IADIS International Journal on Computer Science and Information Systems
Vol. 4, No. 1, pp. 1-20
ISSN: 1646-3692

Logic-Based Information Modelling Rules for Supporting Interoperability in Pervasive Services Management Systems[1]

Serrano, J. Martín; Serrat, Joan
Universitat Politècnica de Catalunya - UPC. Barcelona. Spain.

Abstract

Next generation services and networks require information and communications systems able to support context-awareness applications and especially pervasive services. This paper present research challenges in context-awareness and context information modelling rules for supporting management operations within pervasive services into the framework of information systems interoperability. Logic-based rules for context information are discussed and studied and then the system architecture for context handling and delivery of context information using logic models is presented. Logic-based information modelling techniques with semantic enrichment are presented as one of the most proper for pervasive applications in the framework of distributed context information handling and delivery for data query-oriented applications. We support the idea of end user‘s applications and the management complexity of such services as well as communications networks, following the states streaked by logic models.

Keywords: Context Information, Information Modelling, Information Systems, Information Systems Interoperability, Business-Oriented Solutions, Next Generation Services and Networks, Pervasive Services, Service Management Systems.

1. Introduction

Interoperability is a critical issue from the perspective of business-oriented solutions that need to access information from multiple information and multiple technology systems. Cross-layer information interoperability involves both fields the service capabilities of the middleware environment for supporting information and the representation capabilities and nature of the information. Middleware capabilities influence the performance of the information systems, their impact on the design of new services, and the adaptation of existing applications needs to represent and disseminate the information. Modeling information development impacts the design of syntax and semantic tools for achieving the interoperability necessary in next generation services. Interoperability is the ability for interpret information, process and exchange it between systems, inclusive if the systems are in different abstraction plane using different technological support, thus information interoperability is the data free-exchange that

[1] This research activity is partially funded by Ministerio de Educación y Ciencia under the project number TSI2005-06413.

provides cooperating business with the ability to bridge abstraction plane conflicts arising from differences in syntax, perspectives, meanings and assumptions.

This vision creates a compatible information environment based on the agreed deals between different systems and requires some degree of cooperation between systems, to enable data exchange and correct interpretations. Thus in mostly cases the way to reach interoperability is generating information models with its respective interfaces, schemas and formats for information exchange, making their semantic as explicit as possible, so that they can be easily handled and understood by the business-oriented systems and platforms with the objective that services are more pervasive.

When in this paper, it is making reference to pervasive services, it is describing services available to any user at anywhere, anytime [1][2] and from this definition the global nature of context-awareness service perspective is implicit. Context-awareness refers to the properties that make a system able to detect, interpret and react to its user's state and the network environment [3]. These properties provide the systems the ability to adapt its behaviour according to desired business rules, all the while offering interoperable and scalable personalised services.

However the descriptions and rules that coordinate the management operations of a system are neither the same that govern the data present in each information system. Information present in end user applications few times are being using for controlling the operation of a service and even more never the managing of the services. In this paper are presented the research challenges in context-awareness and context information modelling to describe the rules supporting management operations of context-aware services with the objective of providing to the information systems the interoperability necessary in next generation networks, under the perspective of the next generation services demands. Aiming to assess the principal objective of this paper, research efforts are done in the sense of clarifying the modelling information rules necessaries to create solids middleware applications for providing to the information the interoperability necessary that pervasive services demand.

The work presented in this paper also contributes with middleware field presenting the system architecture for context handling and delivery using logic-based information models into the framework of a distributed context handling and delivery information system representation. The information is handled at the service-level, schema-level and the data-level. The solution doesn't require any changes to existing information systems in order to preserve data integration and interoperability, and at the same time and as the interoperable added value of this contribution, the architecture aims to be a standard functional middleware for management operations and handling information using business-oriented interfaces.

The structure of this paper is as follows. In Section 2 a review of the most compelling contributions on information interoperability is presented to end with a comparison of different information service requirements from the perspective of their contribution to accomplish the interoperability requirements in the information system. Section 3 presents the concerning logic-based information modelling rules for promoting the information interoperability in management systems supporting pervasive services from the vision of semantic enrichment to achieve this goal. Section 4 presents the system architecture for handling context information based on logic-based information rules. Section 5 presents compelling research work towards information systems interoperability. Finally Section 6 presents the conclusions and the future research lines.

2. Logic-Based Modelling: Basis & Principles

This section describes the envisaged requirements for modelling information interoperable models. The key question to understand this section is: what benefits can be obtained for each of logic-based information model approaches? and then relate the approaches with information systems requirements for supporting pervasive services.

With the introduction of context information and the service's requirements, the effective functionality of pervasive services in a vendor and technology neutral format can be guarantee. This section is the result of experiences in IST EU research projects related with pervasive computing, context information research advances and also the critical study of the bibliography that defines the state of the art on information systems interoperability into the pervasive computing and knowledge engineering area.

Pervasive systems make high demands on the context information that can be expressed in terms of a variety of properties which must be collected between organizations, technologies and community for diverse effects, applications and activities of the different operators, providers and end users involved in a service. In this paper those properties are defined as requirements for pervasive services. Let us suppose that the context information from an end user can help in the control operations of the service life cycle, what benefits from this activity are coming out? This information can then be shared by multiples communities but, is that possible the same scenario for different organizations and even more different technology operators and/or providers? Figure 1 illustrates the problem since an organizational scope. The information is transparently shared to the technology as result in intervene of certain organizations.

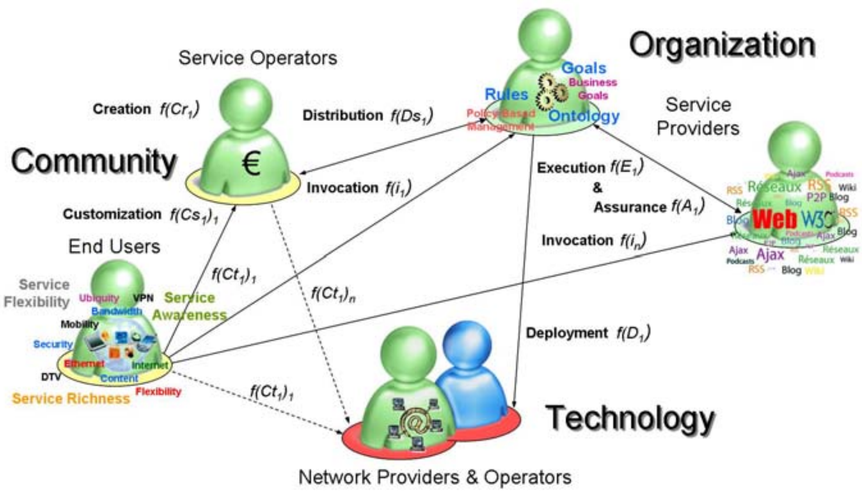

Figure **1**. Organizational view on the Interactions Information Interoperability.

This paper makes reference to the properties of the information described formerly in [4] and relates them with the current pervasive computing demands, underlying the interoperability in management systems for supporting pervasive applications and services. This paper makes reference to the context-aware services as pervasive services due to the mobility requirements that they have from context information inherently.

2.1 Requirements of Pervasive Services: Analysis & Outlook

We have studied the information's requirements for pervasive services proposed in [4][5] and added complementary requirements since the scope for pervasive services perspective to satisfy the services management requirements. A classification into two groups has been necessary according to pervasive requirements, the first group related to the demands that the systems make of the context information itself for providing information interoperability (nature-associated), and the second group associated with advantages that using context information are offered to the management systems for supporting pervasive services (operations-associated). Table 1 shown this categorization and its detailed explanation is included in subsequent sections.

We reference to previous research in information interoperability described in [6] and relate them to the current demands of services and their interoperability. Previous research in information interoperability can be divided into three sub-categories: mapping-based, intermediary-based, and query-oriented approaches. The mapping-based approach "attempts to construct mappings between semantically related information sources. It is usually accomplished by constructing a generalised (or global) schema and by establishing mappings between the generalised (or global) schema and the participating specific or local schemas". The intermediary-based approach "depends on the use of intermediary mechanisms (e.g., mediators, agents, ontologies, etc.) to achieve interoperability" and the query-oriented approach, "is based on interoperable languages, most of which are either declarative logic-based languages".

The high demands on the context information that current pervasive systems make can be expressed in terms of a variety of properties that we have compiled as information requirements and categorized in three levels, service, schema and data. This section study and integrates, with novelty and specific vision (interoperability information requirements) the current demands from the pervasive services.

Table 1 shows the requirements for pervasive services classified into two groups: nature-associated and operations-associated to identify clearly the requirements that information systems need to satisfy for providing and supporting information interoperability. Nature-associated attributes refers to the capability to handle and create new pieces of information from distributed sources with a certain level of quality, avoiding introducing ambiguity in the content of the information and ensuring that the information can be validated. Operations-Associated attributes are related to the scalability and automation of the pervasive systems using interoperable information. The envisage results are increasing the flexibility as well service as the information, reducing the management operations. It is also advantages of information interoperability in pervasive systems.

The schema-level contains the information requirements about type definitions (naming), a part of these types is fixed, since they may not be modified by process enactment. This part corresponds to the definition of process language constructs, which are common to all process models. The other part describes the templates of the objects, manipulated in a specific process model. This second set of requirements may be modified as a consequence of changes to the process model. Finally the schematic discrepancies attributes corresponds to the definition of relationships constructions to control the models processing. The schema-level requirements must be met to ensure that information operability work efficient and correctly: Each generated table needs to have primary keys defined with clear identifiers that allows the scheme isomorphism and can be established relationships between entities and additionally, key constraints can be defined to manage schematic discrepancies efficiently.

The data-level contains the information requirements that need to be characterized at a scale suitable for a single semantic unit into a language. In other words data-levels are based on the "language of processing" with the level of data being considered as raw or origin. Data level has minimum requirements and data level is not increased until all those requirements have been met. The value, the representation, the unit and the precision need to be specified and then the reliability and spatial domain are defined in consequence.

The requirements dictated by pervasive services in terms of the information have been referenced to the different types of models that have been proposed for the information interoperability in the described and referenced approaches [6][7]. Known the information requirements dictated by pervasive services and the different types of models that have been proposed so far for information interoperability, it is wise to try to establish a classification of such approaches in terms of how much they support or contribute to satisfy the service requirements.

Table 1. Information Interoperability Approaches vs. Information Requirments - Evaluation Table.

Information Requirement Type \ Information Interoperability Approach			Mapping-Based*	Intermediary-Based*	Query-Oriented*
Service-Level	Nature-Associated	Composition	Y	Y	Y
		Validation	Y	Y	Z
		Quality	X	Z	Y
		Ambiguity	X	Z	Z
		Formality	X	Y	Y
	Operations-Associated	Extensibility	Y	Y	Y
		Scalability	X	Z	Z
		Automation	X	Z	Z
		Flexibility	X	Y	Z
		Management	Y	Y	Y
		Idependence	X	Z	Y
		Integration	X	Y	Y
Schema-Level	Entity Identifier		Y	Y	Y
	Schema Isomorphism		Y	X	Y
	Generalization		Y	X	Z
	Aggregation		Y	X	Z
	Naming	Attribute Synonyms	Y	Y	Y
		Attribute Homonyms	Y	Y	Y
		Entity Synonyms	Y	Y	Y
		Entity Homonyms	Y	Y	Y
	Schematic Discrepancies	Data Value-Attribute	Y	X	Y
		Attribute-Entity	Y	X	Y
		Entity-Data Value	Y	X	Y
Data-Level	Value		X	Y	X
	Representation		Y	Y	X
	Unit		X	Y	X
	Precision	Granularity	Z	Y	X
		Spatial Resolution	Z	Z	X
	Data Value Reliability		X	Z	X
	Spatial Domain		X	Z	X

* See [4][6][7] for detailed description & examples. Representation: Z-Partial Support Y-Full Support X-No Support

Table 1 shows ranking regarding capability to support the requirements from pervasive services and relates with information interoperability approaches. In this paper the research concentrates in intermediary-based and query-oriented approaches as they seem to be most suitable approaches for support the pervasive requirements, a middleware solution combining both approaches seem as the best way to solve the problem in managing complexity and enhance the context information into the pervasive computing.

The best way to read and understand the table 1 is, for example, let us assume the fact that Query-Oriented approach is ranked “Y” in “Formality” requirement, it must be interpreted as: approaches in this category highly support such property for information interoperability purposes within pervasive services, and in the same line, then if Mapping-based approach is ranked “X” in the same requirement, “X” must be interpreted as: approaches in this category don’t support such property for information interoperability purposes within pervasive services. Finally, following with the same methodology, the rank “Z” for the “Intermediary-Based” approach in “Quality” requirement must be interpreted as: approaches in this category support partially such property for information interoperability purposes within pervasive services.

3. Logic-Based Rules for Supporting Interoperability

This section presents the concerning logic-based information modelling rules for promoting the information interoperability in the management systems supporting pervasive services. The management systems are coordinated by reasoning systems able to support the activities and actions as well re-actions as consequence of basic management operations independently of the reasoning engines or mechanisms. An extended mechanism used is the policy-based management which offer not only adaptation of the management systems to the variations in the rules else further more extensibility when new rules need to be implemented in format of policies. Other technique as Artificial Intelligence (AI) is being explored and from the point of view of the operations to reasoning finite state machines are good representations for the variations that rules have into a services management system.

A variation of rules is translated in semantic variations or as the introduction of new words, which must be understood by the systems in benefit and for providing new capabilities to the systems for processing information more efficiently. Management systems must be capable to support semantically variations of the information and be flexible enough to manage not only the current but also future aspects of services variations as result of context information. Sophisticated processing must be used so that systems using context information can react in real time.

3.1 Logic-Based Management Rules

An important aspect of policy-based service management, in which this paper is centred, is the deployment of services throughout the programmable elements. For instance, when a service is going to be deployed over any type of network, decisions that have to be taken in order to determine in which network elements the service is going to be installed and/or supported by such network element. This is most effectively done through the use of policies that map the user and his or her desired context to the capabilities of the set of networks that are going to support the service. Moreover, service invocation and execution can also be controlled by policies, which enable a flexible approach for customizing one or more service templates to multiple users. Furthermore, the maintenance of the code realizing the service, as well as the assurance of the service, can all be related using policies. A final reason for using policy management is that when some variations in the service are sensed by the system, one or more policies can define what actions need to be taken to solve the problem.

Figure 2 shows the use case representation for pervasive services where the common operations to manage the services, which are present in the pervasive scenarios, are described as process between two actors in the service lifecycle. The basic operations are highlighted as common operations present in the most of the pervasive services as result of the research activity and analysis of the scenarios.

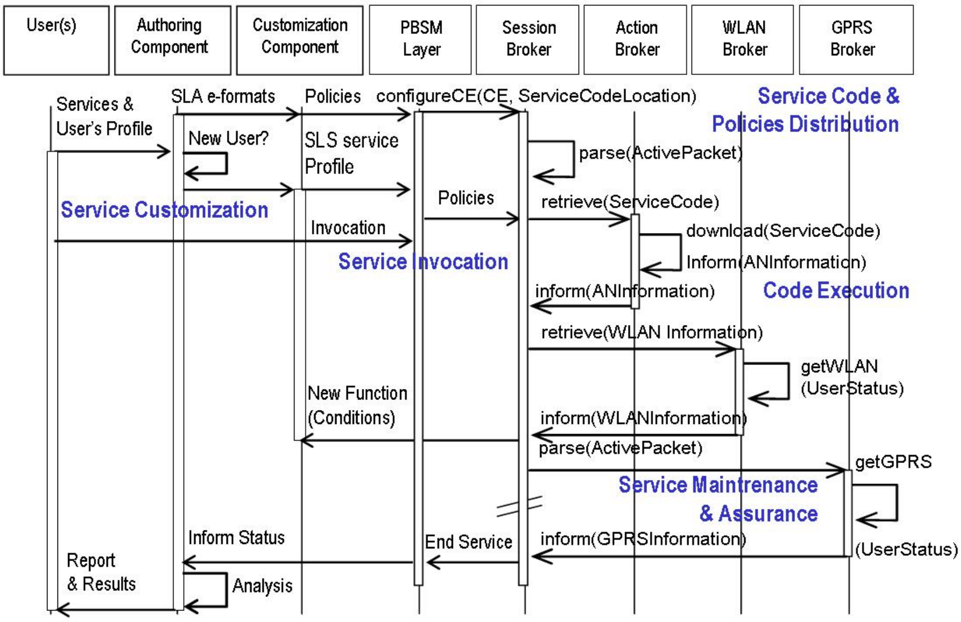

Figure **2**. Representation for Service Management Operations.

Use cases are an excellent way to define the interaction between stakeholders in the system and, as in this paper is shown, it can be used to simplify and better understanding of the activity in the scenarios. In this paper, experiences with both UML and use cases as well as other approaches to extract the meaning of the use case(s) and links, provide meaning to the system representations. Then it is wise to describe the basic management operations to support the lifecycle of pervasive services as here after is being done.

3.1.1 Service Code & Policies Distribution

This step takes place immediately after the service creation and customisation into the service lifecycle. It consists of storing the service code in specific storage points. Policies controlling this phase are CDistribution Policies. The mechanism controlling the code distribution will determine in which storage points the code is to be stored. The enforcement will be carried out by the components called typically Code Distribution Action Consumers. A high level example of this type of policies is presented as follow:

> "If (customized service event $f(Cs_1)_1$ is received = 1)
> then (distribute service code $f(Ds_1)$ in optimum storage points selection with parameters $f(Ds_n)$)"

Figure 3 represent how the context information as the event in a function $f(Ct_1)_1$ trigger the distribution of the code and the policies in functions $f(Ds_1)$ until $f(Ds_n)$ as result of context variations $f(Ct_n)_m$, where n is the context number to identify the type of context while m is the number of samples or variations of the same type of context information.

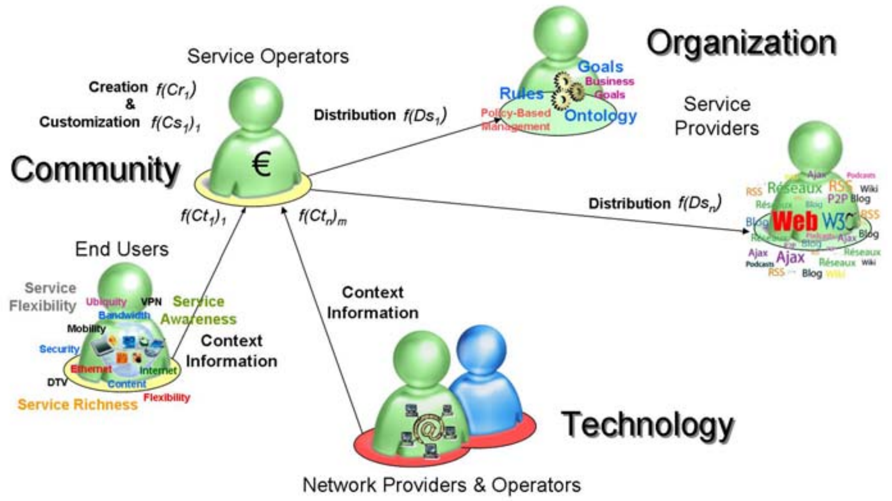

Figure **3**. Service Code & Policies Distribution Using Context.

3.1.2 Service Code Maintenance

Once the code is distributed, it must be maintained in order to support updates and new versions. For this task, we have the CMaintenance Policies. These policies control the maintenance activities carried out by the system on the code of specific services. A typical trigger for these policies could be the creation of a new code version or the usage of a service by the consumer. The actions include code removal, update and redistribution. These policies will be enforced by the component typically named Code Distribution Action Consumer. Three high level examples of this type of policies are shown here:

"If (new version of service code defined by $f(Ds_{1+n})$ = 1)
then (remove old code version of service $f(Ds_1)$) & (distribute new service code, function of $f(M_{1+n})$)"

"If (customized service code expiration date defined by $f(Ct_{1+n})_m$ has been reached)
then (deactivate execution for service $f(Ds_{1+n})$) & (remove code of service, in function of $f(M_{1+n})$)"

"If (The invocation's number for service is very high defined by $f(Ct_{1+n})_m$)
then (distribute more service code replicas $f(Ds_{1+n})$ to new Storage Points as function of $f(M_{1+n})$)"

Figure 4 represent how context information $f(Ct_n)_m$ control the maintenance of the code and the policies in functions $f(M_{1+n})$ as result of context variations $f(Ct_n)_m$, triggering the deployment of $f(Ds_1)$ until $f(Ds_{1+n})$ new services. in those functions n is the context number to identify the type of context while m is the number of samples or variations of the same type of context information.

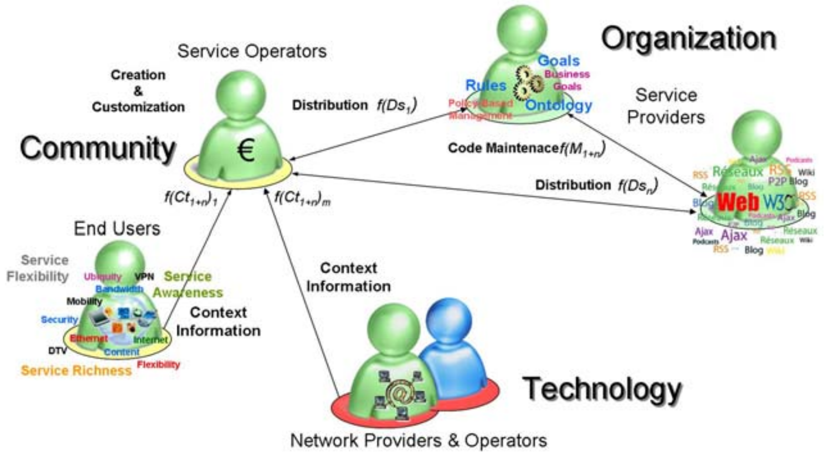

Figure **4**. Service Code Maintenance as Result of Context Variations.

3.1.3 Service Invocation

The service invocation is controlled by SInvocation Policies. The Service Invocation tasks are realized by components named Condition Evaluators which detect specific triggers produced by the service consumers. These triggers also contain the necessary information that policies are going to evaluate in order to determine the associated actions. These actions will consist of addressing a specific code repository and sending the code to specific execution environments in the network. The policy enforcement takes place in the Code Execution Controller Action Consumer. A high level example of this type of policies is presented:

"If (invocation event $f(Ct_1)_1$ is received)
then (customized service must be downloaded as function of $f(Ds_1)$ until $f(Ds_n)$ to IP addresses) "

Figure 5 represent the way in how context information $f(Ct_n)_n$ is directly the event as invocation in executing services as function of $f(I_{n+1})$. As result of context variations $f(Ct_n)_m$, new invocations result in code executions as function of $f(E_1)$ until $f(E_n)$ defining new services. In these functions n is the context number to identify the type of context while m is the number of samples or variations of the same type of context information.

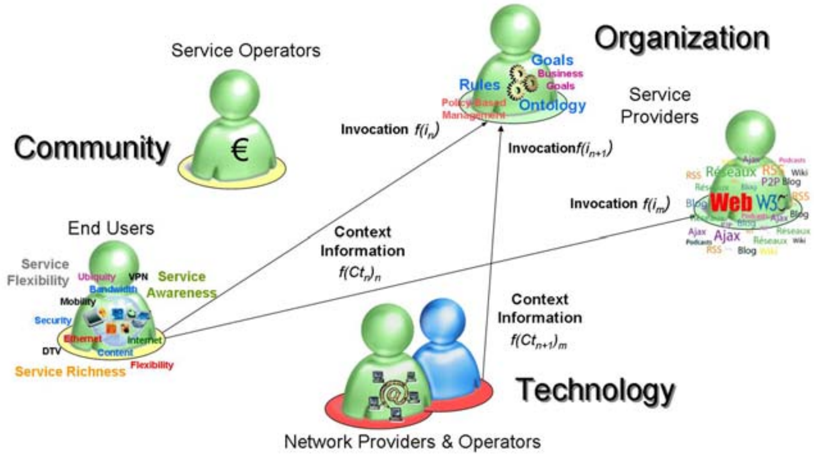

Figure **5**. Context Information Triggering the Service Invocation.

3.1.4 Service Code Execution

CExecution Policies will drive how the service code is executed. This means that the decision about where to execute the service code is based on one or more factors (e.g., using performance data monitored from different network nodes, or based on one or more context parameters, such as location or user identity). The typical components with the capability to execute these activities are commonly named Service Assurance Action Consumers, which are entrusted to evaluate network conditions. By other side the enforcement will be the responsibility of the components known, as typically are called, Code Execution Controller Action Consumers. A high level example of this type of policies is presented as follow:

"If (invocation event $f(I_{n+1})$ is received or $f(I_{n+1}) = 1$)
then (customised service must be deployed as function of $f(D_n)$"

Figure 6 represent the deployment of a service code and how the context information as part of the invocation function $f(I_n)$ is directly the event for executing services as function of $f(E_{n+1})$. Where n relates the context number identifying the type of context and m is the number of samples or variations of the same type of context information from the source, in this scenario the end user but not only limited. As result of context variations $f(Ct_n)_m$, new deployment result in code executions as function of $f(E_1)$ until $f(E_n)$ defining new services deployment as $f(D_n)$.

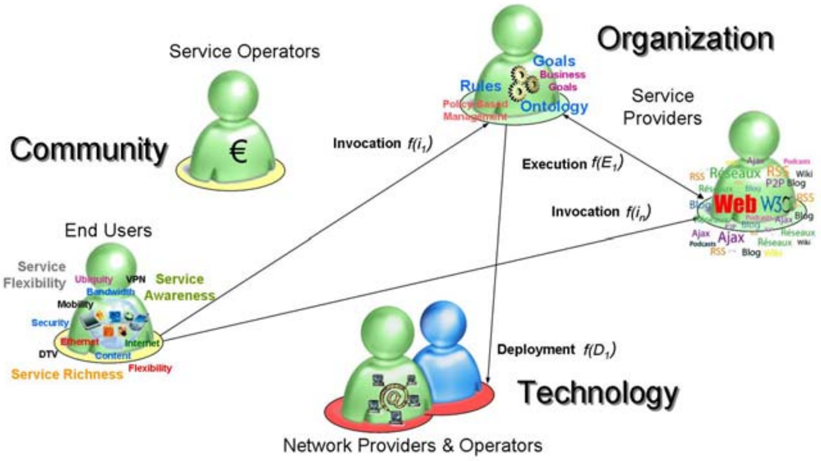

Figure **6**. Deployment of Services as Result of Service Invocation.

3.1.5 Service Assurance

This phase is under the control of SAssurance Policies, which are intended to specify the system behaviour under service quality violations. Rule conditions are evaluated by the Service Assurance Condition Evaluator. These policies include preventive or proactive actions, which will be enforced by the component typically called Service Assurance Action Consumer. Information consistency and completeness is guaranteed by a policy-driven system, which is assumed to reside service creation and customization framework. Examples of this type of policies are presented as follow:

"If (customised service is running as result of a deployment function $f(D_n)$)
then (configure assurance parameters $f(A_{1+n})$ service) & (configure assurance variables $f(D_{1+n})$)"

"If ($f(E_n)$=1)&(parameterA>X) then (Action defined by function $f(E_n)$)"
"If ($f(D_n)$=1)&(parameterB>Y) then (Action defined by function $f(D_n)$)"
"If ($f(A_n)$=1)&(parameterC<Z) then (ParameterA>X)&(Action defined by function $f(D_{1+n})$)"

Specifically in this phase, the externally provided information can either match pre-defined schema elements to achieve with certain management activities or, more importantly, the management systems can use these schema elements to extend and share the information to other management systems. The extension requires machine-based reasoning to determine the semantics and relationships between the new data and the previously modelled data. Reasoning and take decisions using ontologies is a new work, an overview as first approach of this complex task is contained in [8], and is beyond the scope of this paper.

Figure 7 represent the service assurance of the approach. It shows associations of context information $f(Ct_n)_m$ to $f(Ct_{1+n})_{1+m}$ which can be expressed using policy structures, conditions and actions with this information coming from the external environment for executing and controlling the management operations in a management systems scenarios for ensuring the service.

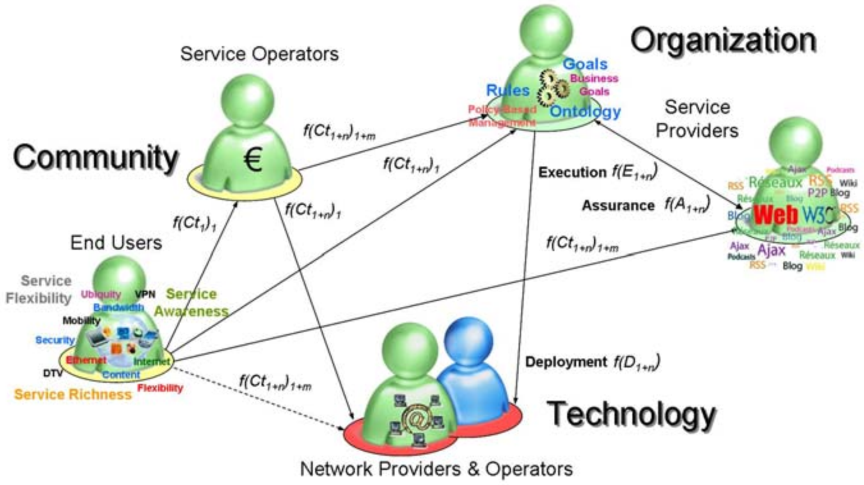

Figure 7. Context Information Controlling the Service Assurance.

The proposed service management operations do not assume a 'static' information model (i.e., a particular, well defined vocabulary that does not change) for expressing policies. In contrast, the service management operation proposal can process policies that can be defined dynamically (e.g., new variable classes can be defined at run-time creating new policies automatically). By supporting dynamically defined policies, the flexibility of policy-based management is demonstrated and achieved. Flexibility for create, deploy and execute policies is by itself a requirement at the time of designs the overall pervasive management systems with the objective for achieving rapid context-aware service introduction and automated provisioning.

3.2 Logic-Based Semantic Rules

This sections present research advances on semantic rules for combining service management operations and context information models for promoting the interoperability within pervasive applications. This section describes briefly a novel management technique using context information and ontology-based data models. We describe the use of Ontology-based management and modelling techniques within distributed and scalable frameworks, and outline representative ontology solutions for information management to support pervasive services using ontology-based models.

We propose the use of ontologies for support the integrated management covering service management operations such as the creation, customization, delivery and maintenance of pervasive services, and also for the integration of a user's context information in such service lifecycle management operations as described in the previous section. We aim to align this objective behind the DEN-ng approach [9], in order to make use of a standards-based approach for describing the management concepts and management information of interest for an information system.

The language used for expressing the ontology is OWL as the nature of pervasive applications is web-oriented. Additionally the semantic web have becomes recently in a guide at the time of design applications of next generation. One of the advantages of using OWL to express the ontology is to provide easy and fast tools for parsers, reasoners and text editors. OWL is capable to provide to the information model, and made aware of, constraints imposed by the information model over which it defines instances. OWL is then used to describe constraints over model elements that cannot be expressed with graphic notation..

The Table 2 shows a set of logic-based rules in a format of policies (Conditions-Actions) for managing the service life-cycle operations that need to be expressed in OWL. The objective of generate this Ontology is to create a logic-based and extensible context information models supported by formal languages like ontologies to provide information interoperability towards the semantic web and then support web services for exchanging information which is coming from business rules defined in the customization and creation processes. By other side we have chosen as ontology editor an open source solution as Protégé, a tool which construct ontologies, customize data entry forms, and enter data. It is also a platform which can easily be used to include graphical components (graphs, tables) and it use various storage formats such as OWL, RDF, XML, and HTML.

Table 2. Logic-Based rules for Managing Service Life-Cycle Policies.

Service Life-Cycle	Logic-Based Rules Examples

Customization	If (*serviceNew* $f(Cr_n)$=*"Service001"*) & (*LocatedIn* $f(Ct_n)_m$ =*"WebServer"*)
	Then (*CreateConfService001* $f(Cr_n)$=*"Service001"*)
	If (*userOf* $f(Ct_n)_m$=*"ConfService001"*) & (*LocatedIn* $f(Ct_n)_m$ =*"NetServer"*)
	Then (*CustConfServ001* $f(Cs_n)$=*"001"*) & (*CustConfServ002* $f(Cs_n)$=*"002"*)

Distribution & Deployment	If (*ConfServiceScheduleAt* $f(Ct_n)$= *"00:00:0000"*)
	Then (*DistConfServ001* $f(Ds_n)$=*"001"*) & (*DistConfServ002* $f(Ds_n)$=*"002"*)
	If (*userOf* $f(Ct_n)$=*"ConfService001"*) & (*locatedIn* $f(Ct_n)$=*"Reg001"*)
	Then (*StartConfService001* $f(D_n)$=*"ConfService001"*)

Execution & Maintenance	If (*userOf* $f(Ct_n)$=*"ConfService001"*) & (*LocatedIn* $f(Ct_n)$=*"Cell001"*)
	Then (*StartConfService002* $f(E_n)$=*"Service002"*)
	If (*ConfServiceDateAt* $f(Ct_n)$=*"00:00:0000"*)
	Then (*StopConfService001* $f(A_n)$=*"001"*)&(*StartConfService002* $f(A_n)$=*"002"*)

A novel characteristic when ontologies are introduced in service management operations and systems is the integration and harmony between context information awareness and policy based management. Ontologies aims to solve one of the main problems in the management of services and networks, which is integrating context information in tasks for managing networks service operations. In particular, Ontologies following logic-based rules seeks to detect context changes and change the functionality offered by the system in response to the new context. The synergy obtained between context-awareness, ontologies and policy-driven services promotes the definition of a new, extensible, and scalable knowledge platform for integration of context and services support in pervasive applications for support the next generation networks support.

Figure 8 shows policy object definitions in its simplest form using OWL. This is defined as a set of relationships from the concepts of the ontology; the free exchange of context information between policies enables events to be used to trigger actions.

```
<owl:Class rdf:ID="Events">
     <rdfs:subClassOf rdf:resource="#Policy"/>
</owl:Class>
<owl:Class rdf:ID="Condition">
     <rdfs:subClassOf rdf:resource="#Policy"/>
</owl:Class>
<owl:Class rdf:ID="Action">
     <rdfs:subClassOf rdf:resource="#Policy"/>
</owl:Class>
<owl:Class rdf:ID="ManagedEntity">
     <rdfs:subClassOf rdf:resource="#Policy"/>
</owl:Class>
<owl:Class rdf:ID="Obligation">
     <rdfs:subClassOf rdf:resource="#Policy"/>
</owl:Class>
<owl:Class rdf:ID="Authorization">
     <rdfs:subClassOf rdf:resource="#Policy"/>
</owl:Class>
```

Figure **8**. Policy Classes Definition in OWL – Simplest Form.

Figure 9 shows a representation example of the structural OWL grammar. It is a description of a object class and represent the simplest definition and relationships in sense of disjoints with other classes, this definition represent an object pertaining to a DomainContextEntity with disjoints of a set of classes as Application, Place, Person, DataBaseIM and Task.

```
<owl:Class rdf:ID="Object">
     <rdfs:subClassOf
           rdf:resource="#DomainContextEntity"/>
     <owl:disjointWith rdf:resource="#Application"/>
     <owl:disjointWith rdf:resource="#Place"/>
     <owl:disjointWith rdf:resource="#Person"/>
     <owl:disjointWith rdf:resource="#DataBaseIM"/>
     <owl:disjointWith rdf:resource="#Task"/>
     <owl:disjointWith
          rdf:resource="#DomainContextEntity"/>
     <rdfs:label xml:lang="en"
           >OSMCtxtInfoModel</rdfs:label>
</owl:Class>
```

Figure **9**. Object Class Description using OWL Representation.

Finally, one advantage more of using OWL is that once an OWL representation is available, it can be enriched with further semantics coming form shared ontologies; future work will be done to better characterization and ontology interactions. The semantic expressiveness provided by using formal languages such as OWL is not possible to achieve with *just* an object-oriented information models.

4. Logic-Based Context Handler System (L-HandS)

The Logic-Based Context Handler System (L-HandS) is an extension of previous research work using object-oriented paradigm for modelling and handling context, as detailed in [8]. This approach was enough good for satisfying some specific context-aware services management requirements but it lacks of the formalism necessary to support other data-oriented applications, i.e. date query-oriented applications, due to the content of classes of the objects is closed, then we address our research work towards demonstrate using logic-based models the information interoperability is increased as result of the synergy obtained from logic-based models and the information handler system proposed in this section.

A context dissemination and interoperable scheme within the L-HandS ensures the efficient and scalable distribution of context information among the different players and levels as result of the rule-based mechanism distributed among the networks supporting information systems. As Figure 10 depicted we assume overlay network with different Semantic Context Managers over different networks associated to logic-based rules controlling the context information, the service life cycle and the dissemination of the information.

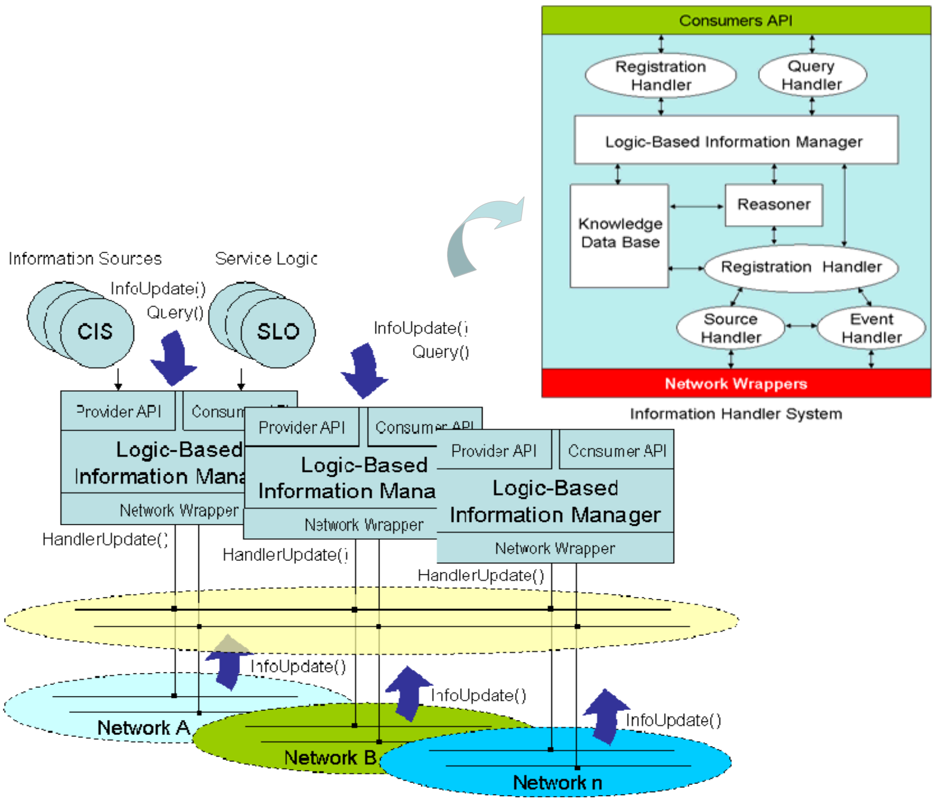

Figure 10. Logic-Based Context Handler System (*L-HandS*)

The L-HandS is oriented towards an efficient collection and dissemination of context information for information interoperability supporting pervasive services applications. Our proposal consists of context producers, context consumers and context brokers that coordinated by logic-based managers and semantic parsers deal with pervasive services requirements for context information pieces that control the service life cycle. Producers of context information in our system include all the Context Information Sources (CISs) that are attached or that can be part of specific network nodes for providing context information from mobile access technology wrappers (WLAN, GPRS), information from other wrappers (e.g. temperature or position sensors) or network performance or conditions (IP Network performance).

Context Consumers are the instances implementing the end user services following the service logic. During the operational phase of the service, the service logic issues requests for the acquisition of context information in order to carry out the actions for which it has been conceived as response of a logic-based operation defined and implemented by the logic-based components.

The L-HandS managers, namely Logic Information Managers (LIM), act as third-party players between producers and consumers and implement interfaces for interact with logic components for logic-based control and decisions. LIM's accept requests from the service logic elements to provide service life cycle actions and context information. LIM's collect context information from producers and logic-based disseminate the information from producers to consumers, or other LIM's dictated by a dissemination scheme that can be contained in a rule-based system. LIM's provide two different types of APIs to accomplish their task: the Producer APIs and the Consumer APIs. The Producer APIs include interfaces that enable context producers to publish the information that they provide, either raw or complex, so that the LIM's will be able to access it. In this sense, the heterogeneity of the different context sources and the information interoperability supported due to the logic-based model; which is the same that is used for managing the service logic and then dissemination of the context information is transparent to the service logic.

4.1 Logic-Based Context Handler Detailed Functionality

The functionality of this system can be understood as decomposed in more elementary functions as depicted in Figure 10 and described mostly in the previous research job [4][8], hereafter we describe only the extended components as result of our ongoing research activity for extend towards logic-based information models and support information interoperability and also control service lifecycle (service managements operations).

4.1.1 Logic-Based Information Manager - LIM

Logic-Based Information Manager is in-charged for managing the entire context information related with a specific query-oriented service and keeps it updated regarding the surrounding of the registration context handler. In contrast of previous centered applications that use extensive context data bases completely centralized, complex and sometimes difficult to access, this component will contain a data-base containing the terms from a general ontology and the logic sentences in order to relates concepts and find semantic equivalences following rule-based queries, semantic descriptions are out of the objective in this paper and then by now just to say that we use ontology descriptions for represent and contain the context knowledge.

The Figure 11 shows part of the OWL Ontology in XML representation, as example of the content of the ontology for the context information in pervasive applications. The management is performed by using of rule-policies adapting the behaviour of the context handler containing the conditions that the services will require and using this profile to construct the logic-based sentences that the component will use for handling service life-cycle operations.

```
<owl:Ontology
 rdf:about="http://nmg.upc.edu/ontologies/1.0/CTXTModel">
<rdfs:comment>
 An ontology for context-aware systems in pervasive
 environments.
</rdfs:comment>
<owl:versionInfo>1.0</owl:versionInfo>
</owl:Ontology>
<owl:Class rdf:ID="Place">
     <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#AtomicPlace"/>
          <owl:Class rdf:about="#CompoundPlace"/>
     </owl:unionOf>
     <rdfs:subClassOf>
         <owl:Restriction>
               <owl:onProperty rdf:resource="#object"/>
               <owl:object>1</owl:object>
          </owl:Restriction>
     </rdfs:subClassOf>
     <rdfs:subClassOf>
          <owl:Restriction>
               <owl:onProperty rdf:resource="#device"/>
               <owl:device>1</owl:device>
           </owl:Restriction>
     </rdfs:subClassOf>
</owl:Class>
```

Figure 11. Context Model Using Ontologies – Representation Example.

4.1.2 Knowledge Data-Base – KDB

The *Data Base* associated is in charge of stores the general or global ontology and the contextual information following the contextual information (logic-based) data model. The global ontology and the local ones (e.g. WordNet) are stored in this data base then a user's query is expressed in terms of the global ontology recalls and in order to be mapped with local information sources, the query has to be rewrote and expressed on the local ontologies concepts also contained in this data base.

4.1.3 Reasoner

The *Reasoner* is the engine that takes decisions during the ontology mapping between general ontology terms and local context terms. The enrichment of this component is supported in two ways, by one side the logic-based information model and by other an ontology language.

5. Related Works

We have supported, within this paper and based on functional component descriptions, the idea of information interoperability can be categorized into three main sub-categories, already referenced in section 2 (mapping-based, intermediary-based, and query-oriented approaches). Mappings are not limited to schema components and may be established between domains and schema components (i.e., classes, relationships, and attributes). Drawbacks of the mapping-based approaches [10][11][12] are that they are not designed to be independent from particular schemas and specific applications and as consequence those are restrictive.

The intermediary-based approach may have domain-specific knowledge, mapping knowledge, or rules specifically developed for coordinating various autonomous information sources. In most cases, such intermediaries use ontologies to share standardized vocabulary or protocols to communicate with each other. [13][14][15][16]. The advantage of using ontologies is its ability to capture the implicit knowledge within a certain domain in great detail in order to provide a rich conceptualization of data objects and their relationships. The knowledge within an ontology is domain-specific, but independent of particular schemas and applications, even though the approach may be theoretically valid, However it is practically infeasible to develop and maintain an ontology in autonomous, dynamic, and heterogeneous databases due to the inherent complexities of the knowledge domain.

The query-oriented approaches are capable of formulating queries spanning several databases [17][18][19]. One of the main drawbacks of this approach is that it places too heavy a burden on users by requiring them to understand each of the underlying local databases. This approach typically requires users to engage in detection and resolution of semantic conflicts, since it provides few support for identifying semantic conflicts.

6. Conclusions and Further Work

The information interoperability arranged approaches referred in this paper are not mutually exclusive. For example, the intermediary-based approach may not necessarily be achieved only through intermediaries. Approaches based on intermediaries also rely on mapping knowledge established between a common ontology and local schemas. It is also often the case that mapping and intermediaries are involved in query-oriented approaches. Based on this the approached on this paper had been developed following intermediary-based approach with ontologies and query-oriented solution with logic-based models.

We have addressed our research activity towards the alternatives for supporting the information interoperability in pervasive environments and also the definition, deployment and management of context-aware services and then we support the idea of using logic-based information models for achieving such interoperability.

We have proposed a solution suitable for any particular technology to handle and gather context information. Different “wrappers” and specific brokers would need to be developed to adapt Cisco Routers, Linux based machines, WLAN Access Points or SIP servers, from different vendors. In order to test the validity of the solution proposed with the logic-based information model we have simulated L-HandS system operations within the framework of Context project [20], part of descriptions corresponds to the solution.

ACKNOWLEDGMENT

This paper refers to ongoing research work in emerging IT technologies and information systems in UPC, which involves assessing of service requirements for information interoperability, support of pervasive services and information modelling techniques in next generation networks (NGN), ITU-T.

REFERENCES

[1] Henricksen. K., et al. "*Modelling Context Information in Pervasive Computing System*". Pervasive 2002, LNCS 2414, pp. 167-160, 2002.

[2] Serrano, J.Martín; Serrat, Joan; Galis, Alex. "*Ontology-Based Context Information Modelling for Managing Pervasive Applications*" 2006 IEEE/IARIA Int. Conf. on Autonomic and Autonomous Systems -ICAS'06; July 19-21, 2006, Silicon Valley, Santa Clara, Ca, USA.

[3] Dey, A. K., "*Understanding and using context*", Journal of Personal and Ubiquitous Computing, Volume 5(1), pp.4-7, 2001

[4] Strang, T., Linnhoff-Popien, C., "*A Context Modelling Survey*". Workshop on Advanced Context Modelling, Rasoning and Management Nottingham, England. September 2004.

[5] Strang, T. Service Interoperability in Ubiquitous Computing Environments. PhD thesis, Ludwig-Maximilians-University Munich, Oct. 2003.

[6] Jinsoo Park, Sudha Ram; "*Information systems interoperability: What lies beneath?*". ACM Transsactions on Information Systems 22-4,2004.

[7] Fileto, R., Bauzer, C. "*A Survey on Information Systems Interoperability*", Technical report – IC-03-030, Dec.2003.

[8] Serrano, J.Martín; Serrat, Joan; O'Sullivan, Declan; "*Onto-Context Manager Elements Supporting Autonomic Systems: Basis & Approach*". IEEE Workshop on Modelling Autonomic Communications Environments-MACE 2006, Oct. 23 - 27, 2006. Dublin, Ireland.

[9] Strassner, John; Stadler Rolf; Ulema Mehmet. "DEN-ng: Achieving business-driven network management". IEEE/IFIP Network Operations and Management symposium, Florence, Italy, 15-19 April 2002, pp. 753-766.

[10] Hayne,S. and Ram,S.*"Multi-user view integration system (MUVIS):An expert system for view integration"*. Proceedings of 6th Int. Conf. on Data Engineering, Los Angeles, CA, 1990Feb. 5–9). 402–409, 1990.

[11] Sheth,A.P., Galas,S. K., and Navathe,S.B., *"On automatic reasoning for schema integration"* Int. J. Intell. Coop. Inf. Syst. 2,1, 23–50. 1993.

[12] Collet,C., Huhns,M.N., and Shen,W.-M. *"Resource integration using a large knowledge base"* in Carnot.IEEE Comput. 24, 12, 55–62. 1991.

[13] Sciore, E., Siegel, M., and Rosenthal,A. *"Using semantic values to facilitate interoperability among heterogeneous information systems."* ACMTrans. Database Syst. 19,2, 254–290. 1994.

[14] Kuokka,D. and Harada,L. *"Integrating information via matchmaking"*. J. Intell. Inform. Syst. 6, 2/3, 261–279. 1996.

[15] Papakonstantinou,Y., Garcia-Molina, H., and Ullman,J *"Medmaker: A mediation system based on declarative specifications"*. Proc. of 12th IEEE Int. Conf. on Data Engineering, New Orleans, LA, Feb. 1996.

[16] Genesereth,M. R., Keller,A. M., and Duschka,O.M. "*Infomaster: An information integration system*". In Proc ACM SIGMOD Int. Conf. on Management of Data, Tucson, AZ, May 13–15, 39– 542. 1997.
[17] Czejdo,B., Rusinkiewics, M., and Embley,D.W. "*An approach to schema integration and query formulation in federated database systems*". In Proc. of 3rd Int. Conf. on Data Engineering, Los Angeles, CA, Feb. 3–5. 477–484, 1987.
[18] Krishnamurthy, R., Litwin,W., and Kent,W. "*Language features for interoperability of databases with schematic discrepancies*". In Proc. of ACM SIGMOD Int. Conf. on Management of Data Denver, CO, May 29–31, 40–49., 1991.
[19] Calmet,J., Debertin,D., Jekutsh,S., and Schu,J. "*An executable graphical representation of mediatory information systems*". In Proc. of the 12th Int. Conf. on Data Engineering, New Orleans, LA, Feb. 26–March 1, 124–131., 1996.
[20] IST-CONTEXT project, Active Creation, Delivery and Management of Context-Aware Services. http://context.upc.es