Services Management Using Context Information, Ontologies and the Policy-Based Management Paradigm: Towards Integrated Management in Autonomic Communications

Serrano, J. Martín[1]; Serrat, Joan[1]; Strassner, John[2]; Cox, Greg[2]; Carroll, Ray[3]; Ó Foghlú, Mícheál[3]
[1]Universitat Politècnica de Catalunya - UPC, Barcelona, Spain.
{jmserrano; serrat}@tsc.upc.edu
[2]Motorola Labs, Schaumburg, IL, USA
{john.strassner; greg.cox}@motorola.com
[3]Telecommunications, Systems and Software Group, Waterford Institute of Technology, Ireland
{mofoghlu; rcarroll}@tssg.org

Abstract – The management of pervasive services on next generation networks involves self-management capabilities for improving performance and achieving interoperability necessary to support current and next generation services. This paper introduces a case study on an ontology-based, policy-driven architecture for managing pervasive services using context information. The architecture employs ontologies and policy-based paradigms for the definition, customization, deployment and management of pervasive services. We present a shared ontology for capturing and then modifying the capabilities of the context information in the governed functionality and increase interoperability using a novel ontology called Ontology for Support and Management of pervasive services (OSM). We then use the OSM ontology in a pervasive system that highlights the importance of formal information models in policy-driven service architectures. Finally this paper describes some fundamental components for policy-driven management of pervasive services and presents a context information model following the OSM ontology. We contribute to the research challenges in middleware technologies for Integrated Management, setting the basis for service management architectures using context information.

1. Introduction

Pervasive services use heterogeneous devices and functionality, requiring a mix of technologies such that each one have their own management data and programming models. Most systems use **data models** (*not* information models), and hence each system generates management data that is specific to it. This precludes reusability and makes sharing management data very difficult. Hence, the way to achieve efficient and clear interaction between the systems is to create architectures with information models that support information exchange that is inherently reusable and shareable.

Context information involved in pervasive services is also getting more interoperable. Contextual information is key to driving the adaptation of services according to changing user and business needs. This means that pervasive services must respond to highly dynamic and adaptive computing environments. Hence, services must become more context-aware.

Pervasive computing is an interdisciplinary area in the research community. This is because pervasive computing requires multiple levels of abstraction (end user vs. network operators) that define and describe resources and services that are truly interoperable. Pervasive computing requires a mix of many technologies, spanning from networking devices up to many middleware and user applications and services. However, it is that mixture of technologies in conjunction with the context information awareness and policy-driven management that makes more interesting and useful the integrated management of operations in the applications, as Figure 1 shows.

Pervasive computing is maturing rapidly and, when combined with new wireless and information technologies, has become a great support in the mobility and ubiquitous computing areas, and then the context information for supporting service lifecycle operations under policy-driven rules and ontologies acquire more relevant towards the interoperability in autonomic communications.

We propose that the DEN-ng [1] information and data models be used, as they are inherently extensible and already have rich policy and context models and together with the Ontology for Support and Management of pervasive services - OSM, presented in this paper, support the way towards the integrated management in autonomic communications area.

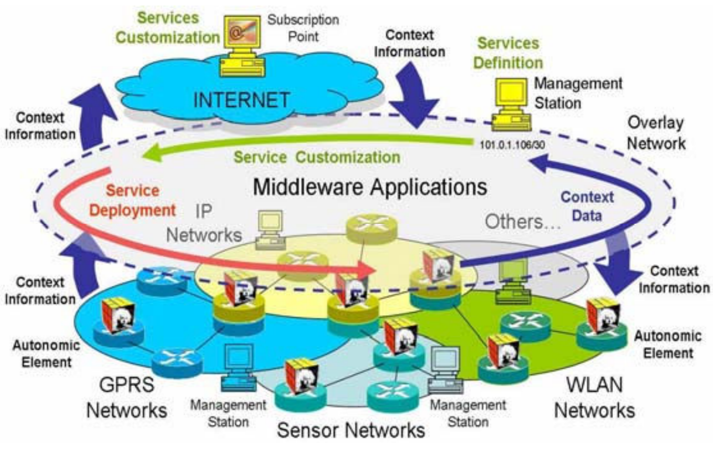

Figure 1. Context Information Role in Management Operations within Integrated Management Environments.

Section 2 of this paper presents the context modelling activity with ontologies for services management that use context information to control the lifecycle of pervasive services using policy management techniques. Section 3 describes our Onto-context middleware solution, which is a policy-based, context-aware architecture for managing services using context information. In this architecture, the context information modifies the interoperable capabilities of the applications and the services. Section 4 presents a brief review about pervasive applications and information models using ontologies. Section 5 presents conclusions and future research work lines.

2. Context Modelling, Ontologies and Policy-Based Management

Currently the context information modelling using ontologies for managing pervasive services relies on knowing what elements of context are actually relevant; this in turn drives the selection and use of the proper ontology for discovering the meaning(s) or helping with reasoning about the context information [2]. However, context information has not been represented or considered yet in management ontologies as a relevant part in managing services. Our activity addresses this, and we concentrate on an ontology-based, context-aware information model as part of service management, and relate this research activity to our onto-context middleware solution. The combination and integration of the context information as well as the use of policies and ontologies are part of the innovations described in this paper.

We believe that by creating ontologies for supporting and managing pervasive services using the properties and characteristics from the context information model, and managing pervasive services using the policy-based paradigm, we can better control the service lifecycle for managing those services operations.

2.1. Context Modelling

Pervasive systems use environmental information to enhance the interaction among people and improving the performance of their devices and systems. We classify these applications under the umbrella of Context-Aware Services (CAS). CAS takes into account relevant context information in order to determine which resources and services are offered [3] to perform the requested functionality (note: this could involve the generation of new services). The complexity of designing and deploying CAS is very high; hence, appropriate tools and infrastructures are required to acquire, share, and reuse information for CAS.

Data models, such XML-Schemas, specify the structure and integrity of data in the form of sets or containers of information. Current practice creates application- and technology-specific data models, which often creates management silos that inhibit reusability. The terms used in data models often constitute an informal agreement between the developers/programmers and the users of the data model [4]. However, unless a single common information model exists, there is no way to harmonize and integrate these diverse data models.

The most important challenge in pervasive services is the modeling and structure of context information [5][6]. The context model must be extensible and flexible enough to accommodate existing and future aspects of context information. The highly distributed nature of context information introduces a great challenge to consistent and coherent context information management. An additional challenge is the sharing and exchange of information between levels of abstraction. This is an area where the ontologies, functioning as a mechanism for specification of knowledge, can help and provide a solution.

We make use of the ontology-context information model described previously in [7]. This ontology-based model approximates how to structure, express and organize context information. The most important challenge is that the context information varies quickly, and it is naturally distributed across many layers in the systems. The model must be robust, semantically rich and flexible enough to be used in multiple platforms. In addition, the model must consider the current status of the managed object as well as future aspects of the context information describing the managed object. This also applies to other managed objects that affect the state of the object being managed. Most current applications are narrowly adapted to specific uses and don't provide sufficiently rich descriptiveness to support a generic context information model. We step through and describe, using ontology-based vocabularies for expressing concepts describing context information, concepts that are associated with service management operations. Figure 2 shows the ontology interactions between the classes of the information model and the context information model. Note that the structure merely suggests certain vocabularies related to each other for creating associations between concepts. For example, a "User" is a "Person" that *isDefinedAs* a "ContextEntity" and *isStoredAs* part of the "DataModel" with some other specific properties.

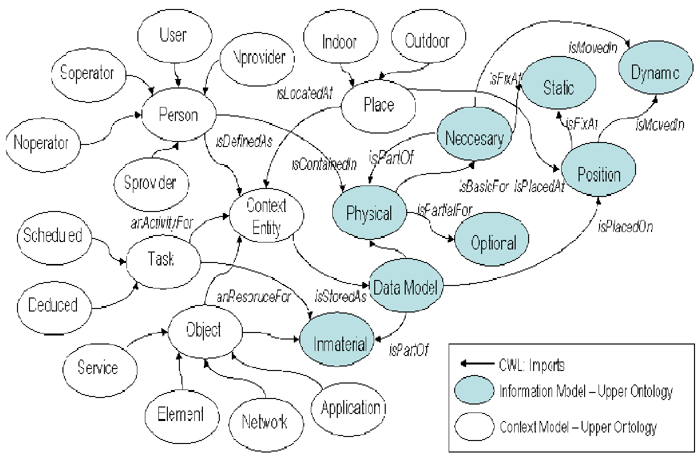

Figure 2. Information Modelling - Ontology Interactions.

2.2. Policy-Based Management

Several initiatives have used policy management approaches to tackle the problem of fast and customizable service delivery. These include OPES [8] and E-Services [9]. We go one step further and use the policy-based paradigm to control the full service life cycle. In addition, our approach takes into account the variation in context information and relates those variations to changes in the service operation and performance. All those managing concepts for service lifecycle operations are contained in the ontology for integrated management. The synergy obtained from context information modelling and the policy-based paradigm is the knowledge platform for creating the ontology. This synergy is an important aspect of our work.

Policy management is best expressed using a language. However, there are multiple constituencies involved (e.g., business people, architects, programmers, and technicians). Each of these have different backgrounds, and are best represented using different abstraction levels. This language and the interactions within different abstractions levels are shown in Figure 3. While most of these constituencies would like to use some form of restricted natural language, this desire becomes much more important for the business and end users. This notion was codified as the Policy Continuum in [1][10]. Our approach is based on producing a formal language (an ontology with appropriate dialects matched to each constituency) that is capable of knowledge sharing and integrated management. This language is to be standardized in the near future.

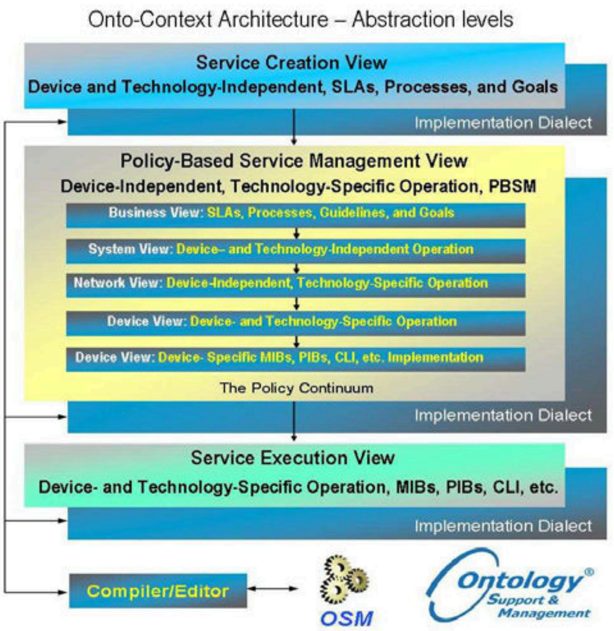

Figure 3. Policy Languages to the Policy Continuum - Interactions.

Our approach defines this set of languages using XML to ensure platform independence. XML is also easy to understand and manage, and the large variety of off-the-shelf tools and freely available software provides powerful and cost-effective editing and processing capabilities. Each of the implementation dialects shown in the Figure 3 is derived by successively removing vocabulary and grammar from the full policy language to make the dialect suitable for the appropriate level in the Policy Continuum. For some levels, vocabulary substitution using ontologies and inclusion of "noise words" are used to enable more intuitive GUIs to be built, but this and other linguistic aspects of our work are beyond the scope of this paper.

Policies are used to manage various aspects of the service lifecycle (e.g. authoring and customization). However additional activities are necessary to support the service lifecycle. The Figure 4 shows that these activities are also supported by policy management.

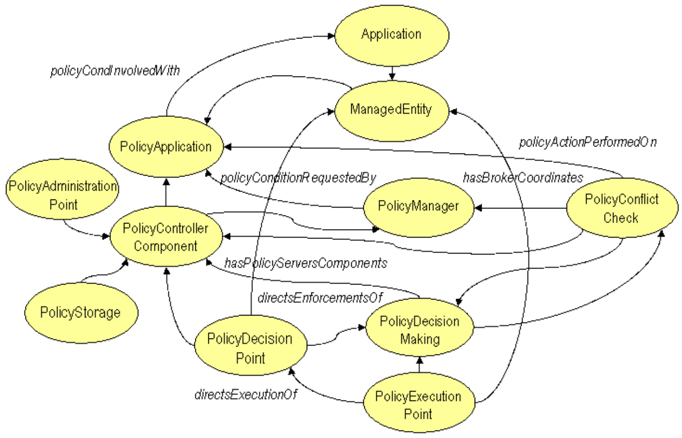

Figure 4. Policy-Based Management - Ontology Interactions.

An important aspect of policy-based service management is the deployment of services in multiple network elements. For instance, when a service will be deployed over any type of network, decisions must be taken in order to determine which network elements will support the service and thus have their configurations changed. This is most effectively done through the use of policies that map the user and his or her preferences to the capabilities of the set of network elements that will support the service. Moreover, policies can also control service invocation and execution, which enables 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 driven by policy management [1][11]. In addition, when the system senses undesirable service behavior, one or more policies can define what actions need to be taken to solve (or at least mitigate) the problem.

The main management operations that a policy-based system executes on a general managed entity are shown in Figure 4. The PolicyManager works to satisfy the *policyConditionsRequestedBy* PolicyApplications that are related to the ManagedEntity. The PolicyManager governs and coordinates management decisions and their enforcement (using various entities, such as PolicyExecutionPoints and PolicyDecisionPoints.

There are important approaches and projects dealing with Policy Based Service Management (PBSM) described in related art. [12] Provides a survey of policy specification approaches. The ANDROID Project [13] aims to prove the feasibility of providing a managed, scalable, programmable network infrastructure. [14] proposes a more service-layer-oriented approach to PBNM. The overall objective of the TEQUILA project [15] is to study, specify, implement and validate a set of service definition and traffic engineering tools to obtain quantitative end-to-end quality of service guarantees through careful planning, dimensioning and dynamic control of scalable and simple qualitative traffic management techniques within Internet services. Another interesting proposal is the CONTEXT project [16] which is the inspiration of the Onto-Context Architecture. The objective of CONTEXT is to provide a solution oriented for creating, deploying and managing context-aware services. This approach might be well suited for policy-based systems, but it was developed without considering the multiplicity of context information and the need for technology independence.

A generalized policy-based service management architecture for autonomic systems will articulate a functional process model and include process inter-relationships for an organization. Such an architecture will encompass a methodology that identifies the necessary policies, practices, procedures, guidelines, standards, conventions, and rules needed to support the business and their process inter-relationship enabling the organization to govern the application of policy management mechanisms. Hence, application of Policy-Based paradigm in the service management area seems to be a feasible alternative for meeting next generation service management goals.

2.3. Ontology Engineering

It is well know that in current management applications, different data models are a reality. This is due to many reasons; perhaps the most important is because different management data requires different tools, since the use and manipulation of those data requires different functions. For example, the simple text-based functionality of LDAP (for directories) is not sufficient for more complex tasks that require (as an example) SQL. This is the *trap* that developers fall into when they only use a *data* model and do not use an *information model.* In contrast, we use a single common information model (DEN-ng) that can harmonize the information present in each of these different data models. Using a single information model prevents different data models from defining the same concept in conflicting ways. In addition, the use of a single common information model enables the reuse and exchange of management information. [1][11].

However, current information models do not have sufficient semantic capabilities and functionality for representing all the descriptions, definitions (and hence, the ability to choose the (one) correct definition that should be used) and relationships for promoting integrated management. We chose the DEN-ng information and data models because their inherent extensibility and *pattern-based architecture* provides the necessary control points for ontological information to semantically enhance the data that they represent. In particular, they lack the ability to describe the behavior and semantics required to ensure interoperability. We believe that by augmenting management information described in information and data models with ontologies [17][18], we can meet our goal of providing an extensible, reusable, common manageability layer that integrates the context information into the management operations. This then achieves our goal of managing the service lifecycle. These cognitive operations are shown in Figure **5**. The interaction shown is very complex and beyond the scope of this paper; for this paper, the use of ontologies [19] provides semantic augmentation, addressing the cited weaknesses of current information models, resulting in improved system management.

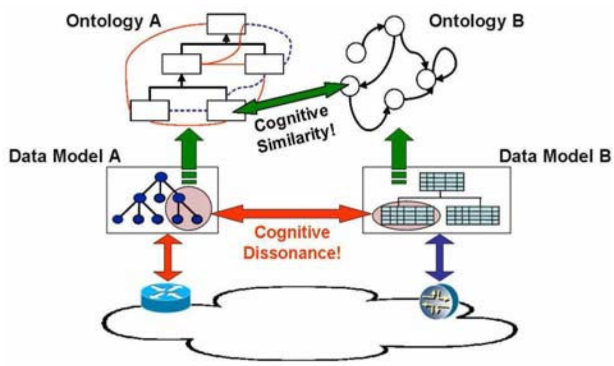

Figure 5. Information Model - Ontology Interactions.

We propose using a set of ontologies for managing the creation, delivery and reconfiguration, and execution also retirement of pervasive services. The ongology also allows for integration of user context information in service management operations (OSM). OSM is driven by a set of use cases on pervasive service management using policy-based architecture [16]. OSM is inspired from the necessity for integrated management in autonomic management [11]. The synergy obtained between context-awareness, ontologies and policy-driven services promotes the definition of a new, extensible, and scalable knowledge engineering platform for the integration of context information and services support in autonomic networking.

OSM defines a set of dialects that each use all or part of a formal lexicon defined in OWL [20]. This formal lexicon can be used to support the integration of context information in generic service management operations and policy-based systems, as well as to promote the integrated management necessary in autonomic communications.

Figure 6 shows the OSM Upper Ontology representation in terms of concepts and their relationships. Upper Ontologies are used to describe very general concepts, which are in turn used to provide high-level concepts that can link all root terms of all existing ontologies being used by the application. For example, change in the attributes of a ContextEntity can be detected using a PolicyEvent, which triggers the evaluation of a PolicyCondition. Any PolicyActions executed by this triggering affects one or more Managed Entities.

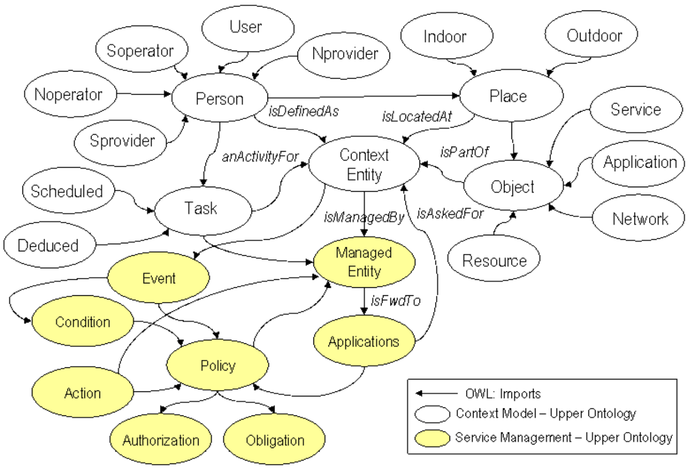

Figure 6. OSM Upper Ontology Representation.

Ontology Scope and Definition

We build upon and extend the entity model for modelling context in context-aware services from the EU IST-Context project [16], using ontologies as a formal mechanism to integrate both the context information defined in our information model as well as the policy-based services management system. The synergy obtained in this process results in our ontology for integration of context information. This ontology is extensible and interoperable, for services in autonomic networks, under the umbrella of the programmable, adaptive technology inherent in autonomic networks.

Class Hierarchy Definition

In business support systems, applications and services are usually organized into administrative domains. As a result of the dynamic nature of context, and following previous research work [6], we have identified that person, place, task and object (e.g., a device) are the most fundamental contextual data required for representing and capturing the notion of context in the ontology. Figure 6 shows the OSM upper- level ontology. The ontology is structured as a set of abstract classes describing a physical or virtual object in the service domain with attributes and relationships.

Ontology Capture

Currently, there has been an increased use of XML in different stages of knowledge capture, as well as in abstractions for formalizing the knowledge. It is in this context that we use XML to represent the context information model. Using XML has several advantages: a) XML is a mark-up language for documents containing structured information, but can also be used as a mechanism to exchange and store data; b) DTDs (Document Type Definitions) and XSDs (XML Schema Definitions) can be used to validate the documents created automatically when representing the context information; we have implemented a JAVA program for this as well as for creating and maintaining XML documents; c) the use of XQuery to find specific context information inside the XML documents that contain all the information related to a specific entity provides powerful searching capability.

We have chosen Protégé as our ontology editor, and defined a set of plug-ins for creating, editing, and customizing the content of our ontologies. It is also a platform which can be easily used to include graphical components (graphs, tables) and offer various storage formats, such as OWL, RDF, XML, and HTML.

Ontology Coding

Currently, the biggest ontology driver is the Semantic Web. Software tools are available to accomplish most aspects of ontology development. While ontology editors are useful during each step outlined above, other types of ontology building tools are also needed along the way.

Development projects often involve solutions using numerous ontologies from external sources as well as existing and newly developed in-house ontologies. Ontologies from any source may progress through a series of versions. In the end, careful management of this collection of heterogeneous ontologies becomes necessary to keep track of them. Tools also help to map and link between them, as well as compare, reconcile and validate them, merge them, and convert them into other forms. Other tools can help acquire, organize, and visualize the domain knowledge before and during the building of a formal ontology derived from or transformed into appropriate formats, such as W3C XML Schemas, database schemas, and UML, to achieve integration with associated enterprise applications that use these formats.

Figure 7 shows part of the links as definitions and properties of upper classes. Policy (short for PolicyRule) is part of a set of related policy classes for managing services. In the IST Context project, Policy is defined as an aggregation of Conditions and Actions, and we integrate ContextEntity as the Event for triggering management operations. Likewise the DEN-ng models consider event separately, a Policy Rule is defined as follows [1]:

> *A PolicyRule is an intelligent container. It contains data that define how the PolicyRule is used in a managed environment as well as a specification of behavior that dictates how the managed entities that it applies to will interact. The contained data is of four types: (1) data and metadata that define the semantics and behavior of the policy rule and the behavior that it imposes on the rest of the system, (2) a set of events that can be used to trigger the evaluation of the condition clause of a policy rule, (3) an aggregated set of policy conditions, and (4) an aggregated set of policy actions.*

Classes highlighted in blue in figure 7 are part of a domain-specific lower ontology (i.e., one that is used to define specific concepts and content). The upper ontology can contain and relate concepts defined in domain-specific lower ontologies to create new concepts that all applications can use. For example, City is part of Country, and this concept is defined by a ContextEntity as Outdoor Place in Country. In a service, when a user (modelled as a Person) is related to the service (e.g., the user is identified as a consumer of the service) and the user enters a City (i.e., Outdoor Place) that hosts the service, then the service can be offered to the user. This is an Event, which is part of a Policy Rule governing whether the user can use this service at this time.

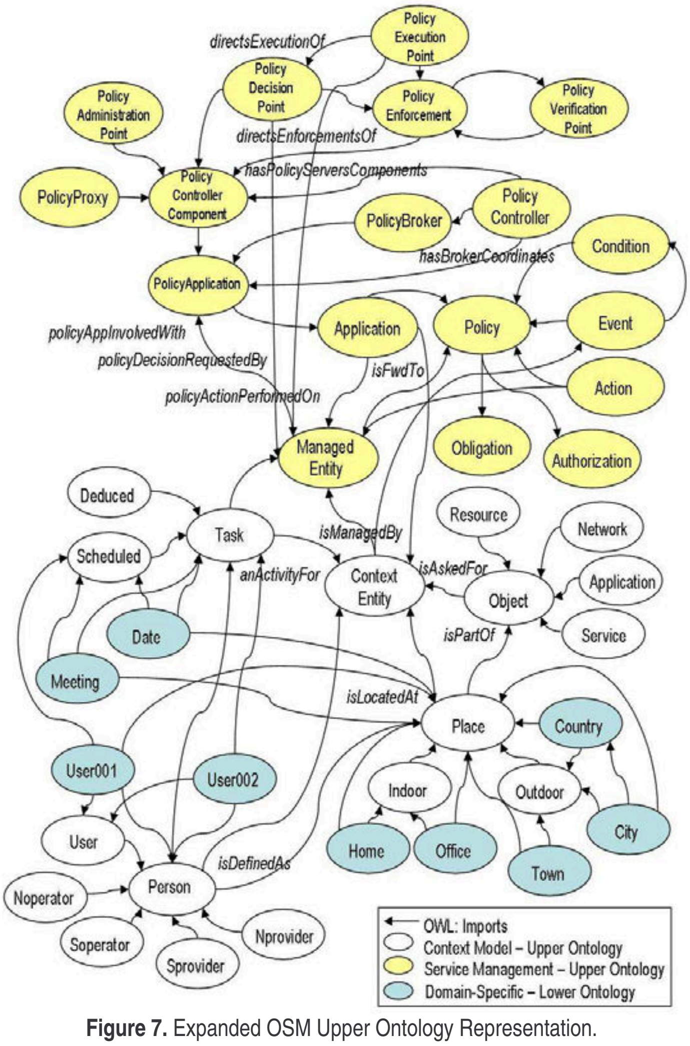

Figure 7. Expanded OSM Upper Ontology Representation.

The OSM ontology is built using the OWL Ontology Language. Figure 8 shows a part of the OWL Ontology represented in XML. This is an example of one of many formats supported by Protégé for representing, editing, and managing ontologies. The objective of integrating this work in OWL is to create an extensible, standardized, information model with ontological data that is usable by the semantic web for better supporting web services.

```
<!DOCTYPE owl [
        <!ENTITY OSMOntology "http://nmg.upc.edu/ontologies/5.0/OSMOntology#">
        <!ENTITY xsd "http://nmg.upc.edu/ontologies/5.0/XMLSchema#">
        <!ENTITY owl "http://nmg.upc.edu/ontologies/5.0/owl#"> ]>
<rdf:RDF
        xmlns:owl =http://nmg.upc.edu/ontologies/5.0/owl#
        xmlns:rdf ="http://nmg.upc.edu/ontologies/5.0/01-rdf-syntax-ns#"
        xmlns:rdfs=http://nmg.upc.edu/ontologies/5.0/rdf-schema#
        xmlns:xsd ="&xsd;"
        xmlns ="&OSMOntology;"
        xml:base ="http://nmg.upc.edu/ontologies/1.0/CONANModel" >
<owl:Ontology rdf:about="http://nmg.upc.edu/ontologies/5.0/OSMOntology">
<rdfs:comment>
An ontology for Integrated Management using context Integration in Pervasive Services Operations.
</rdfs:comment>
<owl:versionInfo>1.0</owl:versionInfo>
</owl:Ontology>
<!–
OSM Ontology: This ontology attempts to capture vocabularies and concepts that are
often used for integrated management covering service management operations such as creation,
delivery and management of pervasive services, and also for integration of user's context information
in service management operations.
Author: Martin Serrano
CVS Version: $Revision: 5.00 $, $Date: 2005/12/05 23:10:15 $
-->
<owl:Class rdf:ID="Person">
    <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#AtomicObject"/>
        <owl:Class rdf:about="#CompoundObject"/>
    </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>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#application"/>
            <owl:application>1</owl:application>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
```

Figure 8. OWL OSM Ontology Example.

Ontology Integration and Evaluation

Reuse of existing ontologies is a task that ontology development must anticipate. Indeed, it is this feature that will speed up the development of new extensible and powerful ontologies in the future. The integration of ontologies is an important task in the ontology development area; however, this issue is out of scope of this paper. We are considering this for future research.

The evaluation of an ontology involves the syntactic revision to resolve logical and semantic inconsistencies, as well as ensuring that all semantic relationships are present and make sense. Exposing the ontology to subject-matter experts in the area of concern, as well as in other related domains, is usually the best way to determine if the ontology is complete and extensible. We employ formal versioning control of the ontology to support this.

3. Onto-Context Architecture

The architecture presented in this section is a middleware solution for efficient management of services using context information in next generation networks. The motivation of our research activity was to extend this architecture to manage the active creation, delivery and maintenance of context-aware services [16] using ontology-based support. This architecture allows adaptability and dynamism to current and future pervasive services to the benefit of its users. The architecture is inspired by the IST-Context project [16] that was implemented and successfully evaluated with a set of service scenarios. However, remaining research challenges regarding information model extensibility and information dissemination exist. Our approach addresses these challenges, and helps us show the advantages and improvements for services management using an ontology-based information modelling methodology.

The Onto-Context architecture is a set of functional elements directly involved in the provisioning of context-aware services. The architecture, shown in Figure 9, includes a multiple interface ontology-toolkit that provides the highest level of abstraction to application developers (service providers) or to consumers themselves for supporting them in customizing the requested pervasive services using context.

The toolkit can be seen as providing high-level language constructors for communication service customization and instantiation on next generation networks. The new functionality of this architecture is based on the interaction and exchange of context information in an easy and efficient cross-layering way; this characteristic is one of our innovative contributions to enable Onto-Context architecture to be used with the multiple technologies and networks proposed. The context information model captures efficient, dynamic and proactive/reactive action based on user and network context information. We also demonstrate that ontologies are tools for improving context information models, and facilitate the dissemination and usage of context at any level of abstraction into communication services area.

The Onto-Manager manages the entire set of context information related to a specific service. This is accomplished in a distributed manner. The Onto-Manager updates context information that describes its environment. Thus, an Onto-Manager in charge of service customization will be responsible for updating all contextual user information, and an Onto-Manager governing network resources and services is responsible for updating all network layer information. Context information then will be available for use to trigger events that may affect one or more aspects of the system (e.g., the creation, deployment, and/or customization of the service).

The associated Database stores the ontology-based information following the data model concepts proposed in this paper. We have chosen Protégé as our ontology editor. Protégé is an open source solution that can construct and maintain ontologies through a set of tools (e.g., graphical and/or textual display of context data, queries, and data entry forms for entering and customizing content). It can use various storage formats, such as OWL, RDF, XML, and HTML [19]. The use of XQuery is advantageous for finding specific context information inside the XML documents that contain information related to a specific entity. These queries can select whole documents or sub-trees that match certain conditions.

We define the components that are used for representing and implementing the ontology-based information model (see Figure 9). These components begin the integration of the information model into the more general context-aware services architecture. A detailed description of this process is beyond the scope of this paper. However, we can say that the architecture is being developed and tested.

The enrichment of this proposal is founded on the use of the OWL ontology language. The content of the ontology for representing context information in pervasive applications has been edited with Protégé. The objective of generating this ontology in OWL is to create a perfectly extensible and interoperable ontology-based context information model. This model will be extensible to the semantic web for supporting to web services. Such extensions are left to future lines of research.

Increased use of XML to represent knowledge in different abstraction levels has driven us to propose XML to represent our Context Information Model. Figure 10 illustrates the Onto-Context architecture, highlighting the components that we consider necessary for implementing the onto-context system. The testbeds used in the IST-Context project can be reused for testing arch purposes.

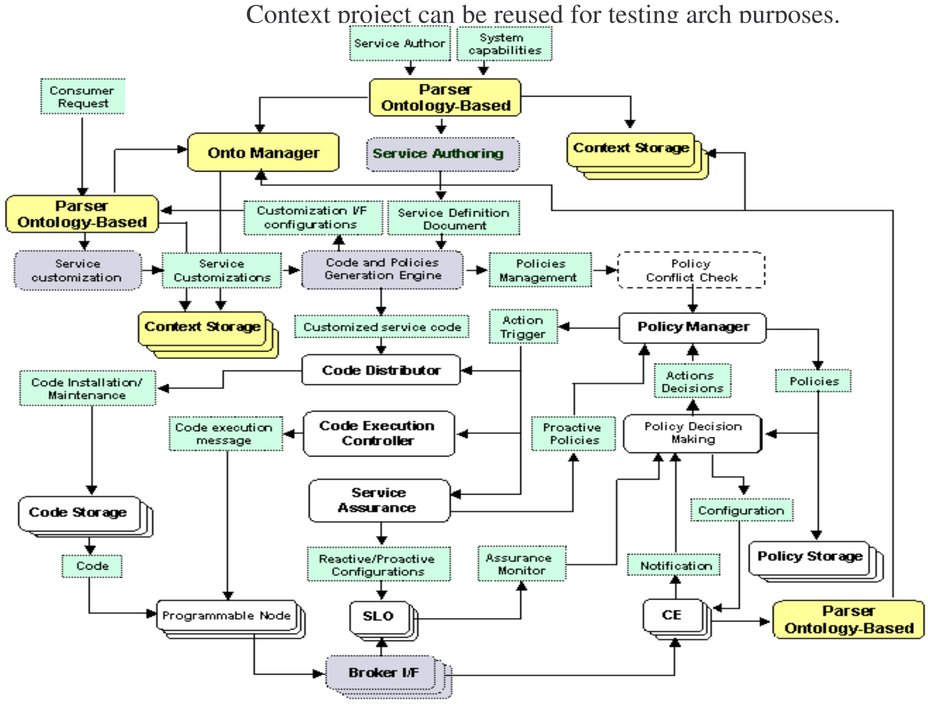

Figure 9. Ontology-Context Information Framework Detailed Elements.

3.1. Onto-Context Service Creation Layer - OSCL

The onto-context service creation layer (OSCL) deals with the modelling and complete technical definition of pervasive services using the semantic descriptions for the context information involved (CAS Authoring). OSCL aims to establish the required framework for the creation and customization of such pervasive services under the scope of web services technology, with XML as the language to represent the information (CAS Customization). The service code and management policies are created and distributed in XML format via SOAP for ensuring the deployment of the service (Onto-Policy-Based framework). Finally the produced code is made available to the OPBSM Layer by an Http server (Apache).

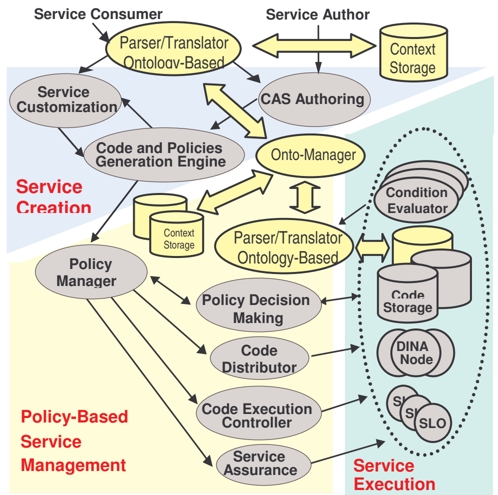

Figure 10: Onto-CONTEXT Architecture Functional Layers.

3.2. Onto-Context PBSM Layer - OPBSML

The onto-policy-based system management layer (OPBSM) planned in onto-context follows policy-based management philosophy to manage the services deployed and code generated (Policy Manager). As soon as a new service and Service Level Agreement (SLA) are received, they are downloaded to appropriate code repositories and/or storage points (Code Distributor). Context information is used to define a PolicyEvent as the trigger for evaluating whether the set of services specified in the SLA are ready to be created, deployed, or changed, as appropriate. Code is then generated and deployed in the service execution logic (Code Execution Controller); once the service is running to keep track of the appropriate metrics as defined by the SLA (e.g., quality of service), appropriate conditions are deployed (Service Assurance) to manage all processes of this layer. Semantic dissonances must be considered between the various forms and structure of context information from different sources, but this matter is out of scope for this paper (future research).

3.3. Onto-Context Service Execution Layer - OSEL

The onto-context service execution layer (OSEL) is supported by a mixture of programmable nodes and dedicated servers. This layer supports execution environments that offer a controlled way to host code for the purpose of running and managing the service. The motivation for selecting programmable technology has been the advantages that it offers as far as context collection, processing and storage is concerned.

We use the programmable platform based on concepts used in the ABLE & ABLE++ systems [21] and developed in the framework of DINA, as a programmable network middleware, which can be attached to different types of routers to turn them into programmable routers [21]. Furthermore, it contains APIs to allow the introduction of additional service components that help for ontology-based operations to the upper layer. It converts to Onto-Context in an extensible platform for supporting pervasive services.

4. Related Work

The broad quantity of applications using context information to provide new services [2][3][22] introduces a great challenge in the manageability of information in a consistent and coherent manner. There are research projects dealing with context-aware services and using context information models [23][24]. Nevertheless, each proposal defines an information model that is incompatible with other information models; such lack of inoperability is being solved mainly in the world of knowledge engineering through the use of Ontologies [25]. This topic has recently gained increased attention, as result of the increasing demand of the web-services that are supported principally by the Semantic Web [26].

In the field of ontologies for pervasive applications, we have studied proposals such as SOUPA [27], a standard ontology for ubiquitous and pervasive applications. SOUPA includes modular components with vocabularies to represent intelligent agents with associated beliefs, desires, and intentions. The Friend-Of-A-Friend (FOAF) Ontology [28] allows the expression of personal information and relationships, and it is a useful building block for creating information systems that support online communities. COBRA-ONT & MoGATU BDI ontologies are aimed for supporting knowledge representation and ontology reasoning in pervasive computing environments. While COBRA-ONT [29] focuses on modelling context in smart meeting rooms, MoGATU BDI [30] focuses on modelling the belief, desire and intention of human users and software agents. However, all these ontologies have been created to specify and describe the context information concepts as individual statements into a service, but never to be considered as information relevant for managing services.

5. Conclusions

The Ontology for Support and Management of services (OSM) is versatile and solves some of the problems in the area of pervasive services management.

Ontology Support and Management means that OSM is an ontology-based information model approach for pervasive applications that integrates its information model describing different aspects of context with ontologies that augment the model with semantic information; this augmented data enables context-aware service requirements to be modeled. Structuring this as a formal language using standardized XML provides extensible and efficient context information for managing services. *Ontology Service Management* means that OSM aims to cover the requirement to enable pervasive services to be automatically customized, delivered and managed. *Ontology for Services and Management* means that OSM is suitable not only for representing and defining context information concepts, but also for defining management operations that motivate and promote the integrated management of pervasive services.

We have presented an ontology-based, policy-driven architecture for managing services using the context information that provides support for customization, definition, deployment and management. Our efforts have been directed toward providing a novel, shared ontology for the integration of contextual information into service management operations and toward supporting a Policy-Driven, Ontology-based Architecture. The Onto-Context framework demonstrates how the use of an ontology-based information model can be used for managing context information and supporting pervasive services, thus improving cross-layer management and interoperability.

This paper contributes to middleware technologies for management, defining an extensible ontology ad robust support and management of pervasive services. Future work will continue the study of formal ontologies and their interaction with information and data models. One of the key areas that we will address is how to automatically validate (as much as possible) the relevance and correctness of both information and data models as well as appropriate ontologies in a context-aware system.

Acknowledgements

This research activity is co-funded by both the Ministerio de Educación y Ciencia under the project TSI2005-06413 and the Science Foundation Ireland (SFI) under the Autonomic Management of Communications Networks and Services programme (grant no. 04/IN3I404C).

References

[1] Strassner, J., “Policy Based Network Management”, Morgan Kaufman, ISBN 1-55860-859-1

[2] Gu, T., Wang, X.H., Pung, H.K., and Zhang, D.Q., An ontology based context model in intelligent environments, in Proc. of Communication Networks and Distributed Systems Modeling and Simulation. Conf. (CNDS’04), San Diego, CA, Jan. 2004.

[3] Henricksen. K., et al. “*Modelling Context Information in Pervasive Computing System*”. Proceedings of Pervasive 2002, LNCS 2414, pp. 167-160, 2002.

[4] Meersman R., (1999), “The Use of Lexicons and Other Computer-Linguistic Tools, Semantics, Design and Cooperation of Database Systems, in The International Symposium on Cooperative Database Systems for Advanced Applications” (CODAS 99), Springer Verlag, Heidelberg, pp. 1 – 14.

[5] Dey, A. K. and Abowd, Gregory K. “*Towards a Better Understanding of Context and Context-Awareness*” Netherlands. Georgia Institute of Technology, 2000.

[6] Gray, P. D.; Salber, D. *Modelling and Using Sensed Context Information in the Design of Interactive Applications*; Proceedings of 8th IFIP Working Conference on Engineering for Human-Computer Interaction 2001 Toronto, Canada.

[7] Serrano, J.M. Serrat, J. Galis, A. “Ontology-Based Context Information Modelling for Managing Pervasive Applications” Autonomic and Autonomous Systems, ICAS 2006, July 2006.

[8] G. Tomlinson, R. Chen, M. Hoffman, R. Penno, “A Model for Open Pluggable Edge Services”, draft-tomlinson-opes-model-00.txt, http://www.ietf-opes.org

[9] Giacomo Piccinelli, Cesare Stefanelli, Michal Morciniec. “Policy-based Management for E-Services Delivery” HP-OVUA 2001.

[10] TMF, “The Shared Information and Data Model – Common Business Entity Definitions: Policy”, GB922 Addendum July03.

[11] Strassner, J. and Kephart, J., *Autonomic Networks and Systems: Theory and Practice*, Network Operation Management Simposium. NOMS 06 Tutorial, April 2006.

[12] Damianou, N.; Bandara, A.; Sloman, M.; Lupu E. “A Survey of Policy Specification Approaches, http://www.doc.ic.ac.uk/~mss/MSSPubs.html

[13] ANDROID Project. http://www.cs.ucl.ac.uk/research/android

[14] Jun-Jang Jeng; Chang, H; Jen-Yao Chung; “A Policy Framework for Business Activity Management”. E-Commerce, IEEE International Conference. June 2003.

[15] IST-TEQUILA Project. http://www.ist-tequila.org

[16] IST-CONTEXT project. http://context.upc.es

[17] Gruber. T. R. “Toward principles for the design of ontologies used for knowledge sharing”. Presented at the Padua workshop on Formal Ontology, March 1993.

[18] Neches, Robert. Fikes, Richard. Finin, Tim. Patil, Ramesh. Senator, ted. Swartout, William R. “Enabling technology For Knowledge Sharing”. AI Magazine, Vol.12, (3), 1991.

[19] Gruber. T. R. “A translation approach to portable ontologies”. Knowledge Acquisition, 5(2):199-220, 1993.

[20] De Bruijn, Jos., Fensel, Dieter. Lara, Rubén .Polleres, Axel. “*OWL DL vs. OWL Flight: Conceptual Modeling and Reasoning for the Semantic Web*”; Nov. 2004

[21] Raz, and Y. Shavitt, "*An Active Network Approach for Efficient Network Management*", IWAN'99, July 1999, Berlin, Germany, LNCS 1653, pp. 220 –231.

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

[23] Abowd, Gregory D., Dey, Anid K., Orr, R., Brotherton, J. “*Context-Awareness in Wearable and Ubiquitous Computing*”. 1st International Symposium on Wearable Computers (1997)

[24] Hong, Jason I.; Landay, James A.: *An Infrastructure Approach to Context-Aware Computing*. In Human-Computer Interaction, 2001, Vol. 16, 2001

[25] Clark, D., Partridge, C., Ramming, J. C., Wroclawski, J. T. A Knowledge Plane for the Internet” SIGCOMM2003, Karlsruhe, Germany, 2003

[26] World Wide Web Consortium (3WC) http://www.w3.org

[27] Chen, H., Perich, F., Finin, T.W., and Joshi, A., SOUPA: standard ontology for ubiquitous and pervasive applications, in Proc. of the 1st Int. Conf. on Mobile and Ubiquitous Systems: Networking and Services. Boston, MA, August 22–26, 2004.

[28] D.Brickely and L Miller FOAF vocabulary specification. In RDFWeb Namespace Document. RDFWeb, xmlns.com, 2003.

[29] H. Chen, T. Finin and A. Joshi. “An Ontology for context-aware pervasive computing environments” Special issue on Ontologies for Distributed Systems, Knowledge Engineering review 2003.

[30] F. Perich. MoGATU BDI Ontology, University of Maryland, Baltimore County 2004.