

# Service Provisioning in Telecommunication Networks using Software Agents and Rule-based Approach

Ana Vrančić<sup>1</sup>, Krešimir Jurasović<sup>1</sup>, Mario Kušek<sup>1</sup>, Gordan Ježić<sup>1</sup>, Krunoslav Tržec<sup>2</sup>

<sup>1</sup>Faculty of Electrical Engineering and Computing  
Department of Telecommunications  
Unska 3, Zagreb, 10000, Croatia

E-mail: {ana.vrancic, kresimir.jurasovic, mario.kusek, gordan.jezic}@fer.hr

<sup>2</sup> Ericsson Nikola Tesla  
R&D Centre  
Krapinska 45, Zagreb, 10000, Croatia  
E-mail: {krunoslav.trzec}@ericsson.com

**Abstract - Increased competition in the telecommunication market has forced network operators and service providers to roll out new services at a much faster rate, taking under consideration user preferences. Therefore, a business-driven service provisioning, which takes into account business-to-business relationships as well as preferences of targeted user groups, is proposed. It utilizes agent technology and rule-based approach in order to provide flexible as well as easier creation and deployment of new multimedia services in the network operator's infrastructure.**

## I. INTRODUCTION

In 2G mobile telecommunication systems service logic was placed in the network nodes and directly related to technology used by a network operator. Services were developed either by operator or by equipment vendors and offered to operator's subscribers as part of a subscription. The development of a new service has often been very complex and expensive due to the fact that the service had to be implemented directly in the core of the operator's network. Consequently, a very limited number of services were offered to the customers [1]. With deregulation of telecommunication market and advent of the 3G mobile systems, which offer the convergence of the Internet and traditional telecom services, there is a need for more flexible and easier service creation and deployment. Moreover, the network operator is likely to collaborate more tightly with value-added service providers (VASPs), such as content providers and software suppliers, in order to satisfy an increased demand for ubiquitous, personalized and terminal-aware services.

It is expected that increased competition in the telecom market will result in the creation of a variety of multimedia services with different types of content. Moreover, the operators should deploy newly created services in a rapid and cost effective way. In order to provide a solution for efficient provisioning of multimedia services in the network, we have taken advantage of agent technology and rule-based approach [2]. The proposed solution enables automated service

deployment in the network as well on the users' terminals. The agent-based service provisioning solution provides flexible and reconfigurable service offerings by adding/removing and changing versions of software components (e.g. audio/video codec's) that are required for proper service execution in both network nodes and users' terminals. The reconfigurability allows the service settings to be change depending on the user profile and type of used terminal. Adoption of rule-based approach has enabled definition of service deployment strategies that takes into account both content and software components offered by various content providers and software suppliers.

The proposed service provisioning solution is based on three emerging technologies: Web services, software agents and rule engines. Web services provide VASPs an interface to the network operator's provisioning system, e.g. rule engine that executes service deployment strategies. Software agents are used for business-driven deployment of new multimedia services on the network nodes and users' terminals. The rule engine initiates deployment of a new service when all preconditions about facts (i.e. events) in a service deployment strategy are met. The rest of the paper is organized as follows: Section 2 provides related work in the area of automated service provisioning. Section 3 describes technologies used in the proposed solution. Multi-agent system for business-driven service provisioning is explained in section 4. Section 5 considers a case study describing service deployment in the 3G mobile network and Section 6 concludes the paper.

## II. RELATED WORK

There are several proposals for automated service provisioning systems. The MOBIVAS system is one of them [3]. It provides deployment and management of value-added services (VAS) in 3G networks. The Value Added Service Manager is a central component of the system that deals with the deployment and configuration of new services. It also provides service discovery portal, as well as accounting and traffic monitoring

functions. Another example of a service provisioning system can be found in [4]. The paper presents a solution based on Generic Network Model developed by the Tele Management Forum (TMF). The solution provides a technology and vendor independent service provisioning. Independency is accomplished with the use of rule engines. The advantages of using agent technology in service provisioning area are explained in detail in [5]. Our automated business-driven service provisioning solution combines some of the best practices explained in these papers.

### III. USED TECHNOLOGIES

This section gives an overview of software technologies used in our business-driven service provisioning solution.

#### A. Web Services

A Web service [6] provides support for interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format. Other systems interact with the Web service in a manner prescribed by its description using SOAP [7] messages, typically transferred over the network using HTTP with an XML serialization. A Web service description is specified by Web Service Description Language (WSDL) [8]. The description defines the message formats, data types, transport protocols and transport serialization formats that can be used between the requester and the provider of Web service. It also specifies one or more network locations at which the service can be invoked. The Web service is invoked in a way that the requester and provider entities (e.g. software agents) become known to each other through the use of service discovery mechanism. Then the requester and provider agree on the service semantics that will govern the interaction between them. Finally, the requester and the provider entities exchange messages.

In order to prevent security threats regarding Web service usage security mechanisms for message integrity and confidentiality, as well as message encryption (e.g. Secure Socket Layer) can be adopted. All mechanisms can be used separately or in a conjunction with each other.

#### B. Java Agent Development Framework

Java Agent DEvelopment framework (JADE) [9, 10, 11] is a software framework used for development of agent-based systems. An agent is autonomous, proactive, and social software entity that acts on behalf of its user [12]. Furthermore, it can be mobile and, thus, migrate between the nodes in the network in order to perform different operations on them without any user interaction. Mobile agents can migrate from host to host in a network at times and to places of their own choosing. The state of the running agent is saved, transported to the new host and restored, allowing the agent to continue where it left off. Mobile agents differ

from the Java applets which are programs downloaded as the result of a user action and executed from beginning to end on one host.

JADE provides mechanisms for agent communication, migration, control and security. The mechanisms implemented in JADE are in compliance with IEEE Foundation for Intelligent Physical Agents (FIPA) [13] specifications. These specifications define standard agent interaction protocols and other key aspects of a multi-agent system allowing interaction between different agent platforms and software components. JADE can be considered as both an agent middleware that implements an agent platform and a development framework. It deals with all those aspects that are not peculiar of the agent internals and that are independent of the applications, such as message transport, encoding and parsing, and agent life-cycle.

Each running instance of the JADE runtime environment is called a Container as it can contain one or more agents. The set of active containers is called a Platform. A single special Container, so called Main Container, must always be active in a platform and all other containers register with it as soon as they start. If another Main Container is started somewhere in the network it constitutes a different platform to which new normal containers can possibly register. Besides the ability of accepting registrations from other containers, a Main Container differs from normal containers as it holds two special agents (automatically started when it is launched):

- *Agent Management System (AMS)*: AMS provides the naming service (i.e. ensures that each agent in the platform has a unique name) and represents the authority in the platform (e.g. it is possible to create/kill agents on remote containers by requesting that to the AMS);
- *Directory Facilitator (DF)*: DF provides a Yellow Pages service by means of which agents can register services they perform. Agents can also use this service to find other agents providing the services they require in order to achieve their goals.

Fig. 1. shows two JADE platforms, one composed of two containers (Platform 1) and the other composed of one container (Platform 2). JADE agents are identified by a unique name, named Agent Identifier (AID). Assuming that the agents know each other's name, they can communicate transparently regardless of their actual location in the:

- same container (e.g. Agent 2 and Agent 3);
- different containers in the same platform (e.g. Agent 1 and Agent 2);
- different platforms (e.g. Agent 1 and Agent 4).

The communication among agents is based on asynchronous message passing which offers flexible and efficient messaging. The JADE creates and manages a queue of incoming Agent Communication Language (ACL) messages, private to each agent which can be accessed by the agent from any location in the Platform. Agents can access their queue via a combination of several modes (blocking, polling, timeout, and pattern matched). The messages are specified by the ACL Message Structure Specification defined by the IEEE

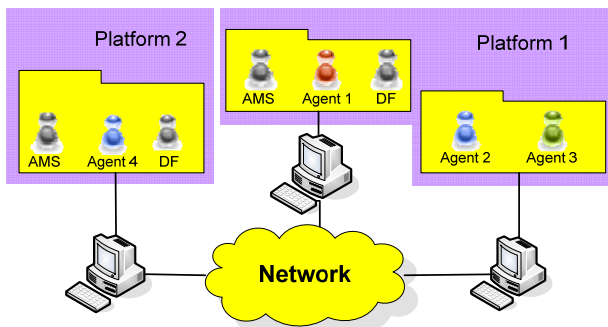


Fig. 1. Example of two JADE agent platforms

FIPA specifications. This allows the agent to communicate with other agents running on agent platforms written in other programming languages.

The JADE enables development of robust multi-agent systems by providing two services:

- *Replication Service.* It solves the problem when the Main Container is disabled. This is accomplished by creating one or more replicated Main Containers on other hosts. Information contained within the AMS on the Main Container is copied on the replicated containers as well. In case the Main Container fails a replicated Main Container located on another host is activated and continues to perform the tasks of a Main Container;
- *Persistence Service.* It allows storing the information about service provided in the network. This information can later be retrieved in case of Main Container failure.

### C. Java Expert System Shell

Rule-based programming is appropriate for problems that are difficult to solve using traditional algorithmic methods [14]. It is declarative in a sense that describes what the program should do, but omits much of the instructions on how to do it (i.e. control flow). Therefore, it has to be executed by some kind of runtime system that understands how to control flow and use the declarative information to solve problems. Because declarative programs include only the important details of the solution, they can be easier to understand than procedural programs. Declarative programming is often the natural way to tackle problems involving control, diagnosis, prediction, classification, pattern recognition or situational awareness, in short, many problems without clear algorithmic solutions.

A rule-based program doesn't consist of one long sequence of instructions; instead, it is made up of discrete rules, each of which applies to some subset of the problem. A rule engine determines which rules apply at any given time and executes them as appropriate. As a result, a rule-based version of a complex program can be shorter and easier to understand than a procedural version. Writing the program is simpler, because you can concentrate on the rules for one situation at a time. Modifying the program is also simpler.

Our service provisioning solution has adopted Java Expert System Shell (JESS) [15] as both a rule engine and a scripting language for specification of rules. A

rule is a kind of instruction or command that applies in a certain situations. Te JESS rules are a lot like the "if-then" statements of traditional programming languages. The rule-based system uses rules to reach conclusions from a set of premises. Every JESS rule engine consists of (see Fig. 2.):

- working memory (fact base);
- inference engine;
- rule base.

A working memory contains facts the rule-based system works with. These facts are then used as premises of rules or to store rule conclusions. An inference engine controls the process of applying rules to the working memory to obtain outputs of the system. Inference engine works in discrete cycles. First, all rules are compared with facts in the working memory to decide which ones should be activated during the cycle. List of rules whose "then" statement will be executed or fired is stored on an agenda. To complete the cycle, first rule on the agenda is fired and the entire process is repeated. Rule that is fired can change the state of the working memory. An inference engine consists of

- pattern matcher;
- agenda;
- execution engine.

The purpose of the pattern matcher is to decide which rules apply (taking under consideration the content of the working memory). An execution engine can add, remove or modify existing facts in the working memory or it can interact with the outside world to start some action. The rule base contains all the rules that system knows about. They can simply be stored as strings of text but very often a rule processor transforms them into some form that the inference engine can work with more efficiently.

Jess rule engine has one disadvantage. Since it is a declarative language where the behavior of the system is defined by rules, triggered by the status of the environment, it is almost impossible to write a program that is executed line by line as can be done in a procedural language. For our application domain this constraint does not present a problem since business rules for service deployment consist out of discrete steps that can be easily represented as individual Jess rules.

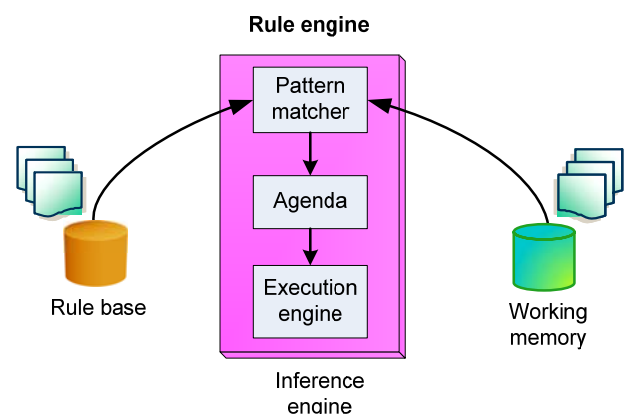


Fig. 2. The architecture of JESS rule-based system

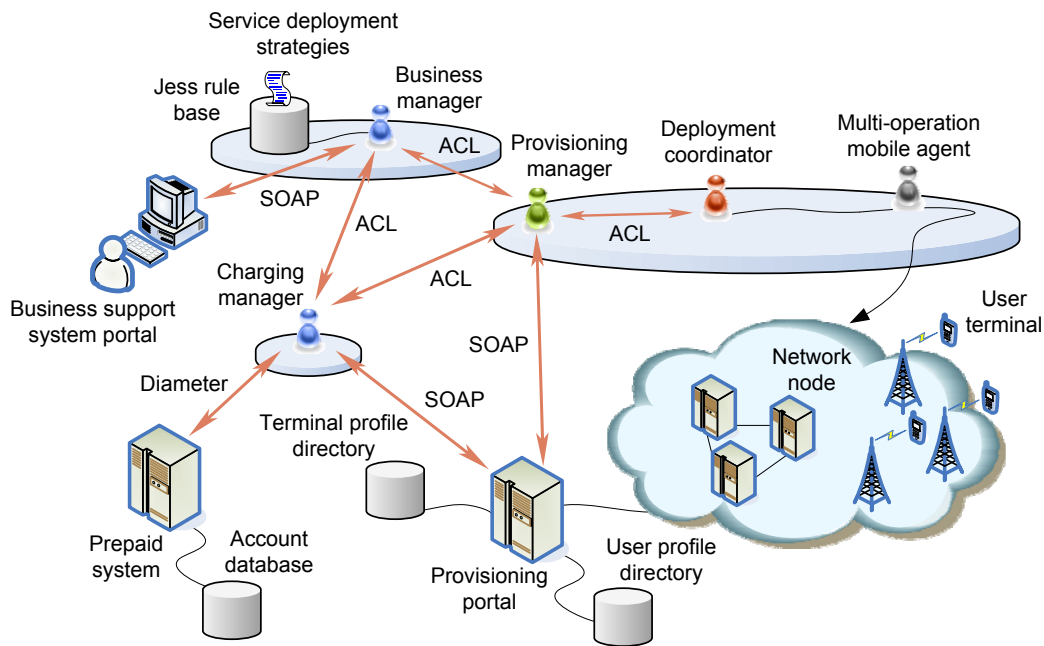


Fig. 3. Multi-agent system for business-driven service provisioning

#### IV. SYSTEM ARCHITECTURE

Our solution combines Web services, software agents and rule-based approach in order to provide automated service provisioning system. Web services are chosen since they enable a network operator to expose functionalities of its service provisioning system as standard interfaces. Furthermore, by using Web services our system can be easily combined with other functions in the operator's network. Software agents are designed for service provisioning purposes, while the rule engine is used for defining of service deployment strategies (that consists of a number of Jess rules).

Fig. 3. shows the proposed system for business-driven service provisioning in the 3G mobile networks. The system is comprised of the following components:

- *Business Support System (BSS) Portal*: It is used either by the operator's personnel in business departments or by VASPs as an interface towards network operator's provisioning system. It allows the VASPs to notify the network operator about business-related events important for execution of a service deployment strategy. It also enables software providers to notify the system about the release of a new version of software;
- *Business Manager*: The agent which receives business-related events and forwards it to its rule engine which is responsible for handling of the rules specified by service deployment strategies. When a deployment strategy is ready for execution in the network, the Business Manager agent initiate the strategy fulfillment;
- *Provisioning Manager*: The task of this agent is to coordinate service deployment and user account related procedures in the network. The Business Manager agent initiates the deployment of a new service by sending a request to the Provisioning Manager agent which then begins fetching required

software components from a software supplier's download servers. After that, it starts the process of installing the software components on the network nodes and user terminals. The Provisioning Manager agent is connected to the Provisioning Portal in order to access information related to users and their terminals;

- *Deployment Coordinator*: This agent is responsible for the coordination of a new service installation on the appropriate network nodes and user terminals. It delegates the installation tasks to RMS multi-operation mobile agent [16]. This agent is responsible for deploying the service on network nodes;
- *Charging Manager*: The agent that is used as an interface toward the prepaid system (PPS), i.e. account database. Communication between the agent and the prepaid system is done by the use of Diameter protocol.

All the components described are located in the network of the network operator. Other VASPs access the functionalities of the network exclusively through the BSS portal. Standards defined by the web services, BSS portal is composed of, allow the operator to deploy services by using their existing systems thus reducing cost of implementing the proposed solution. This approach also allows the VASPs to deploy services in the network without knowing how the deployment is performed. The risk that the process of deployment could cause the network to malfunction is reduced considerably.

Our solution of an automated service provisioning system is designed to reuse components that are already present in the network. This is accomplished by adding new rules that define how these components are used. It also might be necessary to upgrade some of the agents in the system with new interfaces that allow them to use new protocols.

## V. SERVICE DEPLOYMENT STRATEGIES

When a network operator wants to deploy a new multimedia service into its network there are several steps that have to be made:

- Content for multimedia service must be created or purchased from content providers;
- Software components required for service execution have to be developed or obtained from software suppliers;
- Service has to be installed on the appropriate nodes in the network taking under consideration node configuration options. When a new version of a service is released this process has to be repeated;
- User terminals have to be configured as well;
- To provide personalized services to the users every service has to be adapted according to the user and terminal profiles.

These steps are very costly and time consuming when performed manually by the operations support system (OSS) personnel. Automation of these procedures by applying our service provisioning solution, which utilizes software agents and rule-based approach, significantly decrease operations cost.

The proposed business-driven service provisioning solution has been tested during the Summer Camp 2006 [17]. Fig. 4. shows the architecture of the proposed rule based system. It is comprised out of 4 parts. The first part is the JESS rule engine that handles all the rules in the system, and keeps track of its preconditions. Deployment strategies comprise the second part. Every deployment strategy represent a set of rules that defines when this service is going to be deployed. It also defines steps that are needed for the deployment of a service in the network. Network operator defines the rules that specify different operations performed in each steps of the deployment process. Deployment strategies can be easily changed or added to the system without the need for a restart. The Business Manager agent is used to provide an interface toward the other components in the service provisioning system while the fact base is used to store all the relevant service deployment facts.

Every rule defined in the JESS rule engine has one or more precondition. Precondition must be satisfied before a rule can be activated and executed. For example, let us suppose that a network operator wants to define a multimedia service that depends on some software components that are under development. The multimedia service has to be deployed during the summer and has to be offered only when a certain number of users are present in a specified region. One way to roll out this service is to check for all condition periodically and launch the service when all the conditions are met. A more effective way is to define the preconditions in the rule-base system. When they are all met, the JESS rule engine will start the deployment of a service without the need for human intervention. Fact base in this case contains facts about the available software components, season period and the number of users in the specified region.

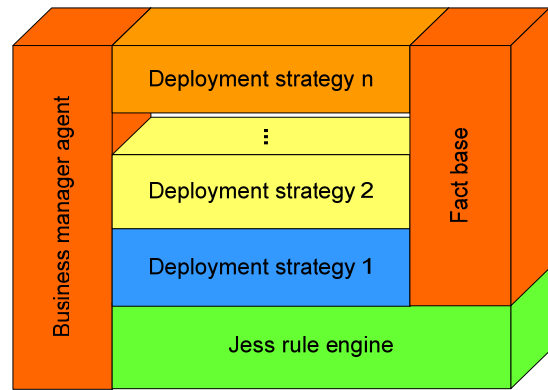


Fig. 4. Architecture of the rule-based system

## VI. CASE STUDY

In order to verify the proposed solution for automated service provisioning, we have performed deployment of a number of multimedia services in the network. Services were deployed according to defined strategies that take into account the following business-related parameters (i.e. events):

- *Content type*: Defines the type of the service;
- *Device type*: Specifies the targeted type of a user terminal. Some services can be created for a specific type of user terminals. In the case study we have limited user terminal types to mobile phones;
- *Target location*: Used to define a region where the service is going to be launched. Some services could be specialized for only some regions in a country or a city;
- *Time period*: Defines the time when the service is going to be launched. In the case study we have limited the periods to the seasons of the year;
- *User groups*: Specifies the targeted types of users this service is intended to. We have chosen to select the "Take five" user segmentation model developed by the Ericsson Consumer research lab [18]. This model defines five groups of users with similar characteristics that are extracted by studying people's basic values and attitudes towards technology;
- *Required QoS*: Defines the application level QoS;
- *Expected service deployment cost*: Cost parameter is used during the process of service deployment optimisation process;
- *Required software components*: The software components required by the service.

The scenario is performed in the following way: First the service deployment strategies are added into the system by the use of Web-based interface of the BSS portal. Events important for the execution of service deployment strategies (e.g. release of required software component) are sent to the rule engine by using the provided Web services. When all the conditions for the service deployment are fulfilled the rule engine launches



the deployment process for each service. At the end of a scenario all the services are successfully installed on the network nodes and users' terminals.

## VII. CONCLUSION

We have implemented business-driven service provisioning system that automates service deployment procedures in the 3G mobile networks. The automation is achieved by the use of software agents in combination with a rule-based approach. The proposed service provisioning solution allows flexible definition of service deployment strategies taking into account business-to-business (B2B) relationships and users' preferences.

The proposed solution was validated in a scenario with a number of multimedia services with different parameters. All the services were successfully deployed on network nodes and users' terminals thus proving that our solution can be used for business-driven service provisioning in 3G mobile networks.

## ACKNOWLEDGMENT

The work presented in this paper was developed at the Summer Camp 2006 in collaboration with the Faculty of Electrical Engineering and Computing, Department of Telecommunications and the Ericsson Nikola Tesla company.

## REFERENCES

- [1] N. Houssos, A. Alonistioti, L. Merakos, E. Mohyeldin, M. Dillinger, M. Fahrmaier, and M. Schoenmakers, "Advanced Adaptability and Profile Management Framework for the Support of Flexible Mobile Service Provision", IEEE Wireless Communications Magazine, Special Issue on (R)Evolution towards 4G Mobile Communication Systems, Vol. 10, No. 4, 2003.
- [2] R. G. Ross, "Principles of the Business Rule Approach", Addison-Wesley, Upper Saddle River, 2003.
- [3] N. Houssos, V. Gazis, S. Panagiotakis, S. Gessler, A. Schuelke, and S. Quesnel, "Value-added Service Management in 3G Networks", In Proceedings of the 8th IEEE/IFIP Networks Operations and Management Symposium (NOMS 2002), p. 529-545, 2002.
- [4] Q. Kong, I. Rose, and D. Cameron, "Towards Technology Independent and Automated Service Activation and Provisioning", In Proceedings of the 8th IEEE/IFIP Networks Operations and Management Symposium (NOMS 2002), p. 931-933, 2002.
- [5] S. Albarak, "Agent-Oriented Technology for Communications", Communications of the ACM, vol. 44, no. 4, p. 30-32, 2001.
- [6] Web Services Architecture, <http://www.w3.org/TR/ws-arch/>
- [7] SOAP 1.1, <http://www.w3.org/TR/SOAP/>
- [8] WSDL 1.1, <http://www.w3.org/TR/wsdl/>
- [9] F. L. Bellifemine, G. Caire, and D. Greenwood, "Developing Multi-agent Systems with JADE", Wiley, Chichester, 2007.
- [10] JADE Administrator's guide, <http://jade.tilab.com/>
- [11] JADE Programmer's Guide, <http://jade.tilab.com/>
- [12] M. Wooldridge and N. R. Jennings, "Intelligent Agents: Theory and Practice", Knowledge Engineering Review, vol. 10, no. 2, p. 115-152, 1995.
- [13] FIPA web site, <http://www.fipa.org/>
- [14] E. F. Hill, "Jess in Action: Rule-Based Systems in Java", Manning Publications, 2003.
- [15] Jess web site, <http://herzberg.ca.sandia.gov/jess/>
- [16] G. Jezic, M. Kusek, T. Marinic, I. Lovrek, S. Desic, K. Trzec, and B. Dellas, "Grid Service Management by Using Remote Maintenance Shell", Grid Services Engineering and Management, vol. 3270, p. 136-149.
- [17] A. Vrancic, "Definition of Service Fulfillment Business Strategies", Proceedings of the Summer Camp 2006, 2006.
- [18] W. Anderson, I. Freeman, A. James, and A. Ljung, "Mobile Media and Applications – From Concept to Cash", Wiley, Chichester, 2006.