# MODELLING KNOWLEDGE FOR DYNAMIC SERVICE DEPLOYMENT

## *Autonomic Networks Modelling*

Gladys Diaz

*L2TI, Institut Galilée, Université Paris 13, 99 Avenue J-B Clément, 93430 Villetaneuse, France*

Keywords: Information systems, Object-oriented Modelling, Autonomic Networks, Networks Management.

Abstract: This paper presents an overview of the modelling aspects in autonomic networks, especially concerning the knowledge level. We treat the specification and modelling of different notions in the context the dynamic deployment of services. We propose a new and extensible object-oriented information model to represent the various concepts involved in this context. Our information model enables us to represent the notions of service, resource and profile and also to deal with the management of networks resources. UML is used to represent our model. We also present an initial view of the knowledge that is needed to manage the resource allocations and distribution of services.

## 1 INTRODUCTION

Autonomic networks are currently an emergent field of research. The term "autonomic network" refers to the networks that have the capacity to be configured, and to self-manage according with the perception of their environment's evolution. Such a system requires a minimal administration, generally implying a management at the political level. To implement this configuration in an automatic way, it is necessary to have an absolute and permanent knowledge of the network environment, in order to apply the functions necessary to make it move dynamically. Thus, autonomic network involves a new level of management: the knowledge level. The knowledge in services deployment involves services and resource information, but also the events to be considered in the environment of deployment.

Management activities need to be tried with information defined at the knowledge level. In the context of network management, several services can be automated: equipment configuration, services deployment, QoS management, etc. To automate and reduce the complexity of information to be treated, it is essential to model the network environment by defining the different components implied and their interdependence. Modelling constitutes a first step to represent the level of knowledge about network environments.

In the context of information modelling, several standards have proposed information models for network management. OSI, ITU-TS and IETF have provided parts of the solutions to standardize the network information model. Currently, very important efforts in this area have been made by TMF, DMTF and 3GPP. These groups propose different models (CIM, SID) to represent management elements in different domains (applications, networks, devices, systems) (DMTF and TMF).

We will go into more detail about the knowledge modelling needed to allow for the dynamic deployment of services. This first section presents the problems encountered in our study and the organization of our proposal.

### 1.1 Problems to Study

The automation and the "dynamic" deployment of services constitute one of the final objectives of the suppliers of the services (Hass, 03). This dynamicity in the deployment requires an absolute knowledge of the deployment environment, that is to say, the network. To determine the optimal deployment strategies it is necessary to analyse the effectiveness of service placement decision in the network.

Dynamic deployment involves the capability of network to assure an automatic process to install the services requested by clients. This process involves


Diaz G. (2008).
MODELLING KNOWLEDGE FOR DYNAMIC SERVICE DEPLOYMENT - Autonomic Networks Modelling.
In *Proceedings of the Third International Conference on Software and Data Technologies - ISDM/ABF*, pages 154-159
DOI: 10.5220/0001885601540159
Copyright © SciTePress

the knowledge about the semantic of services, network elements, services components and the other necessary resources. Thus, to make a service deployment it is necessary make a configuration of each element involved in the deployment. This configuration involves the decision about what network nodes must be selected, in function of the knowledge about the available resources of these elements. The dynamic deployment is involved with two contexts of work:

- Components software deployment. In this context the problemn treated is the representation of service semantics and verification of service composition (Pistore, 04) (OMG, 03).
- Dynamic deployment of networked services. The major problem here is how to distribute the service requested and where we must place the dynamic service, to guarantee its correct functioning (Simoni, 99). We are, for the most part, interested in this second aspect.

### 1.2 Paper Organization

We propose an object-oriented model to represent the different notions in dynamic deployment. Our model enables us to represent the notion of service, and to also treat the management and control of dynamic deployment. The information model is represented by using UML diagrams. First we have given our own terminology. Then, we have given a description of the information model.

This paper is organized as follows. Section 2 presents the context of our work. We present in this section our motivation, and an overview of research in-progress on this subject. Section 3 presents our model to the service deployment. In this section we give the definitions of concepts introduced by our model, and also the different steps in the deployment of services according to our approach. Section 4 presents our conclusions and perspectives.

## 2 KNOWLEDGE LEVEL FOR AUTONOMIC NETWORKS

The knowledge of a complex system such as a network needs the representation of data and its context. Context is an important concept in the treatment and dissemination of knowledge information (Akhtar, 05). The context represents the set of events which must be evaluated in the cycle of life of any system to be able to make relevant decisions. Thus, data generated in a local component must be evaluated in order to provide for an appropriate global behaviour of that system.

Autonomic, or self*-, management of such a complex system will depend on a bottom-up process of knowledge formation, emerging from individual nodes in the system. Each component of the network represents an active element involved in the generation of relevant data (Gaïti, 05). This data must be evaluated in a global and coherent proposal to provide information which is complete and relevant for decision-making. In general, the knowledge information to be represented has to be:

- Dynamically generated and represented.
- Global-to-local, global objectives (high level goals) must be defined at upper levels.
- Global coherence, the coherence of decisions taken in accordance with global objectives, must be made in function of the evaluation of local ones. Low-level decisions are related to the high level goal which has been defined.

Service architectures are defined in the context of autonomic networks, especially in the context of Web Services. In these, the service level is the principal component of management and monitoring services execution in an automatic fashion (Papazoglou, 06).

Knowledge level in autonomic networks is a current subject of study (Clark, 03) (Lewis, 05) (Mulvenna, 05) (Seleznyov, 04) (Stojanovic, 04). The recent propositions present different viewpoints about this notion. The major challenge is composed of two questions: *How do we provide a high level view that defines the purpose of network?* and *How does we do the automatic acquisition and processing of knowledge?*

Several researchers have tried to answer these questions, and most of them agree on the fact that high level goals must be defined for network designers, and must be used to control the behaviour of applications running and user's purposes. Specific information (knowledge domain-specific data) could be used to represent these goals and be evaluated to take decisions, to recognize and mediate conflicts, to perform optimizations in the environment and to automate network functions.

Some works are interesting in the representation of knowledge information. (Wawrzoniak, 03) presents specific languages to represent information (model system knowledge) in a distributed approach like the grid-computer. (Stojanovic, 04) presents the notion and role of ontologies in autonomic computer systems. This work presents the results of IBM research in this context. Ontology represents an expressive conceptual modelling approach for describing the knowledge. (Lewis, 05) proposes an infrastructural service that enables the efficient delivery of network operations knowledge to nodes. The knowledge is represented here by using ontologies that enable one to service and to provide a level of semantic interoperability that will be transparent to the nodes providing and consuming knowledge. (Quirolgico1, 04) describes the current status of research in the modelling of ontologies.

Other researchers work with the representation of a knowledge plane, high level model to represent what the network is suppose to do, in order to provide services and advice to other elements of the network. In this case researchers are interested in the use of artificial intelligent (AI) tools and cognitive systems (Clark, 03). To treat the problem of distribution of knowledge, some approaches are using AI tools with autonomous agents to distribute knowledge management (Seleznyov, 04). (Eymann, 03) presents the utilization of software architectures (application-layers) to coordinate the provisioning of services and resources in distributed environments like the ones in Grids computers and Peer-to-peer (Chao, 04).

In general, agents and policy-based approaches are combined to give a complete infrastructure capable of treating the definition of goals and the distribution of operations. Ontology is used to organize the information and to structure its relationships.

In our context, we define dynamic deployment as the high-level goal. Each network component must work and take decisions, generating and manipulating information in order to enable the dynamic installation of a new service. To best understand the process of dynamic deployment we will detail the set of operations and data involved in the dynamic deployment of services, and how our model can represent this in the following sections.

## 2.1 Knowledge Representation for Dynamic Deployment

Different tasks must be performed in order to deploy a service: Service Discovery, Resource Monitoring, Node Selection, Resource Allocation, Code Download & Deployment. We have situated our information model in the centre of all these tasks (see figure 1).

## 2.2 Objectives of our Work

The aim of our work is to provide a first step to model knowledge in the context of services deployment. To be able to express the information necessary for services and equipment configurations to deploy networked services with a general model, it is important to have only one set of defining vocabulary and one correct representation of the notions treated.

Firstly, we suggested the general definition of the terminology that was to be used. Then we present the object-oriented information model. Modelling enables us to have a formalized representation of the problem.

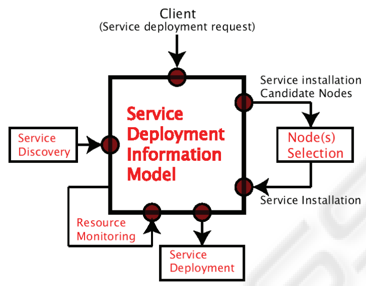


Figure 1: Service deployment model position.

Our information model (called "*Profiles Model"*) permits us to characterize the whole network and its hardware and software components (Diaz, 06). Our model is articulated around the concepts of *"service profile", "equipment profile"* and *"deployment profile"* in order to take into account all of the resources available and their evolution at the time of the activation of distributed services.

In the context of deployment the major notion to be treated is service. Several definitions of service exist in the literature. We have proposed a general definition: "a service provides functionality". This definition is independent of the composition model or implementation questions. The question of how the service will be deployed is treated with the notion of *"profile"* (Diaz, 05). Our proposal enables us to:

- Represent the actual state of network.
- Model the constraints of a service to be deployed.
- Take into account the notion of composition of services (via the concepts of heritage and elementary services).
- Count all the equipment hosts potentially to the deployment of a service.
- Characterize the network and its hardware and software components.
- Combine information of localization with information resources.

# 3 INFORMATION MODEL DESCRIPTION

We present the *Profiles Model* in this section. Two important notions of "Service" and "Profile" are the central points in our model. *Profiles Model* allows us to represent two points of view: the first is the current state of the network environment (that is the existing services and resources consumed or available), and the second is the necessary information needed for the installation of new services or the new profiles for already existing ones.

### 3.1 Notions Proposed

**Service.** a service is defined as a functionality. Several elements characterize this notion:

- a code and the interfaces, which define the algorithms of implementation and how to call the service;
- a behaviour, which shows the way, such as the entry, and are then transformed into exits;
- a state — a service can found in different states throughout its life (running, stopped, sleeping, cancelled, etc.);
- the composition relationships with services components.

These properties can be regrouped into two categories: static part and dynamic part (alternatives according to their instantiation):

- Static part: is related with the identifier, interfaces definition, organization, version…
- Dynamic Part: is related to the environment where the service in question evolves or moves. We define it through the concept of “service profile”.

The general description of a service involves several aspects:

- the semantic of this service itself;
- the requirements about nodes capabilities;
- the cost of instantiation of this service and the other characteristics such as QoS minimum.

**2) Profile.** enables us to represent the configuration of each element involved in the deployment, and also obtain the knowledge of the capabilities of network nodes. A profile defines a model adapted to a certain type of need. We apply this concept to define:

(1) Configuration of a material entity (equipment) with respect to the consumed/available resources (equipment profile)

(2) Particular configuration for the execution of a service with respect to the constraints and needs for this service (service profile)

(3) Precise configuration required by a user to deploy a new service (deployment profile).

**2.1) Service Profile.** allows us to define a particular configuration for the execution of a service. Each configuration specifies the constraints and needs related to its service execution. These constraints represent the use of the resources necessary to make the expected functionalities possible. The same service will be capable of several profiles, according to the specified constraints, and consumed resources. A profile of service will be characterized by:

- the whole of the resources requested for the execution of the service;
- a localization (physical and logical);
- QoS associated;
- constraints on unquestionable parameters: the supplier of the service, geographic coverage, price of the service, availability or length of the service…

**2.2) Equipment Profile.** represents the configuration of a material entity (equipment). This configuration includes several aspects:

- physical components: CPU, Storage, Battery, etc.;
- software components: which are seen through the list of profile services offered in this equipment;
- "physical+logic" components: example a chip with a software of integrated encoding. The idea is to be able to represent an evolutionary view of the equipment by defining its configuration with respect to the resources (physical and/or logical) currently available.

**2.3) Deployment Profile.** permits us to capture the services deployment requests in terms of resources and services required. If the service is a composed service, a deployment profile is made for each one of its services components.

### 3.2 Organization of Information Modelled

*Profiles Model* is based on the major notions defined in the last section. These notions enable us to represent the knowledge of the service’s execution environment. To deploy services in an automatic fashion it is necessary to control this knowledge about the environment (services, resources and equipments). The knowledge refers to information describing: the service's requests, service definitions and service deployments. Three major packages are defined to represent it. These packages permit us to define the lifecycle of services and to organize the classes by according the phases of deployment of services (see figure 2).

**RequestDeployment Package.** regroups a class to represent the request of service to be deployed. We define the actors involved in the deployment and the deployment profile class in order to represent the constraints in the service request.

**ServiceDefinition Package.** groups a class to define the semantic of service types known by the system (type of service: simple or composed, and data of each service), and the relationship with current information about the services deployed (represented by the service profiles class).

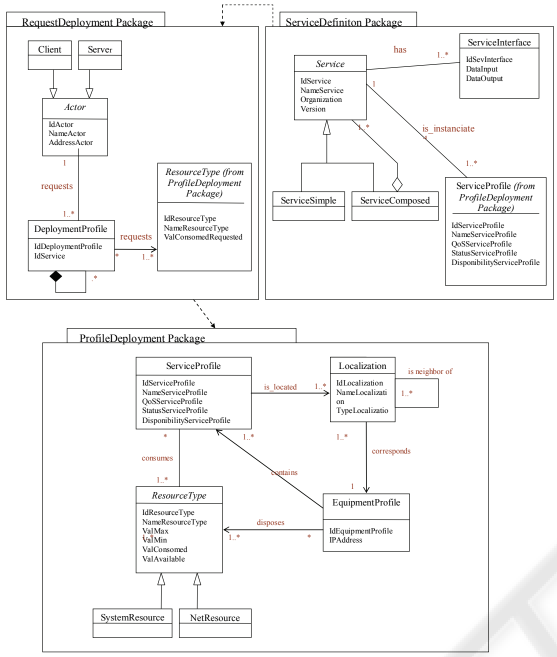


Figure 2: Packages for the definition of classes.

**ProfileDeployment Package.** groups a class to represent the service installed, and also the localizations and equipment used. In this package we show the relationships with the resource type class. Resource class is used to define the resources available (in the equipment profile), the resources consumed by a service profile, and the resources requested by a deployment profile.

## 3.3 Class Description

In this section we will describe the principal classes and their relationships offered by *Profiles Model*.

**Service Class.** allows for the defining of the various types of services known and handled by the system. We define here all the common characteristics of every service: name of service, API and version. This class is an abstract class; the instances are the classes *ServiceComposed* or *ServiceSimple*. When the service is composed it is necessary to define its relationships with other services. The code associated with each service, and the description of the data flows in entry and exit, are defined in the *ServiceInterface class*.

As an example, a high level service (Voice over IP) can be carried out in various manners, by several components and protocols. By modelling the needs for such a service in the form of elementary services and according to the availability of the resources, the activation of the service could be carried out in very heterogeneous contexts, for example by selecting the option best adapted for the resources and the context, by inter-connecting services meeting the same needs, or by activating footbridges.

**ServiceProfile Class.** makes it possible to define a particular configuration for the instantiation of a service. An example of service is a Video on Demand. Several other services are necessary for it instantiation: type of coder-decoder (TV-MPEG-2 or TV-MPEG-4), service of adaptation which definite which protocol(s) can be used for the transfer of the video, and service multicast. These different services can have different profiles. Each profile refers to the whole of the consumed resources, by example the power of the processor, the memory size, the size of storage in disc, etc.

**Equipment Profile Class.** refers to one view on localization. This class defines the configuration of a material entity. This configuration refers to characteristics of physical and also software components, in particular defining the list of service profiles offered by this equipment. The equipment profile gives a view of the resources available in one localization. Information about the equipment profile is updated (in terms of resource and service profile available) each time that one new service profile is installed.

**Deployment Profile Class.** defines the resources and also others services requested for a new service installation request. This profile, provided by the customer, is given to the system manager of the network. The deployment profile can be composed of some other deployment profiles, for example in the case of modelling the request of a composed service. In this situation, we express the relationships between the different services requests, such as in a composite relationship (black rhombus), because requests of services components refer only to the composed service in that relation.

## 3.4 Management Approach to Service Deployment

Deployment process implied different steps (See figure 3). We have presented in (Diaz, 08) a distributed approach in applying our model.

The first step to deployment is the request for service sent by user①. This request represents the entry into our system and is represented by the *Deployment Profile Request*②. The deployment profile enables us to verify the resources needed by the service request, and give a positive response (in the case where the system can find one or more nodes that satisfies this request) or a negative one (in the other cases)③. If the system can satisfy the service request, a new service profile is created④, and this information is added to the system⑤. At this time the system can update the information about available resources in the equipment profile, and the installation of services is then run by the deployment protocol⑥ ⑦.

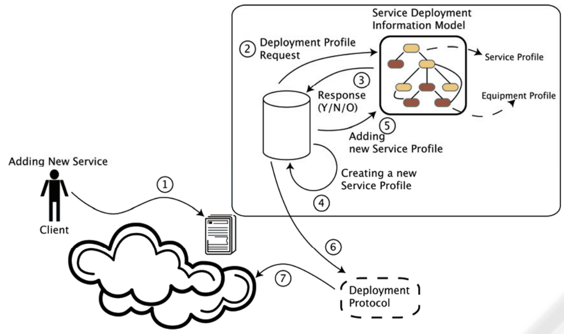


Figure 3: Service deployment process.

## 4 CONCLUSIONS

In this paper we have presented a discussion of knowledge modelling and our proposition to represent information needed in deployment tasks. Modelling information is the first step to representing knowledge. Our principal goal is to be able to dynamically facilitate the deployment of the services networks. This dynamic deployment of service is based on the characterization of the needs of service and the current state of the network. *Profiles Model* proposes the notions of profiles of service, equipment and deployment.

A first implementation of our information model is made to represent the different interfaces of our system. For any service request our tool makes it possible to establish all of the components necessary along with their state (localization, availability of resources, etc), including their relationship of interdependence. This cartography will make it possible for a manager to make effective decisions of deployment. Currently, we are working on the modelling of environment events to be manipulated, in order to complete the representation of knowledge in the service deployment domain.

## REFERENCES


Akhtar N.and al. Context Dissemination for Autonomic Communication Systems. WAC 2005, Athens, Greece, October 2-5, 2005.

Chao I., and al. Design, Implementation, and Evaluation of a Resource Management Multiagent System for a Multimedia Processing Grid. *OTM Workshops 2004*.

Clark D. and al. A Knowledge Plane for the Internet. *SIGCOMM'03*, August 25–29, 2003, Germany.

Diaz G., and al. Déploiement contrôlé des services de réseau, *Livrable L1.1 projet RNRT/Amarillo*, 2005.

Diaz G., and al. Modeling Data to Management Dynamic Services Deployment in Autonomic Networks, *IEEE/ICTTA 2006*, 24-28 april, Syria, 2006.

Diaz G., and al. Distributed Management to Services Deployment in Autonomic Networks, *IEEE/ICTTA 2008*, 7-11 april, Syria, 2008.

DMTF,Common Information Model (CIM) Standards. http://www.dmtf.org/standards/cim/

Eymann T., and al. Self-Organizing Resource Allocation for Autonomic Networks. *DEXA'03*, September 1-5, 2003, Prague, Czech Republic.

Gaïti D., and al. Autonomous Network Equipments. WAC 2005, Athens, Greece, October 2-5, 2005. P. 177-185.

Hass, R., and al. Autonomic Service Deployment in Networks. *IBM Systems Journal*, Volume 42, Issue 1 (January 2003) pp 150 – 164, ISSN:0018-8670.

Lewis D.,and al. Semantic Interoperability for an Autonomic Knowledge Delivery Service. *WAC 2005*, Athens, Greece, October 2-5, 2005.

Mulvenna M., and al. Knowledge Networks. *WAC 2005*, Athens, Greece, October 2-5, 2005. 99-114

OMG: Deployment & Configuration of Component-based Distributed Applications Specification, ptc/03-07-08.

Papazoglou and al. Service oriented computing: a research roadmap, SOC, 15.-18. Nov. 2005.

Pistore, M., and al. Planning and Monitoring Web Service Composition. *ICAPS'04*, Canada, June 3-7, 2004.

Quirolgico1 S., and al. Toward a Formal Common Information Model Ontology. *WISE 2004*, LNCS 3307, pp. 11–21, 2004.

Seleznyov A., and al. Distributed Knowledge Management for Autonomous Access Control in Computer Networks. *ITCC'04*, Volume 2 p. 433.

Simoni N., and al. Distributed Service Management for Intelligent Network. *ICCC'99*, 14-16 september, Tokio-Japon.1999.

Stojanovic L., and al. The role of ontologies in autonomic computing systems. *IBM SYSTEMS JOURNAL*, VOL 43, NO 3, 2004.

TMF, http://www.tmforum.org/browse.aspx?catID=2008. NGOSS-SID.

Wawrzoniak M., and al. Sophia: An Information Plane for Networked Systems. *HotNets-II,* Cambridge, MA, USA, November 2003.