MDA-based Service Creation for OSA/Parlay within 3Gbeyond Environments

Thomas Magedanz[1], Karsten Knüttel[1], Olaf Kath[2]

[1]*TU Berlin/FhG FOKUS*

[2]*TU Berlin/IKV++ Technologies AG*

{ magedanz, knüttel }@fokus.fraunhofer.de, kath@ikv.de

Abstract

At the Fraunhofer Institute for Open Communication Systems, an open and independent 3Gb infrastructure test bed is being provided. The infrastructure test bed acts as a technology play ground stimulating ideas and identifying research challenges by bringing together academia and industry, i.e. content providers, application developers, vendors and operators. Corresponding development, provisioning and management tools do support this infrastructure. As part of these, a chain of development tools, interconnected to each other using MDA principles, is being experimented with in order to allow the rapid definition and implementation of 3Gb services. The chain is defined based on some core specifications under the MDA umbrella. In this paper, we give an overview of the approach and discuss several research directions.

Keywords: 3Gb, EDOC, FOKUS 3Gb testbed, IMS, MDA, medini, IN, Next Generation Network Infrastructure OSA, Parlay, Service Delivery Platforms, Service Creation

1. Introduction

Today the mobile telecommunications infrastructure is driven by the need to support dynamically the provision of an open set of seamless mobile applications on top of different wireless network technologies in face of changing value chains and business models. Based on shorter technology innovation cycles this requires the ongoing replacement and extension of service delivery platforms enabled by new information technologies and software tools.

New open network service delivery platform standards, such as the 3GPP Open Service Architecture (OSA) and Parlay application programming interfaces (APIs), are based on the principle of service programming support with network protocol abstraction and the exploitation of state of the art information technologies, such as CORBA, Java and web services and related development tools.

The most innovative software development approach, the Model Driven Architecture (MDA) aims for providing total freedom to the applications development. The MDA is an approach to the full lifecycle integration and interoperability of enterprise systems comprising of software, hardware, humans, and business practices. It provides a systematic framework to understand, design, operate, and evolve all aspects of such enterprise systems, using engineering methods and tools. The framework is based on modelling different aspects and levels of abstractions of such systems, and exploiting interrelationships between these models.

Consequently it seems logical to apply the MDA approach in the mobile telecommunications environment in order to provide as much as possible abstraction from the underlying service delivery platform technologies to enable a highly flexible service development.

In this paper we propose and illustrate the use of MDA for the rapid and highly automated development of mobile telecommunication services on top of OSA/Parlay based service delivery platforms. Besides the general architecture we provide an usage overview to illustrate the approach. The approach presented here was prototypically realized within the globally recognized open OSA/Parlay playground, which is part of the FOKUS 3G beyond test and development centre. We will also illustrate that this usage of the MDA approach will also enable the smooth transition from OSA/Parlay SDPs (SCFs) towards upcoming SIP-based 3G IP Multimedia Systems (IMS).

2. Open Telecom Service Delivery Platforms

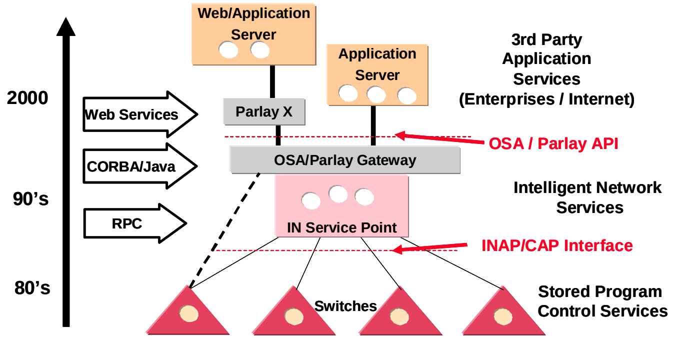

Figure 1 Telecom Service Delivery Platform Evolution

In face of the convergence of telecommunications, internet and IT and the changing value chains and increasing market deregulation, telecom service delivery platforms (SDPs) have been under frequent evolution for the provision of open set of value added mobile multimedia services (as shown in figure 1). Based on the Remote Procedure Call concept, AT&T and later Bellcore (now Telcordia Technologies) has invented the Intelligent Network (IN) concept – a centralized computer based system, controlling remotely by means of a dedicated signalling protocol (the IN Application Protocol – INAP) via a dedicated signalling network – the famous signalling system no. 7 (SS7) in real time the switching nodes for the provision of value added services [1]. Moreover, the idea of the Intelligent Network has been to offer a set of generic service building blocks (SIBs) to create a multitude of value added services, similar to a distributed operations system, that enables plenty of applications to be executed. This means that the combination of IT middleware and telecommunications system enlarged drastically the programmability of the telecommunications network. It has to be recognized, that the IN concept decoupled the service provision from the underlying network, which allows in principle to provide IN-based services on top of different bearer networks. Thus the IN has been applied originally to the PSTN, subsequently to ISDN, and in the end of the 90th to mobile networks (i.e. GSM) under the name of “Wireless IN” or CAMEL (Customized Logic for Mobile Enhanced Logic). The later approach has been standardized by ETSI (European Standardisation Institute) and the 3rd Generation Partnership Project (3GPP) as extension to GSM/GPRS systems of Phase 2+. Today the IN is used all over the world for value added services provision, like Universal Access Numbers, Virtual Private Networks, Freephone, Premium Rate calls and messages, and most particular prepaid.

One main principle of the IN concept was to exploit the capabilities of the state of the art information technologies in order to enlarge the developer community. As in the nineties the notion of object oriented programming, software languages like C++, and coincidently distributed object-oriented systems, such as OMG´s Common Object Request Broker Architecture (CORBA) and Sun´s Java 2 Enterprise Edition (J2EE) appeared, the IN architecture itself has undergone some extensions [2, 3]. The notion of Distributed Broadband IN systems have emerged. However, a substantial change of the IN system architecture has not been adopted, as the IN system had besides all described advantages some inherent limitations. It was still coupled with the underlying network protocols and switching equipment and thus limited. This means that there has not been a full decoupling of the service level and the switching level, and thus the programming of IN services was quite complex and only possible by a limited set of special telecom experts. Additionally, the business models of IN-based telecommunication networks were quite closed.

In face of these limitations, the ongoing convergence of telecommunications, IT and the internet, the notion of a new programming paradigm for telecommunications emerged – open application programming interfaces (APIs) [4, 5, 6]. Driven by the need for a common multi media service platform for converging networks, and the proven commercial usability of distributed object oriented platforms, new standards for open service platforms emerged. The Parlay Group, a group of operators, vendors, and IT companies, has started in 1998 with definition of an open network Parlay API. This API is inherently based on object oriented technology and the idea to allow – if desired by the business model – third party application providers to make use of the network or better speaking the value added service interfaces (see figure 3). However, today the best view on Parlay is to consider it as some kind of telecom specific enterprise application integration (EAI) platform technology, allowing service providers to develop value added applications on top of a different and/or changing network environment. This allows a smooth network technology evolution below the developed applications.

3GPP has aligned in 2001 their work on an Open Service Access (OSA) API with Parlay, as also ETSI does for their SPAN (Service Provider Access to Networks) APIs in the same year. Today all three standards are completely aligned and thus represent probably the most accepted standard in this domain.

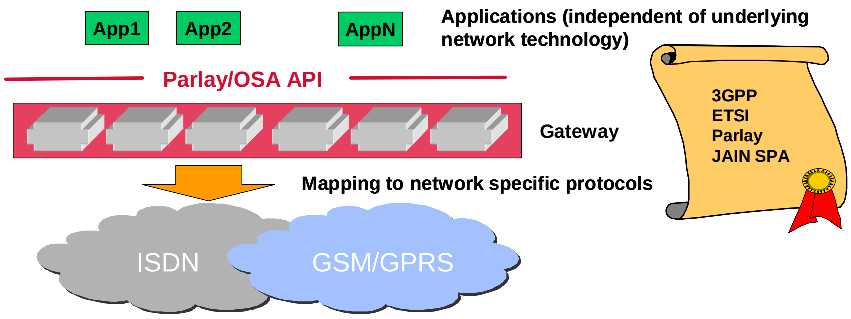

Figure 2 State of the art SDP approach: Open Network APIs (OSA/Parlay)

In summary, it can be stated, that today OSA/Parlay APIs represent the state of the art technology for implementing open service delivery platforms on top of various bearer networks, including all IP networks. Particular in the last regard, it has to be stated, that OSA/Parlay is considered as one out of three options (CAMEL, OSA, SIP Application Server) for implementing value added services in the so called UMTS Release 5 all IP Multimedia Subsystem (IMS).

3. The OSA/Parlay API in a Nutshell

Looking at the OSA/Parlay API in some more detail, it is important to recognize the open / extensible nature of the API architecture. The main idea is to provide in a dedicated network node known as OSA/Parlay Gateway operated by the network operator an open set of service interfaces, which exhibit specific value added service capabilities. This comprise call control, user interaction, messaging, data session control, location, presence, charging, etc.

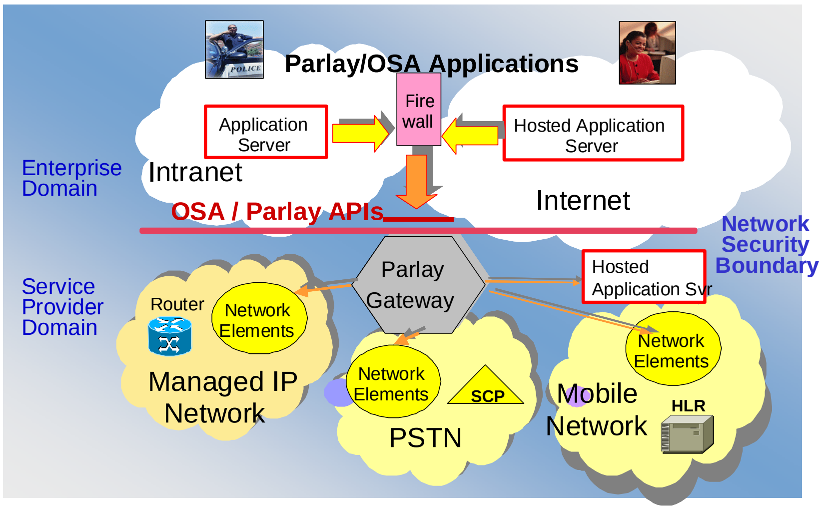

Figure 3 The OSA / Parlay ecosystem

A dedicated interface known as the framework is in charge to register/discover new service interfaces, perform application / network authentication, service level agreements, etc. Important is the fact that the API is network independent, i.e. in principle each network (note that the type of the network doesn't matter) will provide its own OSA/Parlay gateway, and one application can make use of several gateways. This means that an application can run with the same logic simultaneously on top of a fixed telephone network and an voice over IP network. Today OSA/Parlay technology is going to be deployed slowly all over the world.

As one major target of OSA/Parlay is to make the networks programmable by means of the state of the art in middleware technologies in order to make the network programmable by the application providers, the API has been enhanced by the use of the new web services paradigm, which combines the eXtensible

Markup Language (XML), Web Service Description Language (WSDL), and Simple Object Access Protocol (SOAP).

These are considered also as the main starting point for what the Open Mobile Alliance (OMA) is looking for – mobile web service enablers. OMA is considered today as a super standards forum, bringing together various other for a, such as wireless village, WAP forum, etc. The main target is to approve useful wireless standards for enabling rapid service delivery.

4. medini – Techniques and Tools for MDA

4.1. What's medini?

medini [16] essentially is a set of tools for building a modelling and development infrastructure. In such an infrastructure, different tools from different vendors and for different purposes are tight together allowing the output of one tool to be used as an input for another tool. Each of these tools or modelling techniques supports a different phase or activity in the development process for a software system. medini is used to tie modelling and development tools supporting different phases and activities of systems engineering process together to form a homogenous environment. The foundation for medini is the Meta Object Facility (MOF).

In Figure 4, an overview of the medini modelling infrastructure is depicted. It can be seen, that the infrastructure forms an extensible bus, containing a number of logically distinct repositories. System modelling and development tools are connected to this infrastructure. A modeller may define a certain piece of a system specification, for instance a collection of interconnected requirements. For this purpose, he uses a requirements engineering tool. The tool is being connected with a requirements models repository within the medini modelling infrastructure, which stores the defined requirements. These requirements can be read by another tool, e.g. a tool for system analysis, and can be used as an input for the modelling task. The developed analysis model in turn can be used by a system design tool and so forth.

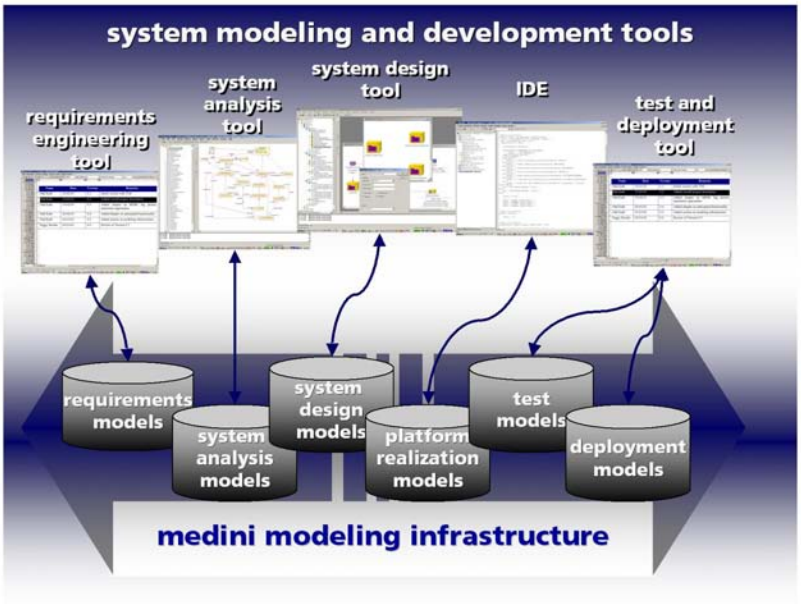

Figure 4 medini Modelling Infrastructure Overview

Now, the medini infrastructure is being used to realize a model driven style of system construction. The platform independent modelling tools are based on UML and UML extensions such as the Enterprise Distributed Object Computing [18] and Enterprise Application Integration profiles. The platform specific modelling tools are also based on UML and specific UML profiles for telecom platforms. Integrated development environments are usually being used for programming or testing purposes. A particular integration platform supports the integration of these components, their operation and the maintenance of the resulting system. Each of the tools alone support a particular task in the system development process. But, how can the result of one task automatically be used in a subsequent task? medini provides the solution to this problem in term of a modelling infrastructure, a kind of "modelling and development bus", that interconnects the different tools.

Technically, medini supports the following aspects and technologies:

- Persistent storage of models and other artefacts in repositories based on Meta Object Facility (MOF),
- Automatic creation of these repositories out of MOF compliant metamodels,
- Transactional access to models in the repository based on CORBA and XML,
- Semantic verification of models within repositories based on the Object Constraint Language (OCL),
- Transformations between models in repositories based on an early implementation of the upcoming Query/View/Transformations (QVT) standard.

With the Meta Object Facility, medini is being constructed to integrate different, possibly domain specific modelling techniques and tools. With a MOF compliant metamodel, the semantics of the modelling concepts can be defined in a formal way. Out of this metamodel, model databases (repositories) can be automatically generated which are used to manage models. These model repositories host models and allow for mechanisms such as model traceability and model consistency. Tools are attached to these model databases which express models to the user. Based on the notation definition(s), these tools can express the same model in different notations.

Like the tools to express models, tools to automate transformations between models and the generation of code use the model repositories. The rules for these processes are defined based on the (source and target) metamodels.

The technologies used to construct the modelling infrastructure are very flexible in the sense of being free to integrate any modelling technique or platform for which a MOF compliant metamodel is available or can be defined. This fact remains true for

- different modelling techniques at the PIM and PSM level,
- different platforms as a target of model transformations,
- different programming languages and programming patterns and
- different model transformation rules.

4.2. How's medini being used in an OSA/Parlay environment?

In the actual project context, medini is being customized to especially support the development of telecom services out of platform independent models:

- a concrete platform independent modelling technique - the Enterprise Distributed Object Computing (EDOC) standard – is the basis for PIM modelling,
- the platform for which specific models are developed is already determined to be a Parlay/CORBA environment deployed at FhG FOKUS,
- a concrete programming environment to express generated code (fragments) is being chosen to be the eclipse [19] platform and
- the model transformation engine takes specific input sets of rules particularly for EDOC PIM to Parlay/CORBA PSM and Parlay/CORBA PSM to the Java programming environment transformations.

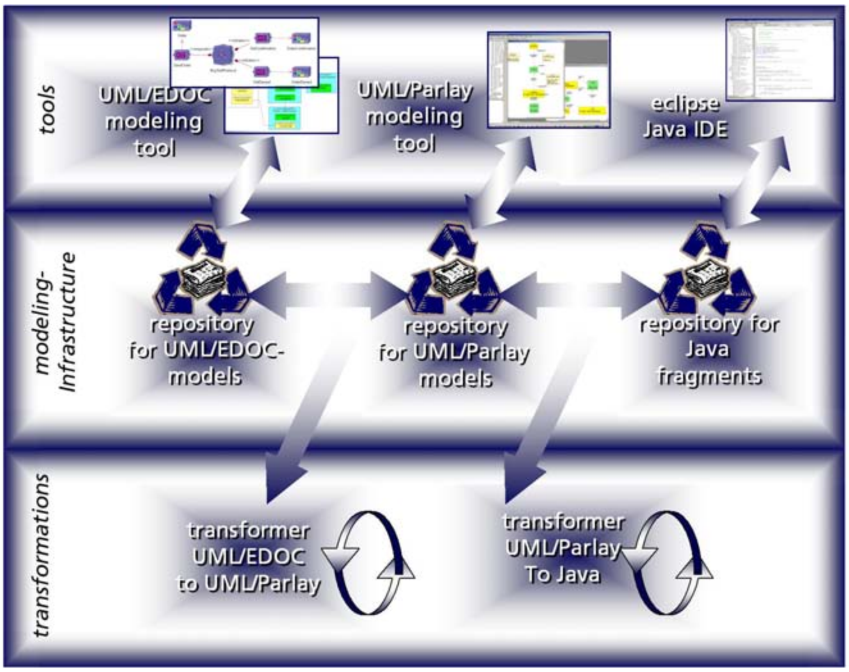

Figure 5 medini customization for Parlay

An overview of the customized medini infrastructure can be found in Figure 5. In the following table, we give an overview of the features the customized medini infrastructure for OSA/Parlay supports.

category	medini part	Features
modelling infrastructure	medini UML2MOF	- support for MOF 1.4 in combination with OCL 2.0 - several extensions to standard for consistency maintenance, conformance validation and change management - Support for persistency and serialization - Rational Rose based modelling tool for easy creation of metamodels - Production of highly performant code in C++ - Near future: support for additional access technologies – JMI and Web Services
platform independent modeling	medini EDOC	- Based on standard modelling vocabulary (EDOC) - wizards and dialogs guide the user through the modelling process - Static structure and behaviour modelling
platform specific modeling	medini Parlay	- Platform specifics can be captured easily using MOF metamodels - Rational Rose and IDL/ODL based modelling support with guidance through modelling process
model transformations	medini MTG	- Transformer skeleton generator, based on early QVT - Processes transformation model - very fast and efficient realization of transformers - forward and backward transformations using the same approach - integrated with medini active repository for consistency management and model change propagation
code generation	medini 4 eclipse	- turns eclipse into a MOF repository for Java - based on Java metamodel, fully supports Java and EJB

5. MDA 4 OSA/Parlay within the open FOKUS 3Gb test bed

5.1. FOKUS 3Gb Testbed and OSA/Parlay Playground

The Fraunhofer FOKUS 3Gb test bed, which has been launched in November 2002, and since October 2003 acts as national host for 3Gb applications development supported by the German Ministry for Education and Research (BMBF), is featuring all the above described state of the art service delivery platforms on top of a broad spectrum of mobile networks in an open manner. I.e., the FOKUS 3Gb network infrastructure comprises besides different local area wireless technologies and WiFi hotspots exclusive access to one cell of the T-Mobile UMTS network as well as access to a satellite network. Also access to ISDN, GSM/GPRS as well as public broadband internet is available.

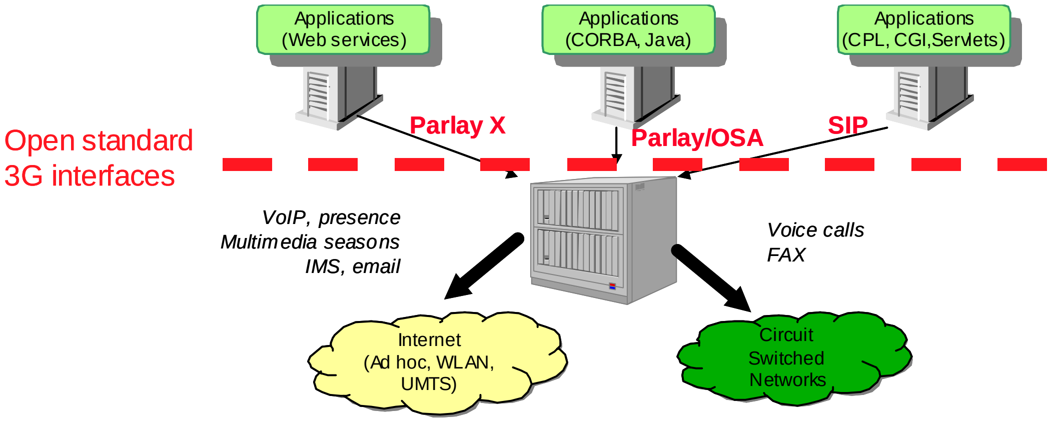

Figure 6 FOKUS 3Gb Service Development and Provision Environments

A major target of the FOKUS 3Gb centre is to provide in contrast to other vendor specific or operator specific technology test beds an open and independent test and development environment, which is open for the use by 3rd parties. This means that content providers and application developers obtain an one stop shopping interface for gaining access to the state of the art in mobile telecommunications technologies to validate their ideas and developments.

The heart of the 3Gb centre forms the 3Gb service platform – the Open Communication Server (OCS) - for value added service provision on top of integrated circuit switched and packet oriented networks, which brings together the current state of the art service platforms, such as SIP application servers (CPL, CGI, servlets), OSA/Parlay gateways (Java, CORBA), and web service (i.e. Parlay X). The main target of this platform is to provide different levels of abstraction form underlying network capabilities and thus different levels of value added service programming within 3Gb environments (see Figure 7).

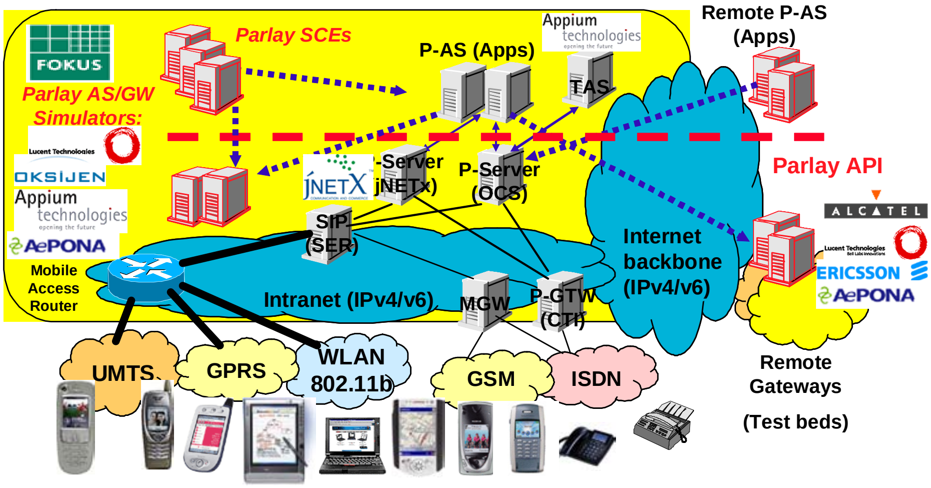

Figure 7 Open OSA/Parlay Play ground @ FOKUS

Content providers and application developers are the prime targeted customers, which could connect their application servers from remote to the FOKUS test bed, or introduce their servers within the test bed or bring their applications on provided application servers.

Based on the experience that obtaining access to 3Gb service platform technology is a challenging task, FOKUS has introduced the concept of so-called “3Gb technology play grounds” in order to focus the development work. Particularly, FOKUS has launched in 2003 the first official open OSA/Parlay play ground [7]. The OSA/Parlay play ground, brings together different OSA/Parlay gateways, gateway simulators, service creation toolkits, as well as demo applications in one single environment. It is an open and independent OSA/Parlay show and education centre to make this enabling technology available to all players of the future open telecommunications market.

5.2. MDA 4 Parlay Usage Scenario

The above described MDA2Parlay toolkit has been successfully deployed in December 2003 with an associated demonstration application. In this section, we shortly outline the usage of the medini environment by a modeller/developer to realize a service on a Parlay platform. We concentrate on a simplified realization process, with no PIM to PIM or PSM to PSM transformations.

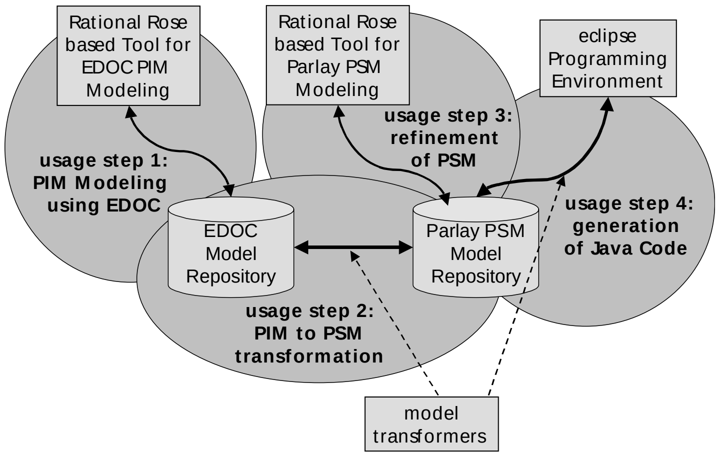

Figure 8 MDA 4 Parlay Usage Scenario

5.2.1. Usage Step 1 - Platform Independent Service Modelling

In a first step, a modeller uses the EDOC framework inside his medini Add-In for Rational Rose to specify a service or a service collaboration in a platform independent manner. The Add-In is online connected with an EDOC model database (also called model repositories) to verify and to store representations of the platform independent model. This connection is to its maximum interactive, meaning that any change of the model inside the Rose tool is immediately communicated to the model database for verification.

If a constraint of the EDOC modelling concepts is being invalidated, this fact together with its position in the model and its reason is being communicated back to the modeller and shown in an according message. After the modeller decides his model to be stable (for a development cycle), this is also being communicated to the model database – the platform independent model is now “frozen”.

5.2.2. Usage Step 2 – Transformation of the PIM into a Parlay PSM

After the platform independent model is being frozen, the developer uses the EDOC to Parlay PSM model transformer to transform the platform independent model into a Parlay platform specific model. Although this step seems to be fully automated, the transformation is being parameterised. The setting of the transformation parameters may occur either in a transformation configuration to allow for batch-processing of transformation steps or can be handled interactively.

In both ways, the transformer receives specific parameters and applies them to the transformation. For each element of the source model (the platform independent model), one or more model elements in a target model (the platform specific model) are instantiated. For each such transformation, the relations between source and target model elements are being stored to allow for traceability. Technically, the PIM to PSM model transformer accesses the platform independent model in the PIM model database, processes the transformation steps defined for it, and stores the resulting target model in the Parlay PSM model database. This resulting target model can now be accessed in the PSM model database, e.g. to express it in a PSM modelling tool.

5.2.3. Refinement of the PSM using Modelling Tool

After the completion of the transformation from the PIM to the PSM, the generated Parlay specific model is stored in the PSM model database. The next step is to refine this platform specific model in a modelling tool. It should be noted, that the transformation does not produce Parlay platform code directly from a platform independent model. The reason is, that all platform specific policies and properties are added to the platform specific model before the final transformation to code. To do this refinements, the modeller/developer uses a graphical UML modelling tool (technically another Add-In for Rational Rose) to develop this platform specific model refinement. He accesses the PSM model database and loads the (generated) PSM model from there. This model is being expressed in the Rational Rose tool, which is connected with the PSM model database and verifies each model change in an interactive manner. After having completed the refinement process, the platform specific model is being stored in the PSM model database as “frozen”.

5.2.4. Generation of Java Code out of the PSM

The last step that is being supported by medini is the transformation from the platform specific model (output of step 5.2.3) to Java code. This Java code is based upon the enago OSP service framework adapted for usage with the Parlay APIs. To do this job, the developer uses another model transformer, that is being defined upon the enago OSP and Parlay metamodels and the Java metamodel. Each transformation step takes a model element of the enago OSP platform specific model and creates one or more Java language constructs, using the eclipse programming environment. Like the transformation from the PIM to the PSM, this transformation is again parameterised, using the same mechanisms of batch-parameters or interactive parameterisation of the transformation. As a result, the generated Java code appears in the eclipse programming environment and can be built to be deployed in a Parlay platform.

6. Summary

In this paper we have illustrated that application of the MDA approach and related tools to OSA/Parlay service delivery platforms is possible and a big step towards providing abstraction from SDP technology changes. MDA provides an extra level of system and network abstraction. As the MDA approach naturally supports the design and extension of real business/enterprise applications, it fits perfect the current trend of network technology abstraction and third party support, envisaged by open network APIs, such as OSA/Parlay. We have proven by prototyping that MDA tools can be successfully used in OSA/Parlay environments.

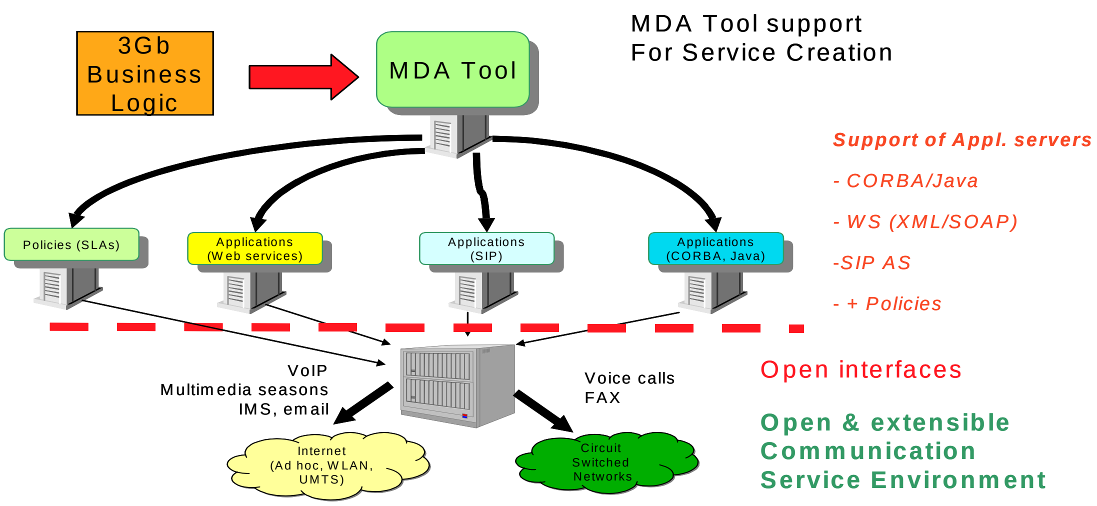

Figure 9 MDA as universal Service Design Tool for different 3Gb SDPs

The next steps of research and development concentrate in two directions:

- extension of the medini toolkit to support different application server technologies such as Web Service platforms and different platform independent modelling techniques, most importantly the UML 2 standard,
- application of MDA onto Parlay X interfaces and application servers and onto emerging SIP based telecommunication environments (e.g. servlet based SIP application servers within the IP-Multimedia Subsytem at FOKUS).

This will enable a smooth transition of existing MDA-based services to be deployed on changing SDPs.

References

[1] Magedanz, T., Popescu-Zeletin, R. (1996a), "Intelligent Networks - Basic Technology, Standards and Evolution",

[2] I. Venieris, F. Zizza, T. Magedanz (Eds.),„Object Oriented Software Technologies in Telecommunications – From Theory to Practice“, ISBN: 0471-6233792, Wiley Publishers UK, April 2000

[3] T. Magedanz, R. Glitho: “Intelligent Networks in the New Millenium“, Feature Topic, IEEE Communications Magazine, USA, Vol.38, No. 6, June 2000

[4] O. Kath, M. Soden, M. Born, T. Ritter, A. Blazarenas, M. Funabashi, C. Hirai: “An Open Modelling Infrastructure Integrating EDOC and CCM”, Proceedings of The 6th IEEE International Enterprise Distributed Object Computing Conference, Brisbane, Australia, Sep. 2003

[5] Parlay home page: www.parlay.org

[6] 3GPP TS 23.127: “Virtual Home Environment / Open Service Architecture”, form 3GPP Website: www.3GPP.org

[7] 3GPP: TS 29.198: “OSA - Application Programming Interface - Part 1“ and TR 29.998 “OSA Application Programming Interface - Part 2“, from 3GPP Website: www.3GPP.org

[8] FOKUS OSA/Parlay Playground home page: http://www.fokus.fraunhofer.de/research/cc/ngni/

[9] Object Management Group. Common Warehouse Metamodel specification, OMG document ad/01-02-01, February 2001.

[10] Object Management Group, MOF 2.0 Query / Views / Transformations RFP, OMG document ad/02-04-10, April 2002.

[11] Object Management Group. MOF 2.0 Core Proposal, OMG document ad/2002-09-14, Sept, 2002.

[12] Object Management Group, Model Driven Architecture (MDA), ormsc/01-07-01, July 2001.

[13] Object Management Group. UML 1.4 with Action Semantics, Final Adopted Specification, OMG document ptc/02-01-09, January 2002.

[14] Extensible Markup Language (XML) 1.0, 2nd edition. W3C Recommendation, 2000, document REC-xml-20001006.

[15] Object Management Group, XML Metadata Interchange (XMI), 2002, OMG document formal/00-11-02

[16] IKV++ Technologies AG: The medini tool chain, http://www.ikv.de/pdf/mediniWhitePaper.pdf

[17] Object Management Group: “Meta Object Facility, Version 1.4“, OMG docu-ment formal/2002-04-03

[18] Object Management Group: “UML Profile for Enterprise Distributed Object Computing Specification”, OMG document ptc/02-02-05

[19] The eclipse Consortium http://www.eclipse.org/