs, K., Lorenzo, J., Valla, M., Fra, C., Paspallis, N., Papadopoulos, G.: A Comprehensive Context Modeling Framework for Pervasive Computing Systems. In: Proceedings of the 8th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS'08), LNCS, vol. 5053, pp. 281-295 Oslo, Norway, Springer-Verlag, (2008) .

13. Schilit, B., Adams, N., Want, R.: Context-Aware Computing Applications. In: 1st International Workshop on Mobile Computing Systems and Applications, pp. 85-90, (1994).

14. Strang, T., Linnhoff-Popien, C.: A Context Modeling Survey. In: 1st International Workshop on Advanced Context Modelling, Reasoning and Management, (2004).

15. Weiser, M.: The Computer of the 21st Century. Scientific American, vol.265, pp. 66-75, (1991).

16. Wirth, N.: Program Development by Stepwise Refinement. Communications of the ACM, vol. 14, no. 4, April (1971).

17. Hoffer, J., Prescott, M., McFadden, F.: Modern Database Management - 7th Edition, Prentice Hall; 7 edition (April 6, 2004).

18. FIPA Gateways TC: FIPA Device Ontology Specification. http://www.fipa.org/specs/fipa00091/PC00091A.html (2001).

19. Miles, R., Hamilton, K.: Learning UML 2.0. O'Reilly Media, Inc. (April 25, 2006).

20. White, I. Rational Rose Essentials: Using the Booch Method. Benjamin-Cummings Publishing Company; Har/Dis edition (July 1994).

21. Kaikkonen, A., Kallio, T., Kekäläinen, A., Kankainen, A., Cankar, E. Usability Testing of Mobile Applications: A Comparison between Laboratory and Field Testing. Journal of Usability Studies, Issue 1, Vol. 1, pp. 4-16 (November 2005).

22. Dunlop, M. and Brewster, S. The Challenge of Mobile Devices for Human Computer Interaction. Personal and Ubiquitous Computing. Volume 6, Number 4, pp. 235-236 (September, 2002).

23. Korpipää, P. and Mäntyjärvi, J. An ontology for mobile device sensor-based context awareness', Proceedings of CONTEXT, 2003, Vol. 2680 of Lecture Notes in Computer Science, pp.451–458 (2003).

24. Elmasri, R., Navathe, S. Fundamentals of Database Systems (4th Edition). Addison Wesley (2003).

25. National Oceanic and Atmospheric Administration. National Weather Service digital weather forecast data. The service's WSDL description is located at http://www.nws.noaa.gov/forecasts/xml/DWMLgen/wsdl/ndfdXML.wsdl.

26. Tao Gu, X., Hang Wang, X., Pung, H. K., Zhang, D. Q. An Ontology-based Context Model in Intelligent Environments. Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference, pp. 270-275 (2004).

27. Korpipää, P., Mäntyjärvi, J., Kela, J., Keränen, H., Malm, E. J. Managing Context Information in Mobile Devices. IEEE Pervasive Computing, vol. 2, no. 3, pp. 42-51. (July-Sept. 2003).

28. Gandon, F. and Sadeh, N. Semantic Web Technologies to Reconcile Privacy and Context Awareness. Web Semantics Journal. Vol. 1, No. 3, (2004).

29. Gandon, F. and Sadeh, N. A Semantic eWallet to Reconcile Privacy and Context Awareness. Second International Semantic Web Conference, Florida (October 2003).

30. Sadeh, N. m-Commerce: Technologies, Services and Business Models. Wiley (2002).

31. Chou, S. C., Hsieh, W.T., Gandon, F. and Sadeh, N. Semantic Web Technologies for Context-Aware Museum Tour Guide Applications, Proceedings of the 2005 International Workshop on Web and Mobile Information Systems (2005).

32. M. Sheshagiri, N. Sadeh and F. Gandon, Using Semantic Web Services for Context-Aware Mobile Applications MobiSys 2004, Workshop on Context Awareness, Boston, June 2004.

33. Miller, N., Judd, G., Hengartner, U., Gandon, F., Steenkiste, P. Meng, I. H, Feng M. W and Sadeh, N. Context-Aware Computing Using a Shared Contextual Information Service. Pervasive 2004, "Hot Spots", Vienna (April 2004).

34. Koukia, S., Rigou, M., Sirmakessis, S. The Role of Context in m-Commerce and the Personalization Dimension. Proceedings of the 2006 IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology, pp. 267-276 (2006).

35. Kurkovsky, S., Zanev, V., Kurkovsky, A. SMMART: using context-awareness in m-commerce. Proceedings of the 7th international conference on Human computer interaction with mobile devices & services (MobileHCI '05), pp. 383-384 (2005).

36. Bulander, R., Decker, M., Schiefer, G., Kolmel, B. Comparison of Different Approaches for Mobile Advertising. Proceedings of the 2005 Second IEEE International Workshop in Mobile Commerce and Services, pp. 174-182 (2005).

37. Looney, C., Jessup, L., Valacich, J. Emerging Business Models for Mobile Brokerage Services. Communications of the ACM, vol. 47, No. 6, pp. 71-77 (June 2004).

38. Kurkovsky, S., Harihar, K. Using ubiquitous computing in interactive mobile marketing. Personal an Ubiquitous Computing, vol. 10, issue 4, pp. 227-240 (March 2006).

39. Brown, R., Ryu,H., Parsons,D. Mobile helper for university students: a design for a mobile learning environment. Proceedings of the 18th Australia conference on Computer-Human Interaction: Design Activities, Artefacts and Environments, pp. 297-300, 2006.

40. Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., Giannopoulou, E. Ontology Visualization Methods - A Survey. ACM Computing Surveys, Volume 39 , Issue 4, 2007.