D 5.4.2

Graphical Evolution Tool

WP 5.4

Application Adaptation, Methods / Tools

Dissemination Level: Public

Lead Editors:
Robert Woitsch (BOC)
Margit Schwab (BOC)

December 30, 2005

Status: v1.0 FINAL

SIXTH FRAMEWORK PROGRAMME

PRIORITY IST-2002-2.3.1.18

Grid for complex problem solving

Proposal/Contract no.:004293

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORISED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

a. **"Collective Work"** means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.
b. **"Derivative Work"** means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.
c. **"Licensor"** means the individual or entity that offers the Work under the terms of this License.
d. **"Original Author"** means the individual or entity who created the Work.
e. **"Work"** means the copyrightable work of authorship offered under the terms of this License.
f. **"You"** means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
b. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats, but otherwise you have no rights to make Derivative Works. All rights not expressly granted by Licensor are hereby reserved, including but not limited to the rights set forth in Sections 4(d) and 4(e).

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' ability to exercise the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any reference to such Licensor or the Original Author, as requested.
b. You may not exercise any of the rights granted to You in Section 3 above in any manner that is primarily intended for or directed toward commercial advantage or private monetary compensation. The exchange of the Work for other copyrighted works by means of digital file-sharing or otherwise shall not be considered to be intended for or directed toward commercial advantage or private monetary compensation, provided there is no payment of any monetary compensation in connection with the exchange of copyrighted works.
c. If you distribute, publicly display, publicly perform, or publicly digitally perform the Work, You must keep intact all copyright notices for the Work and give the Original Author credit reasonable to the medium or means You are utilizing by conveying the name (or pseudonym if applicable) of the Original Author if supplied; the title of the Work if supplied; and to the extent reasonably practicable, the Uniform Resource Identifier, if any, that the Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work. Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.
d. For the avoidance of doubt, where the Work is a musical composition:
    i. **Performance Royalties Under Blanket Licenses**. Licensor reserves the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work if that performance is primarily intended for or directed toward commercial advantage or private monetary compensation.
    ii. **Mechanical Rights and Statutory Royalties**. Licensor reserves the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created

by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions), if Your distribution of such cover version is primarily intended for or directed toward commercial advantage or private monetary compensation.

e. **Webcasting Rights and Statutory Royalties.** For the avoidance of doubt, where the Work is a sound recording, the Licensor reserves the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions), if Your public digital performance is primarily intended for or directed toward commercial advantage or private monetary compensation.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED BY THE PARTIES IN WRITING, THE LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESSED, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OR ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL THE LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF THE LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.
b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, the Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
c. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
d. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. The Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

Context

Activity 5	Integration and Application Case Studies
WP 5.4	Application Adaptation, Methods & Tools
Task 5.4.3	Tool Development
Dependencies	This deliverable requires input from WP 4.3, WP 4.4 and WP 5.1 and provides output for WP 4.4, WP 5.1 and WP 5.2 as well as to the tasks 5.4.4.

Contributors:	***Reviewers:***
Robert Woitsch (BOC) Margit Schwab (BOC) Wilfrid Utz (BOC)	Robert Piotter (USTUT) Nuno Inácio (IT-Aveiro) Burkhard Stiller (UniZH) Josep Martrat (ATOS)

Approved by: Robert Woitsch, Austria as WP5.4 Head

Version	Date	Authors	Sections Affected
0.0	November 11, 2005	Robert Woitsch	All – initial document
0.1	November 22, 2005	Robert Woitsch, Wilfrid Utz	All – integration of all sections
0.3	November 25, 2005	Robert Woitsch, Wilfrid Utz	Chapter 5,6,7
0.4	December 05, 2005	Robert Woitsch, Wilfrid Utz	All, final format, screenshots from the tool
0.5	December 19, 2005	Burkard Stiller, Nuno Inacio, Robert Piotter	All, review and comments of all chapters
0.6	December 22, 2005	Robert Woitsch	All, Consistency Check with D 5.4.1,
0.7	December 30, 2005	Robert Woitsch, Margit Schwab, Wilfrid Utz	Spell Checks, Consistency Check
1.0	December 30, 2005	Robert Woitsch, Wilfrid Utz	Finalization

Executive Summary

The roadmap to Akogrimo convergence is a model oriented assistance system that guides developers from traditional approaches to use the innovative Akogrimo architecture. The goal is therefore to support developers who want to develop services or applications on top of the Akogrimo platform. The Roadmap distinguishes between:

- the conceptual support that provides a method and the according tool to be applied by the developer and
- the knowledge support that provides knowledge from partners in form of a graphical representation.

This document introduces the unique Graphical Evolution Tool[1] (GET) that supports the roadmap to Akogrimo convergence by providing a complex Web-Modelling environment that implements three different graphical modelling methods to encourage the developer to analyse its application or the service to be developed in a wider context.

The Web-Modelling environment provides Web-Components to design (Web-Modeller), to distribute (Web-Documentation), to use (Web-Import / Export) and to document (Web-Repository) the so-called Akogrimo Roadmap. This new Web-Modelling environment is integrated with stand alone model editors to enable the usage of the Akogrimo Roadmap optionally in a stand alone or in a Web-Modelling environment. This sophisticated integration makes the GET a coherent Web-Modelling environment.

The GET supports the new AKBPM, the well-known UML™ and the PROMOTE® modelling methods, which are required for the Akogrimo Roadmap. This complex integration of three different modelling methods in the same modelling environment is technically achieved by using the meta modelling paradigm and conceptually achieved by synchronizing the three modelling methods. The technical and conceptual synchronization makes the GET a unique modelling environment that integrates requirement specification and knowledge support.

The GET encourages the developer to think and analyse the service requirement in much wider context. The provision of a very broad modelling framework that is integrated with concrete knowledge sources makes GET a unique knowledge and development platform.

Two different usage scenarios of the Akogrimo Roadmap are supported by the GET. The application development scenario is concerned to support the implementation of an application on top of the Akogrimo platform and the second one is concerned to support the implementation of services in Akogrimo. The GET uses a simple user concept to provide a set of user specific features and accesses to relevant knowledge.

The Web-Modeller, Web-Documentation, Web-Import/Export and Web-Document Repository have been implemented or adapted. Four databases (one for each modelling method and one for the repository) and nine Web-Services (three for each modelling method to enable load balancing and failure resistance) are installed at one physical machine at BOCs site for the first version of the GET.

The GET is the most sophisticated high level Web-Modelling environment for Service-Oriented Software Development.

---

[1] Url: http://83.65.190.82:8080/, User:AkogrimoRoadmap, Password: AkogrimoRoadmap-FP6-2003-IST-2-004293

Table of Content

List of Figures

List of Tables

Annex: List of Abbreviations

ADL	Adonis Definition Language (Model exchange format for ADONIS®)
AKBPM	Akogrimo Business Process Management (Method)
BPEL	Business Process Execution Language
BSCW	Basic Support for Cooperative Work
D	Deliverable
EPC	Event Driven Process Chain (Method)
GET	Graphical Evolution Tool
HTML	Hyper-Text Markup Language
ID	Internal Deliverable
JSP	Java Server Pages
MSDE	Microsoft Data Engine
PROMOTE®	Process Oriented Methods and Tools for Knowledge Management (Method)
SQL	Structured Query Language
UML	Unified Modelling Language
URL	Uniform Resource Locator
WB	Workbench
WD	Web-Documentation
WDR	Web-Document Repository
WM	Web-Modeller
XMI	XML Meta Data Interchange
XML	eXtended Markup Language
XSLT	eXtensible Stylesheet Language: Transformations

1 Introduction and Problem Statement

One of the key success factors of the Akogrimo platform is the applicability to develop applications on top of it and usability to integrate or modify services.

The new Akogrimo Roadmap approach supports the development of applications on top of Akogrimo in the so-called Application Development Scenario, which is described in D 5.4.1 (Roadmap to Akogrimo Convergence) [1] that has been generated in parallel with this deliverable.

The Graphical Evolution Tool (GET) supports the Akogrimo Roadmap by providing a process-, model- and knowledge oriented tool integrated in a Web-Platform to enable a collaborative usage of the Akogrimo Roadmap by distributed users.

This distributed Web-Platform is a unique configuration of a set of tools, concerned with modelling, distributing, using and documenting the Akogrimo Roadmap. This tool set is configured to enable collaborative usage of the Akogrimo Roadmap by providing a distributed Web-Modelling environment and necessary Web-Components in combination with individual installations of stand alone model editors.

The Akogrimo Roadmap has been developed using three different graphical modelling methods concerned with:

- the requirement definition using a process-oriented approach (AKBPM-Method),
- the specification and design of software using an object-oriented approach (UML™) and
- the implementation of services using a knowledge-oriented (PROMOTE®) approach.

Figure 1 introduces an overview of the Akogrimo Roadmap that is explained in more detail in D 5.4.1. (Roadmap to Akogrimo Convergence).

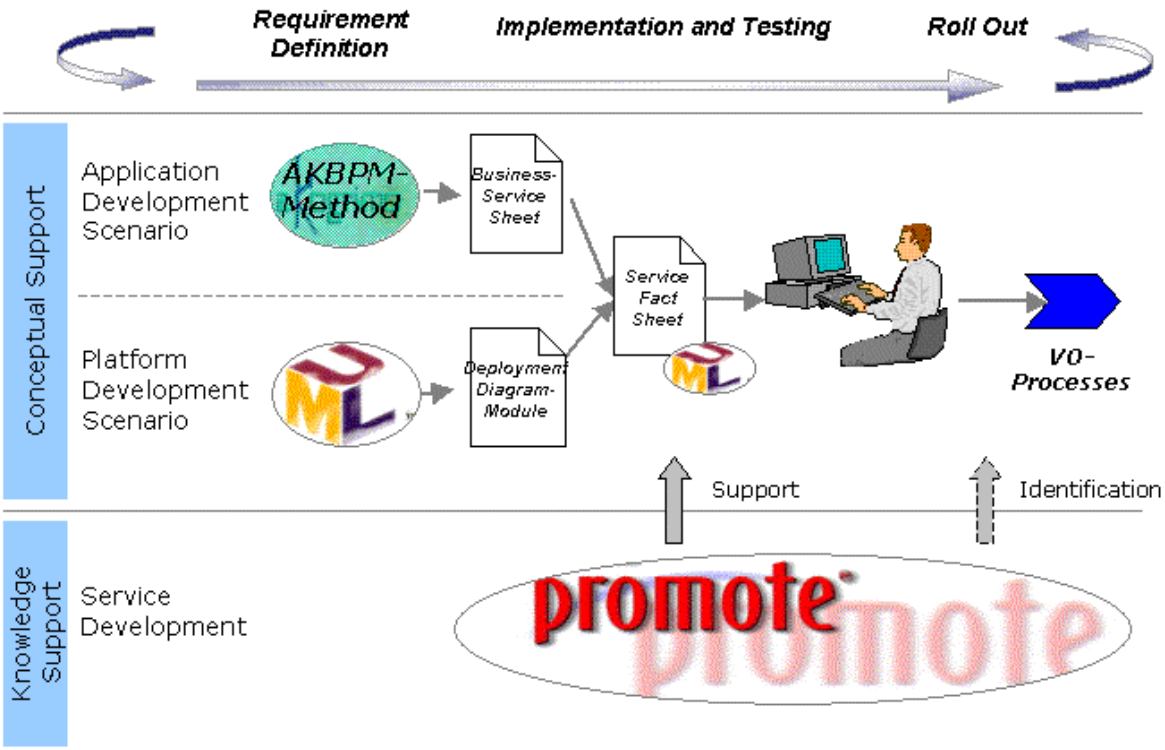


Figure 1 The Akogrimo Roadmap Overview

Figure 1 identifies the different support of the application or service implementation. As described in D 5.4.1 (Roadmap to Akogrimo Convergence), the different phases have different support by the Akogrimo Roadmap. The development phases[2] below provide a framework to classify the different types of roadmap support and separate the three modelling methods, but leave enough flexibility to allow each developer to use its own development procedure (such as agile programming, rapid prototyping or incremental approach).

The **Requirement Definition Phase for applications** is conceptually supported by providing a new method called "Akogrimo Business Process Management" (AKBPM) Method. In this context the term "conceptually supported" means that the AKBPM provides special graphical model types to define the requirements of an application scenario. It can be applied for complex and heterogeneous application scenarios to derive service requirements in form of so-called "Business Service Sheets", which are text documents describing the behaviour of the service.

The **Requirement Definition Phase for platforms** is also conceptually supported by providing the well-known method UML™. Similar to the scenario above the term "conceptually supported" means the provision of UML™ to describe the platform and the services.

The **Implementation and Testing Phase** starts with the definition of the Service Fact Sheet. This is a technical description of the service interfaces and a documentation of the service implementation. This Service Fact Sheet is a text document that is generated at the beginning of the development phase and adapted on defined states of the development process. The Akogrimo Roadmap supports this phase by providing knowledge for the service development within Akogrimo. The modelling method PROMOTE® has been selected to describe the experience of partners with the implementation of Akogrimo services.

The **Roll Out Phase** is supported by identifying the VO-Processes, which have to be considered when registering the new service.

The goal of the Graphical Evolution Tool (GET) is to support the above introduced Akogrimo Roadmap by developing, collecting and configuring a set of tools. Two scenarios describing how the Akogrimo Roadmap supports the development process have been proposed. Based on the scenario proposal a feature set for the GET has been derived, that groups so-called Web-Components. Some of the Web-Components have either been implemented for Akogrimo or have been specially configured to enable a complete support of the Akogrimo Roadmap. The main challenges of the GET are:

1. The provision of a sophisticated Web-Modelling Platform in conjunction with a stand alone modelling environment and the integration of external platforms through interfaces to enable online access to a wide collection of Akogrimo relevant graphical models.
2. The implementation of a modelling environment that provides all three graphical modelling methods - AKBPM, UML™, PROMOTE® - for the Akogrimo Roadmap.
3. The provision of a number of different concepts and approaches to encourage the developer to think about the broader process and support various phases and levels of the development process such as conceptual support for requirement specification and knowledge support for service development.

The following sections provide an overview of the Akogrimo Roadmap and the goals within the Workpackage similar to the introduction of D 5.4.1 (Roadmap to Akogrimo Convergence) to introduce the main ideas of this Workpackage and the overall concept of the Akogrimo Roadmap.

---

[2] The following chapter is based on expert interviews of partners as well as on literature compare [2], [3], [4]

1.1 The Overview of the Roadmap

The two deliverables D 5.4.1 (Roadmap to Akogrimo convergence) and D 5.4.2 (Graphical Evolution Tool) are closely related and written in parallel. This section introduces the different focus of the two documents and shows their relationship.

This document introduces the Graphical Evolution Tool that supports the Akogrimo Roadmap. It is a technically oriented document that describes the proposed usage-scenario of the roadmap and introduces the implemented tools. It focuses on:

- the proposal of a usage scenario to apply the Akogrimo Roadmap,
- the proposal of user roles to group the required access rights to the tools,
- to introduce the Web-Components of the Graphical Evolution Tool that provide the functionality for the above proposed usage scenario and
- to discuss the technical infrastructure and the overall architecture of the Web-Modelling environment for Akogrimo.

In comparison to this focus the closely related D 5.4.1 (Roadmap to Akogrimo convergence) is concerned with:

- the explanation of the new concept Akogrimo Roadmap,
- the analysis for enabling applications to use Akogrimo and
- the introduction of graphical modelling methods that are used to provide a graphical representation of guidelines for service development and application adaptations.

There are three conceptual layers in this workpackage as indicated in Figure 2. The content layer deals with the knowledge about Akogrimo relevant topics, the method layer is concerned with graphical modelling methods for the Akogrimo Roadmap and the technology layer provides the necessary technological infrastructure to support the Akogrimo Roadmap.

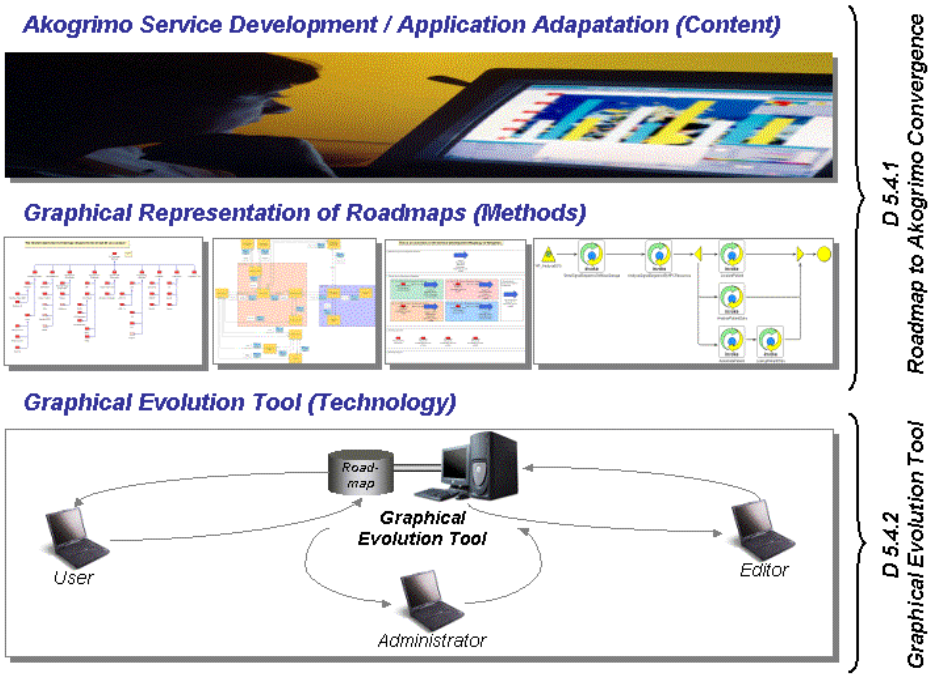


Figure 2 Identification of Deliverables in WP 5.4

Parts of the introduction are overlapping in both deliverables because of the close relationship.

1.2 Goal of Workpackage 5.4

This section describes the objectives of the workpackage and identifies the relation to the deliverables in more detail. This deliverable describes the Roadmap to Akogrimo convergence by identifying the gap between traditional architecture and the innovative Akogrimo approach [5]. The described Roadmap to Akogrimo convergence is supported by a set of graphical modelling methods introduced in this deliverable and a distributed Web-Modelling environment that provides software to model the roadmap and to establish links to relevant documents described in the related deliverable D 5.4.2.

The ***first milestone*** of this workpackage is the definition of the roadmap to Akogrimo convergence, whereas the roadmap is seen as a graphical representation of procedures that describe the implementation, adaptation or usage of the Akogrimo platform. This requires the analysis of the Akogrimo convergence. Based on that analysis a usage concept was developed that defines how the roadmap should be used and which aspects of Akogrimo should be identified. The appropriate graphical modelling methods for the relevant aspects in Akogrimo have been selected to generate the Akogrimo Roadmap considering syntax, semantics and notation. Partners described their individual procedures by questionnaires, interviews and workshops to finally aggregated a reference roadmap to Akogrimo convergence.

The ***second milestone*** of this workpackage is the first version of the Graphical Evolution Tool that is seen as a distributed Web-Modelling environment that generates graphical representations of roadmaps according to the selected modelling methods. The components of the Web-Modelling environment have been identifies based on an identified usage scenarios of the roadmap.

Input into the workpackage was provided by

- the technical annex of Akogrimo (FP6-2003-IST-2) defining the general objectives,
- the expert knowledge of the development of application layer in 4.3 and 4.4 partners for the gap analysis of Akogrimo,
- deliverables from the workpackage 4.3, 4.4, 3.1 for technical input and deliverables from workpackages 2.1, 6.3 and 3.2 for business relevant input and the demonstration scenario and
- experience in modelling method development and method integration to integrate the above content into an applicable roadmap.

The ***procedure during the workpackage*** had been split into two phases with three tasks. The first phase was concerned with setting up the workpackge, identifying usage scenarios, collecting modelling methods, specifying the components of the Graphical Evolution Tool, and analysing the approaches for enabling applications to use Akogrimo. The result of the first phase was the internal deliverable ID 5.4.1 (The Initial Roadmap to Akogrimo convergence) [6].

The second phase was concerned with the modelling of the individual development processes of the workpackage partners ATOS, CCLRC, CRMPA and NTUA using questionnaires, telephone interviews and workshops. Based on these results the reference roadmap as an aggregation of the individual development processes has been modelled.

The eHealth scenario was modelled based on D 3.2.2 (The Akogrimo Business Modelling Framework) [7] and based on reviews with UHOH to demonstrate the procedure in specifying application requirements.

The partners of workpackage 5.1 have been supported by their consolidation of UML™ – models to demonstrate the documentation of the platform requirements.

The ***output of the workpackage*** is:

- the first version of the Akogrimo Roadmap in the form of graphical models that are described in this deliverables,
- an integrated set of graphical modelling methods that enables the modelling of the Akogrimo Roadmap described in this deliverable and
- the first version of the Graphical Evolution Tool in the form of a distributed Web-Modelling environment.

Beside the above results there are the following side effects:

- a common understanding of development procedures and terms in Akogrimo for the application relevant layers (4.4. and 4.3).
- a structured development procedure for the implementation of the services and
- the definition of relevant aspects in Akogrimo based on intensive discussions during the modelling workshops.

The overall architecture of the Akogrimo Roadmap is based on the PROMOTE® [8] approach for the installation of a knowledge management assistant system that distinguishes between:

1. The Content Layer:
   First, the so-called knowledge application layer is analysed to identify the user, their requirements and the content. The content layer includes roadmaps, guidelines, documents and link collections that have been developed or collected based on the partners' experience in workpackage 4.3 and 4.4.
2. The Method Layer:
   Second, the so-called organisational memory layer is specified to represent the applied modelling methods to enable the documentation of partners knowledge. Three methods have been selected to graphically represent and formalise the content.
3. The Tool Layer:
   Third, the technology layer provides the infrastructure to run the above mentioned concepts. The Graphical Evolution Tool enables the design, distribution, usage and documentation of models.

Input	Layer	Output
Experience of 4.3, 4.4 and 5.1 partners.	Content	The Akogrimo Roadmap: • Application Specification support, • Platform Documentation support and • Service Development support
Experience in modelling methods as well as requirements raised in WP 4.3, 4.4 and 5.1.	Method	• Akogrimo Business Process Management Method (AKBPM), • UML™ and • PROMOTE®
Experience in modelling tools as well as requirements raised in WP 4.3, 4.4 and 5.1.	Tool	The Graphic Evolution Tool as a web-based modelling environment.

Table 1 Overview of the workpackages' input/output

Table 1 gives an overview of the workpackage input and output.

1.3 Deliverable Overview

This deliverable is concerned with the Graphical Evolution Tool (GET) that enables the use of the roadmap to Akogrimo convergence consiting of a set of graphical models, documents and guidelines. The roadmap is published in the Internet via the GET, whereas the theoretical concept behind the roadmaps and the selected methods are described in the related D 5.4.1 (Roadmap to Akogrimo Convergence).

Both deliverables are based on the ID 5.4.1 (The Initial Roadmap to Akogrimo Convergence) that introduces the initial analyses for enabling applications to use Akogrimo, describes the used modelling methods, points out the initial setup of the GET as well as shows initial roadmap models.

Chapter 1 introduces the main objectives and milestones of workpackage 5.4 (Application Adaptation, Methods & Tools) to address key questions that have to be answered by this document. There is also an introduction of the Akogrimo Roadmap that is described in more detail in D 5.4.1 (Roadmap to Akogrimo Convergence).

Chapter 2 is concerned with the identification and description of two usage scenarios that are supported by the GET. These scenarios demonstrate the support by the GET for top-level application requirement analysis to technical review of service developments. Furthermore the simple Akogrimo Roadmap user roles conceptis identified.

Chapter 3 assigns the relevant methods and the applied content to the scenario actions defined in Chapter 2 to analyse the scenarios in more detail. Subsequently the GET Components are briefly depicted and the selected methods that are described in D 5.4.1 (Roadmap to Akogrimo convergence) are mapped with the Web-Components of the tool to identify the tool support of the different modelling methods. Furthermore the user role configuration will be presented, which maps the access rights of the Akogrimo user role concept to the Web-Components from the GET.

Chapter 4 briefly describes the overall architecture of the GET in order to provide an insight of the setup and indicates required adaptations that had to be undertaken for the Akogrimo project. This chapter provides a description of the functionality, an overview of the technology and the necessary implementation for each component. It is structured according to the architectural layers of the GET.

Chapter 5 introduces the IT-Infrastructure, which has been set up to provide the roadmap. Possible future adjustments e.g. due to increased workload, are outlined.

Chapter 6 shows a set of screenshots that indicate how the GET may be used within the Akogrimo project. The GET consists of the Workbench, the Web-Documentation, the Web-Modeller, the Web-Import/Export and the Web-Document Repository. Each of this Web-Components is briefly introduced by describing sample actions of users and by providing screenshots for visualisation purposes. This chapter is thought to guide the user through the functionalities of the GET.

Chapter 7 finally summarizes this deliverable providing expectations how the tool will be used in Akogrimo and listing the key objectives of the Graphical Evolution Tool.

2 Usage Scenario Description

The Graphical Evolution Tool (GET) is an assistant system that supports developers to use the Akogrimo Roadmap. The requirement specification for the GET is based on three inputs:

- user requirements have been derived based on proposed usage scenarios of the Akogrimo Roadmap,
- the methodological requirements that are raised by using the model-oriented approach and
- the technological requirements based on the experience of modelling tool development.

The **user requirements** have been derived based on expected usage scenarios of the roadmap supporting the development of an application to run on top of Akogrimo. The tool has to support various development phases such as requirement specification, implementation or testing. Based on the proposed scenarios requirements such as the export of UML™ models in XMI format, or the integration of BSCW documents have been derived.

The **methodological requirements** have been identified according to the used modelling methods. The tool deals with graphical models that consist of model types, modelling objects and attributes. Based on the model-oriented approach requirements such as attribute filter, model type filter have been derived. The meta-model paradigm was used to enable the provision of three different modelling methods in the same tool.

The **technological requirements** have been collected on the current state of development and the experience in the development of modelling environments. Requirements such as read and write access via Web-Services or the integration of Web-Modeller and a standalone modeller have been derived by technological observation.

A rapid prototype approach has been selected to collect the requirements of the GET. First a prototype of different stand alone model editors providing the different modelling methods and interfaces have been built and introduced to the consortium. Based on the feedback an updated version of the prototype were built.

The identification of the modelling methods and the first prototype of the stand alone model editor were the basis to collect requirements for the Web-Modelling environment. The collection of requirements had three phases.

The first phase was a rapid prototype for internal demonstration in the development infrastructure. A complete modelling scenario was defined and test users were testing the tool according the defined modelling scenario. The result was a first test report consisting of failure report, a collection of invalid models, a collection of invalid parts of the methods, server side error logs and client side error logs.

The second phase was an updated prototype for internal demonstration in a special demonstration infrastructure. Internal reviews had been performed by a group of BOC reviewers to individually test the Web-Components. The result was a set of internal review reports, which had been combined to a requirement list.

The third phase was the implementation of a first version for external demonstration using the "life" IT-Infrastructure described in chapter 5. This version was used in the workpackage with the partners. The partner feedback was continuously updated.

The following section introduces the basis concepts of the GET and the two scenarios that are proposed.

2.1 Basic Concepts of GET

The GET is a Web-Modelling environment that enables the use of the Akogrimo Roadmap. In the following the three principles of the GET process-orientation, modelling and knowledge platform are introduced.

1. **Process-orientation:**

   The process-orientation is seen as a reasonable approach to specify requirements by analysing the business processes of the application scenario. In Akogrimo this approach is provided to analyse the application scenario of Akogrimo like the eHealth scenario using business processes and to derive service requirements from the business processes.

   This approach encourages the developer to specify the wider context of the services and define the overall usage concept of the application.

2. **Modelling Platform:**

   Successfully applied modelling approaches have two results. The first result is the graphical model that enables a simplified view of complex interdependencies. The second result is a common understanding and an exact definition of the modelled terms among the participating partners.

   The GET is a modelling platform that integrates three different modelling methods - each of them with a different focus - to provide a simplified view via graphical models and thereby supporting the generation of a common understanding.

3. **Knowledge Platform:**

   Graphical models are simplified aggregations of a complex so-called knowledge objects that can be used to gain an overview and enable navigation within the knowledge object. The Akogrimo platform is seen as such a knowledge-object where the knowledge is stored mainly in the head of the project partners, in the project space BSCW, the homepage, the Roadmap Repository and in the Internet.

   The graphical models of the knowledge method PROMOTE® enable navigation within the Akogrimo content and make expert knowledge explicit via the representation of expert procedures in form of processes.

The GET is therefore seen as a knowledge platform that enables the navigation and usage of Akogrimo content in a process-oriented way where models are used for a simplified representation of the content.

Two usage scenarios of the Akogrimo Roadmap have been proposed based on the methods in D5.4.1 (Roadmap to Akogrimo Convergence):

The first scenario is the **Application Development Scenario** that is supported by providing the AKBPM (Akogrimo Business Process Management) method to first specify the requirements of the application and specify the services. The service development is supported by providing knowledge described with PROMOTE®.

The second scenario is the **Platform Development Scenario** that is supported by providing the UML™ modelling method to first specify the requirements of the platform and link the service specification with the service development using the Service Fact Sheets.

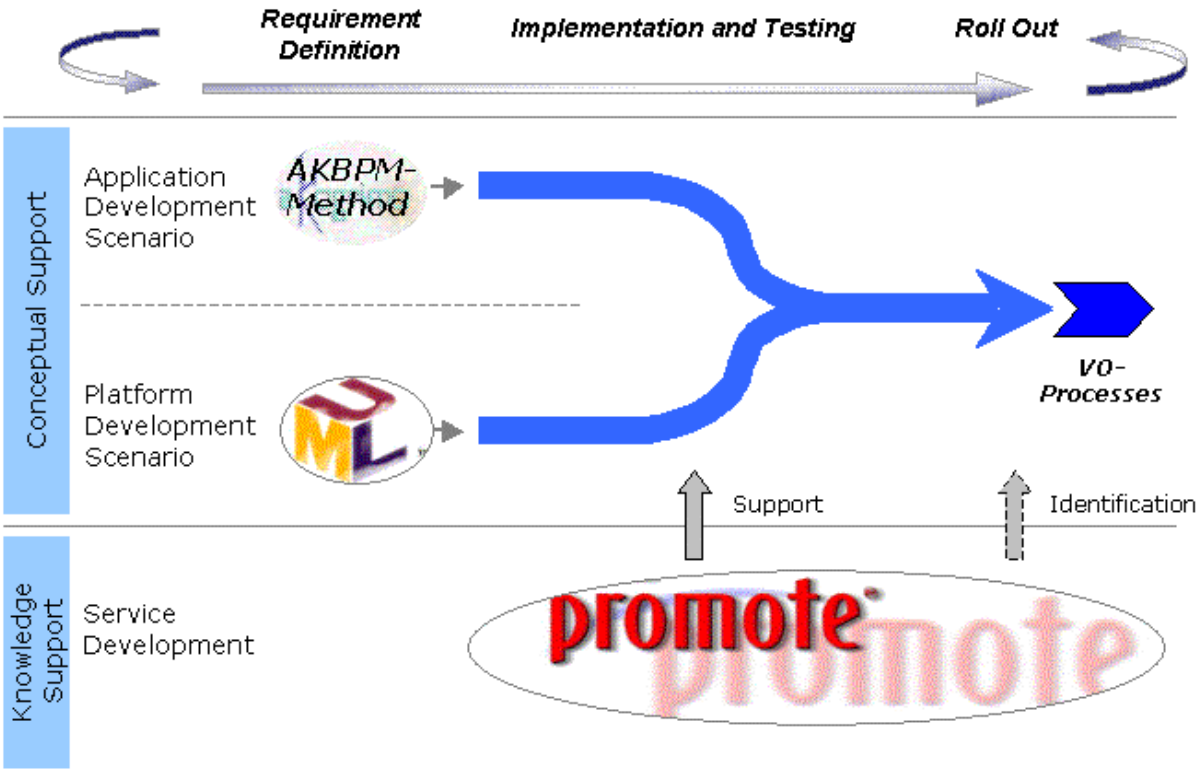


Figure 3 Method Support for Application and Platform Development Scneario

Figure 3 indicates the different methods that are provided for the two development scenarios.

In case an application is developed to run on top of Akogrimo the AKBPM method enables the description of the application use case scenario on a business layer and the specification of services and workflows on a technical layer. The description of the technical layer can be used to start the service development supported by the PROMOTE® method using UML™ Use Cases or Service Fact Sheets to synchronise service requirement specification and service development.

In case infrastructure services have to be configured or developed to better support an application the UML™ modelling method is provided to specify the platform. The deployment models or the Use Cases of UML™ can then be used to synchronise the service development.

The two scenarios described above have been proposed to derive a set of requirements for the first version of the GET.

It is expected that the typical Akogrimo Roadmap user only uses parts of the Akogrimo Roadmap, based on the individual need for support. To simplify the explanation and point out the requirement specification it is assumed that the user of the Akogrimo Roadmap will follow the two scenarios sequentially.

2.2 Akogrimo User Roles

This section introduces a simple user role concept for the Akogrimo Roadmap that was found useful to distinguish the features of the tool according the proposed scenario of the Akogrimo Roadmap. The user roles are used to configure the access rights to the tool to provide each role the necessary tools and components and hide the not relevant features.

Table 2 presents the used roles and provides a brief description. This simple user role concept is thought as a starting point for the Akogrimo Roadmap, if necessary the access rights can be changed as well as new roles can be added.

Roles	Description
Application Provider	The Application Provider is seen as an interface to external customer. He configures a domain specific application and offers it in the external market. The Application Provider is also a Service Consumer, as he consumes the services that are integrated into the application.
Platform Provider	The Platform Provider is also an interface to external customers who configures an Akogrimo platform and provides it to an operator.
Application Configurator / Platform Configurator	The Application / Platform Configurator selects appropriate services and specifies workflows. The task is to map the service specification in form of Business Service Sheets, Use Case Diagrams or Deployment Diagrams with already available Akogrimo services or by defining a Service Fact Sheet on a detailed technical level in case a new service has to be implemented.
Service Developer	The Service Developer is a software developer that gets a service requirement description in form of a Service Fact Sheet and implements the according service.
Service Development Consultant	A Service Developer Consultant is an Akogrimo core partner that is well aware of the Akogrimo architecture and history to provide expert knowledge in a specific domains. The aim of the Akogrimo Roadmap is to externalize the expert knowledge of the Service Developer Consultant using the knowledge management method PROMOTE® to make it accessible.
Guest	A Guest viewer has access rights to view all public models but cannot upload nor download any information.

Table 2 Standard Role Descriptions

It may be interesting to note that in the current usage scenario within Akogrimo from the early release of the GET, there have been two user groups identified based on the workpackages.

- **WP 5.4** has two user groups, one has access to all roadmaps and to the PROMOTE® Web-Modeller, and one user group can additionally upload and download the roadmaps.
- **WP 5.1** has two generic user groups, one has access to all roadmaps and to the UML™ Web-Modeler, and one user group can additionally upload and download the UML™ models.

Beside the generic user groups there are access rights to models per partner, so the partners can only change their own models.

2.3 Application Development Scenario

The GET proposes a usage scenario for the development of applications by indicating a set of actions supporting the implementation of an application to run on top of Akogrimo. This scenario is a proposal to identify relevant actions of Akogrimo Roadmap users. Each action was then analysed according to its knowledge transfer. Explicit knowledge transfers across different roles was identified as actions to be supported by GET.

The proposed usage scenario supports a distributed development of an application, but it has to be pointed out that it is not mandatory to follow the procedure exactly in the indicated way. This usage scenario has the aim to collect the expected requirements GET.

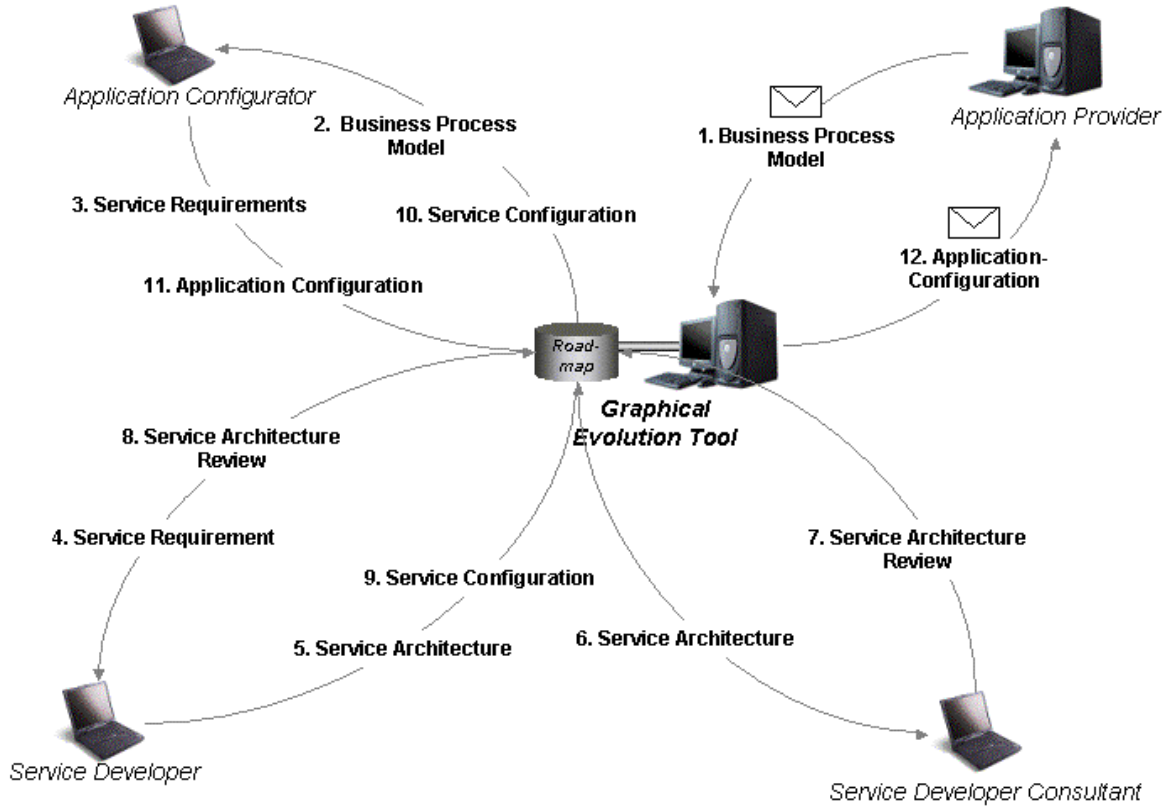


Figure 4 A sample scenario using the Graphical Evolution Tool

Figure 4 depicts the proposed usage scenario identifying different knowledge transfers and development roles for the development of an application to run on top of Akogrimo.

The usage scenario proposes an Application Provider building an application for a specific domain and starting the requirement analysis. For the technical composition of the application the Application Composer is described whose task is to specify the application by composing a set of services.

During the technical composition and testing there may be some services that have to be additionally implemented by the Service Developer. The Service Developer has to describe the service in such a way, that another partner can view, review or test the services.

The Service Developer Consultant reviews and supports the Service Developer .

2.4 Platform Development Scenario

The Platform Development Scenario is similar to the proposed Application Development Scenario. Actions have been analysed according to their knowledge transfer, and identified if explicit knowledge transfer via the GET is useful.

The proposed usage scenario supports the distributed development of application relevant services for the Akogrimo platform. Similar to the Application Development Scenario the proposed development scenario is used to identify GET features but it is not mandatory to follow the procedure exactly in the indicated way.

This usage scenario points out the differences to the previously introduced usage scenario.

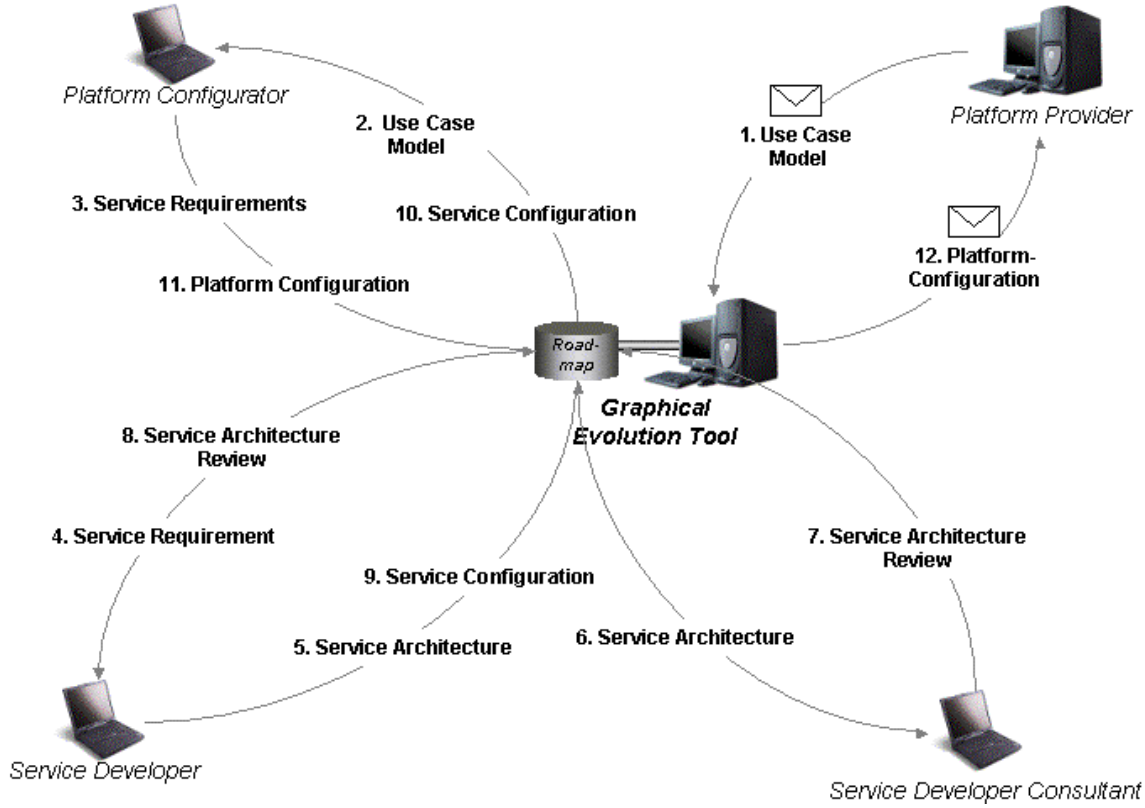


Figure 5 A sample scenario using the Graphical Evolution Tool

Figure 5 depicts the usage scenario that identifies different knowledge transfers and development roles for the development of application relevant services in the Akogrimo platform.

The usage scenario for developing a platform is similar to the usage scenario of application development.

Based on the observation in the project, it is assumed that the requirement specification of Akogrimo will be modelled in UML™. The main difference between the application scenario and the platform scenario is therefore the integration of UML™ models instead of the AKBPM model.

2.5 Relevant Scenario-Actions

This section describes relevant actions of Akogrimo users of the above scenarios.

Action	Scenario Description
1	**Business Process Model / Use Case generation:** An Application Provider develops a Business Model to enter the market in a specific domain (e.g. eHealth). After successfully planning the Business Model the relevant Business Processes that enable the Business Model are identified. The scenario relevant processes as well as the requirement description are uploaded into the GET for further technical development. The Use Case diagram or Deployment diagram of a Platform Developer is also seen as a requirement specification of a service. The "envelop" symbol in Figure 4 and Figure 5 indicates that business process models may be performed on a stand alone model editor and the results are imported into the Web-Modelling platform.
2	**Business Process Model / Use Case usage:** The relevant business processes are analysed, and the service requirements are modelled in more detail to compose an application. Then the according services are selected and configured using Workflow models. Similar to the Application Configurator the Platform Configurator uses the UML™ diagrams to configure a complex infrastructure service.
3	**Service Requirements generation:** In case additional services have to be implemented, a more detailed service description using UML™ and additional documents such as the Business Service Sheets are required. This action requires the interaction between the AKBPM method and UML™ and the interaction between UML™ models.
4	**Service Requirement usage:** The service requirements are published via UML™ and in form of Service Fact Sheets can be viewed by the Service Developer. In case the Service Developer is interested in developing the service, all the requirement specification models can be downloaded to trigger the service development supported by PROMOTE®.
5	**Service Architecture generation:** The implementation of the service is proposed to be started with the Service Fact Sheet that consists of a design chapter. UML™ diagrams can be used to additionally describe the service design.
6	**Service Architecture usage:** The Service Fact Sheet and the detailed description of the service architecture can be used for evaluation. In case a Service Development Consultant – an Akogrimo core partner – is contacted to review the service architecture, the consultant can access all models and documents relevant for the review.
7	**Service Architecture Review generation:** The review, testing or re-use reports are also published via UML™ models and via additional documents in the document repository.
8	**Service Architecture Review usage:** To support the actual service development the UML™ descriptions are available in XMI format for download into the development environment.

9	**Service Configuration generation:** After successfully finishing the service development the service can be again sent to a reviewer for testing or evaluation. In this case the action 5-8 will be performed again.
10	**Service Configuration usage:** The published and deployed services can be used for aggregated services, for workflows or for composed applications.
11	**Application Configuration generation / Platform Configuration generation:** When all services have been developed for the composed application, it may be necessary to define workflows that are related to the business processes or that are parts of the application. The requirements of this action are therefore to enable workflow modelling and to provide an export in BPEL format.
12	**Application Configuration usage / Platform Configuration usage :** Finally the Application Provider can download the whole configuration and import it into the local environment. This includes the download of: • complete AKBPM models with all enhancements made during technical process specifications, workflow modelling and service implementation, • viewing all necessary UML™ models and • downloading the required documents such as the evaluation or test reports from the repository.

Table 3 Application / Platform Development Scenario Action List

Table 3 describes the different actions with explicit knowledge transfer that should be supported by the GET.

The above proposed development scenarios and the derivation of actions are seen as the starting point to identify the GET requirements and the configuration of the tool in more detail.

The next chapter analyses the above mentioned actions in more detail to define the access rights for each Akogrimo user role and to configure the GET.

3 Scenario Configuration

This chapter describes the scenario based requirement analysis of the GET and discusses the relationship between the tool components and the modelling method.

3.1 Method and Feature Mapping

The following table (Table 4) indicates tool requirements that have been deduced from the action lists of the usage scenarios. Additionally the methods and the applied content are indicated for the configuration of the GET.

D 5.4.1 (Roadmap to Akogrimo convergence) describes the selection of appropriate modelling methods. In this section a brief overview on the used methods is provided.

The **Akogrimo Business Process Management (AKBPM)** Method is a new modelling method that has been built especially for the requirements of Akogrimo. In the following this method is referred as AKBPM. It enables:

- a guideline to analyse the business processes based on business models,
- compose technical processes and workflows,
- identify required services and
- identify new services, in case they are not available.

The **Unified Modelling Language (UML™)** is a standardised graphical modelling method to specify system artefacts [9], [10], [11], [12]. In Akogrimo UML 2.0 is used to enable the documentation of the platform and the specification of services.

The challenge was to integrate the UML™ diagrams that have been designed in various workpackages into the Akogrimo roadmap and to enable a homogeneous linkage between the application specification, the experience of the Service Developer and the documentation of the software. It enables:

- the documentation of critical software aspects,
- the description of recommended patterns,
- the explanation of component or platform usage and
- the requirement specification of applications and the platform.

PROMOTE® is a process-oriented knowledge management modelling method developed by BOC in the EU-Project PROMOTE [8], [13], [14], [15]. It is designed to describe the knowledge transfer between knowledge workers in a process-oriented manner.

In Akogrimo this modelling method is used to make the implicit experience of service development consultants explicit in a process oriented way. Knowledge structures are used to identify the core topics in Akogrimo and to classify knowledge sources. It enables:

- the documentation of best practice,
- the identification of critical issues,
- a structured representation of critical topics and
- skill levels that are required to develop Akogrimo services.

	Scenario Action	Method	Content
1	Business Process Model, generation Use Case Diagram, generation	AKBPM, UML™	Requirement Specification
2	Business Process Model, usage Use Case Diagram, usage	AKBPM, UML™	Business (Process) Model, Use Cases
3	Service Requirements, generation	AKBPM, UML™	Workflow Models, Service Model, Use Case Diagrams
4	Service Requirement, usage	AKBPM, UML™	Workflow Models, Service Model, Use Case Diagrams
5	Service Architecture, generation	UML™	Use Cases
	Roadmap, usage	PROMOTE®	Roadmaps
6	Service Architecture, usage	UML™	Service Design
	Roadmap, usage	PROMOTE®	Roadmaps
	Document upload into Repository		Documents
7	Service Architecture Review, upload	UML™	Service Design
	Roadmap, usage	PROMOTE®	Roadmaps
	Document download from Repository		Documents
8	Service Architecture Review, download	UML™	Service Design
	Roadmap, usage	PROMOTE®	Roadmaps
	Document download from Repository		Documents
9	Service Configuration, modelling	UML™	Service Design
	Roadmap, usage	PROMOTE®	Roadmaps
10	Service Configuration, usage	UML™	Service Design
	Platform Configuration, usage	UML™	Platform Design
	Application Specification usage	AKBPM	Workflow
11	Application Configuration, modelling	AKBPM	Workflow, Business (Process) Model
	Platform Configuration modelling	UML™	Platform Design
	Document upload into the Repository		Documents
12	Application Configuration, download	AKBPM	Workflow, Business (Process) Model
	Workflow, download	AKBPM	BPEL – download
	Platform Configuration download	UML™	XMI - download
	Document download from Repository		Documents

Table 4 Requirement List based on the Scenario Overview

Table 4 is based on Table 3 with a more detailed discussion on the used graphical methods and the model types that are considered as content.

3.2 Tool, Method and Content Detailed View

This section describes the previous mentioned steps in more detail, to identify the different components of the first version of the GET and the related modelling method. Each scenario action is divided into sub-actions. For each sub-action the necessary method and the applied content is listed. Table 5 describes each action and lists the according method and model type.

	Scenario Action	Method	Content Type[3]
1	**Business Process Model / Use Case generation**		
a	Modelling the Business Model and the relevant Business Processes using a stand-alone model editor. Modelling the Use Cases and Deployment Model using a stand alone model editor	AKBPM UML™	• Business Model • Business Process Model • Use Case Diagram • Deployment Model
b	Import the models from the stand-alone model editor into the Web-Portal of the GET.		
c	Import EPCs from an external modelling tool into the Web-Portal of the GET to describe technical processes in more details.	AKBPM	• EPC
d	Use the Web-Modeller to combine the models and to adapt the requirement definition.	AKBPM UML™	• Business Process Model • EPC • Workflow • Service Model • Use Cases • Deployment Model

[3] The content type is classified according the model types of the modelling methods.

2	**Business Process Model / Use Case usage**			
	a	View exiting models with the Web-Documentation component	AKBPM UML™	• Business Model • Business Process Model • EPC • Workflow Model • Service Pool • Use Case Diagram • Deployment Model
3	**Service Requirements, generation**			
	a	The workflows are configured using the Web-Modeller	AKBPM	• Workflow • Service Pool
	b	In case services are missing, the according service requirement description is defined.	UML™	• UML™ diagrams that found to be necessary
4	**Service Requirement, usage**			
	a	The Web-Documentation publishes UML™ and AKBPM models for a better analysis of the service requirement specification.	AKBPM, UML™	• Workflow Model • Service Pool • Use Case Diagram – AKBPM • Use Case Diagram – UML™
	b	The service development starts using the Service Development Roadmaps via the Web-Documentation.	PROMOTE®	• Roadmap models • Structure models

5		**Service Architecture, generation**		
	a	Design new services using the Web-Modeller	UML™	• Relevant UML™ Diagrams
	b	Upload the service description fact sheet into the repository		
6		**Service Architecture, usage**		
	a	Viewing the service design using the Web-Documentation and analyse the UML™ classes.	UML™ AKBPM	• Relevant UML™ Diagrams • Workflow
7		**Service Architecture Review, upload**		
	a	After the review the models are changed via the Web-Modeller.	UML™	• Relevant UML™ Diagrams
	b	Uploading the service description fact sheet into the repository		
8		**Service Architecture Review, download**		
	a	Viewing the reviewed service design using the Web-Documentation	UML™	• Relevant UML™ Diagrams
	b	Downloading of the reviewed service description fact sheet from the repository.		
	c	Export of the relevant UML™ models in XMI format.	UML™	• Relevant UML™ Diagrams
	d	Developing the service supported by the recommended Service Development Roadmap. The Service Development Roadmap can be viewed using the Web-Documentation.	PROMOTE®	• Roadmap

9		**Service Configuration, generation**		
	a	In case the service description has to be so detailed that reverse-engineering is necessary, the according XMI files can be generated and uploaded.	UML™	• Relevant UML™ Diagrams
	b	The reverse-engineered UML™ models can be manually modified, by using the web-modeler.	UML™	• Relevant UML™ Diagrams
10		**Service Configuration, usage**		
	a	The service configuration can be viewed using the Web-Documentation to analyse the service and its applicability in the workflow.	UML™	• Relevant UML™ Diagrams
11		**Application / Platform Configuration, generation**		
	a	The new service can be integrated into the workflow of the application specific service model to configure a complete application using the Web-Modeller.	AKBPM	• Workflow Model, • Business Process Model • Business Model
	b	The application configuration will be exported in BPEL format to test it in the Active BPEL engine. After successful tests the configuration is stored as final.	AKBPM	
12		**Application / Platform Configuration, download**		
	a	The Application Provider downloads the complete application relevant configuration from the Web-Portal into the local modelling environment to finally have the complete application configuration on site to prepare the product for selling.	AKBPM	

Table 5 Scenario Details

3.3 GET Web-Components

This section groups the above actions to Web-Components of the GET. Each Web-Component consists of a set of previously mentioned features that have an identifier to simplify the configuration of the platform. Table 6 lists the mapping of the Web-Components and the identifier for the configuration.

Module	Description	Application Development	Service Development	Platform Development
Workbench	The workbench provides role specific access to the Web-Components. Each Akogrimo user role has its own configured workbench.	WB	WB	WB
Web - Modeller	The Web-Modeler enables to change the models. It is a Java Applet that runs in the Internet Browser and accesses the models via Internet.	AKBPM-WM	PROMOTE-WM	UML-WM
Web - Documentation	The Web-Documentation enables the view and the navigation within the roadmap. It is a Web-Application that sends HTML to the Internet Browser to view the roadmap and to follow the specified hyperlinks.	AKBPM-WD	PROMOTE-WD	UML-WD
Web - Import / Export	This Web-Component combines all import and export features:			
	The **ADL Import / Export** enables the up- and download from the stand alone model editor ADONIS® to the GET.	AKBPM-ADL	PROMOTE-ADL	UML-ADL
	The **BPEL Export** enables the export of the Workflow in BPEL format into a local file.	AKBPM-BPEL	---	---
	The **EPC Import** enables the upload of process models in ECP format.	AKBPM - EPC	---	---
	The **XMI Import / Export** enables the export of the UML™ models in XMI format into a local file. Although XMI is a standard, there are differences between the various XMI exports, so a set of XMI – Exports will be provided.	---	---	UML-XMI
Web – Document Repository	This Web-Component is a simple repository of documents to enable the up- and download from documents into the roadmap.	AKBPM-WDR	PROMOTE-WDR	UML-WDR

Table 6 Component Roadmap mapping

3.4 Akogrimo Roadmap User Role Configuration

This section maps the previously mentioned Web-Components with the actions of the two development scenarios. Based on the this mapping the access right from the Akogrimo User Role and the Web-Component can be derived.

Table 7 introduces the Akogrimo Roadmap users with mapped access rights to the GET Web-Components. The columns represent the different Akogrimo Roadmap users whereas the rows represent the different Web-Components of the GET. The access rights are marked with an 'X' based on the action description of Table 5. Two workpackages already uses the GET, so this project specific configuration is also shown below.

The following short cuts are used for the **Akogrimo User Roles**:

Application Provider (AP), Platform Provider (PP), Application Configurator (AC), Platform Configurator (PC), Service Developer (SD), Service Development Consultant (SDC), Akogrimo Roadmap (AR)

The following short cuts are used for the **project specific configuration** that is used in workpackage 5.1 and 5.4:

Akogrimo UML™ Modeller (AUM)[4], Akogrimo UML™ Model Integrator (AUMI)[5], Akogrimo PROMOTE® Modeller (APM)[6] and Akogrimo PROMOTE® Model Integrator (APMI)[7].

Access Right	AP	PP	AC	PC	SD	SDC	AR	AUM	AUMI	APM	APMI
WB	X	X	X	X	X	X	X	X	X	X	X
AKBPM-WM	X	-	X	X	-	-	-	-	-	-	-
AKBPM-WD	X	X	X	X	X	X	X	X	X	X	X
AKBPM-ADL	X	-	X	X	-	-	-	-	-	-	-
AKBPM-BPEL	X	-	X	X	-	-	-	-	-	-	-
AKBPM-EPC	X	-	X	-	-	-	-	-	-	-	-
AKBPM-WDR	X	-	X	X	X	X	-	-	-	-	-
PROMOTE-WD	X	X	X	-	X	X	X	X	X	X	X
PROMOTE-WM	-	-	-	-	-	-	-	-	-	X	X
PROMOTE-ADL	-	-	-	-	-	-	-	-	-	-	X
PROMOTE-WDR	-	X	X	-	X	X	-	-	-	X	X
UML-WM	-	X	-	X	X	X	-	X	X	-	-
UML-WD	X	X	X	X	X	X	X	X	X	X	X
UML-ADL	-	X	X	X	X	X	-	-	X	-	-
UML-XMI	-	X	-	X	X	X	-	-	-	-	-
UML-WDR	-	X	-	X	X	X	-	X	X	-	-

Table 7 Access Rights by Roles Overview

Table 7 indicates the access rights for each proposed Akogrimo User Role based on the identified action of Table 5.

In general the Web-Documentation is provided to each Akogrimo User to enable a complete documentation. The access to the Web-Modeller and Web-Import / Export Component is based on the scenario description where usually on role is only concerned with one method.

[4] Currently all WP 5.1 partners have been assigned to this group.
[5] Currently UPM has been assigned to this role.
[6] Currently all WP 5.4 partners have been assigned to this group.
[7] Currently BOC has been assigned to this role.

4 The Overall Architecture

The GET that has been developed in Akogrimo as a combination from commercial software from BOC, from prototypes resulting from other EU-projects, which have been integrated and improved and new components, which have been implemented for the Akogrimo scenario.

The result is a unique distributed Web-Modelling environment for Service Oriented Software development with some special adaptations for the Akogrimo platform.

It includes the integration of an individual model editor installation with a Web-modelling platform where a Web-Modeller accesses the same repository than individual stand alone model editors.

Both the Web-Modeller and the stand alone model editor implement the meta-modelling paradigm to provide three different graphical modelling methods in the same modelling environment.

Additional concepts and interfaces have been implemented to provide a complete set of features , which support development scenarios that encourage the developer to think about the wider context of its service and support the process-oriented service specification.

The resulting GET is currently the most complex Web-Modelling infrastructure that provides graphical models to develop Service Oriented Software.

The next sections describe in more detail the overview of the GET and list a description for each component.

First an overview of the various components explaining the interaction between the different GET parts is provided.

Second each layer is described in more detail stressing the different components.

Table 8 describes the component fact sheet that is used for each relevant component of the GET to point out the functionality and the work that has been performed within Akogrimo.

Name of the Component	
Functional Description:	The functional description is a brief introduction to component to describe the different responsibilities. The reader should get an insight into the major parts of the GET.
Technical Glimpse:	The technical glimpse provides a brief introduction of the component to provide an overview of the technology that is used and point out some key figures indicating the complexity of the component.
Implementation Effort:	The implementation effort provides an overview of the work that was necessary during the project to integrate this component into the GET to fulfil the requirements of Akogrimo.

Table 8 Description of the Component Fact Sheet

The following section introduces the high level component overview that depicts the overall picture of the GET software components.

4.1 High Level Component Overview

The GET for the roadmaps in Akogrimo is a special configuration of a Web-Modelling platform using a set of components that have been either adapted to the needs of Akogrimo or developed. This chapter provides an overview of the setup of the components and provides some key figures of the software.

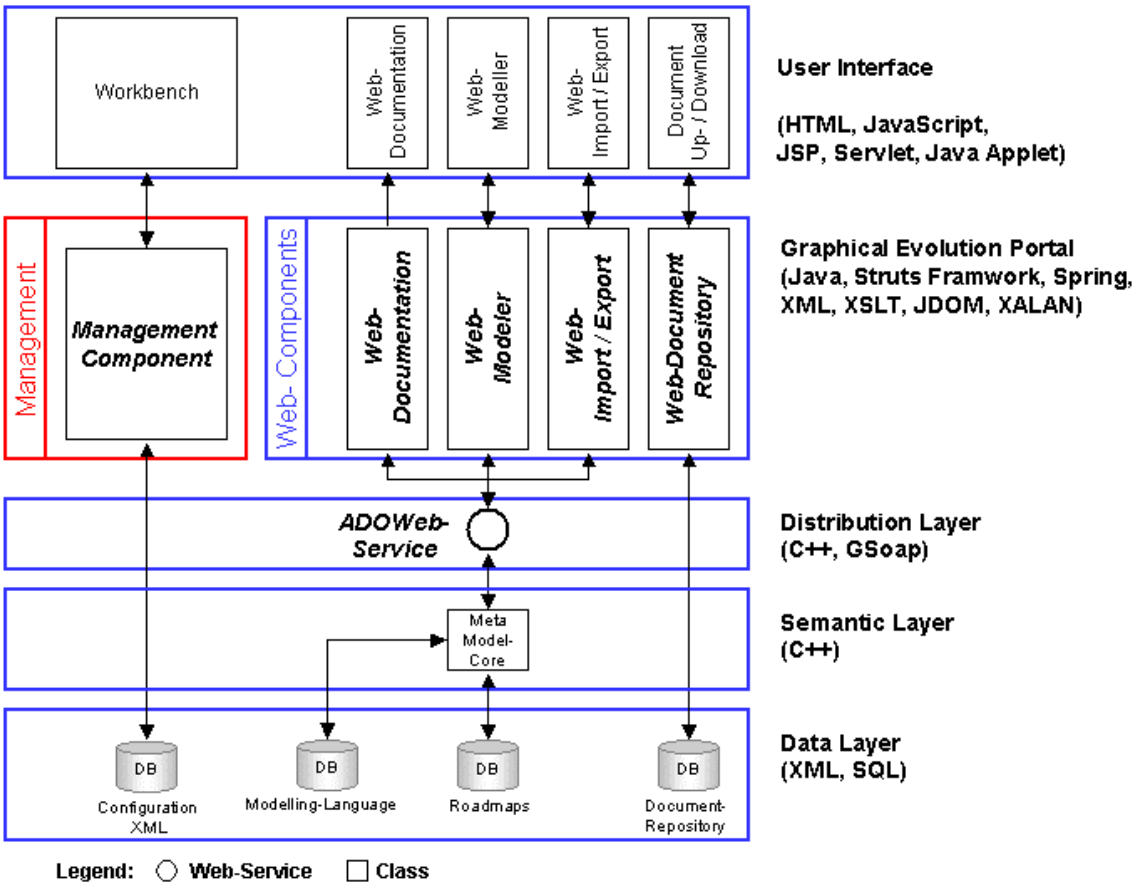


Figure 6 Overall Architecture of the Graphical Evolution Tool Prototype

Figure 6 depicts the major components and layers of the GET. All the components are briefly introduced to identify their role within the platform.

4.2 Data Layer:

The data layer consists of four repositories, which are concerned with data about the configuration of the platform, the modelling method, the roadmaps and the documents for the roadmaps.

Configuration Repository

Functional Description:	The configuration repository consists of XML files with individual XML syntax, which setup the Web-Portal. This includes the user access rights, the user management, the user group management and the style of the workbench.
Technical Glimpse:	The settings in these XML files therefore configure the user-specific workbench and define the access rights. This component is a 1st Version, the planed improvements will include decryption mechanisms for the xml file, the use databases instead of xml files and the integration of LDAP.
Implementation Effort:	The configuration is Akogrimo specific and has been built on the basis of the previous chapters.

Modelling method

Functional Description:	The modelling methods configure the meta-modelling platform for a specific usage scenario like modelling with UML™, modelling with PROMOTE® or requirement modelling with AKBPM. This configuration can be changed with only minor adaptations to the rest of the system to provide rapid improvement of the modelling method and fast responds to user requirements.
Technical Glimpse:	The modelling method is a configuration in text file format of the meta modelling platform using a special configuration language. The text file is encrypted and stored in a database. In the current setup a free licence database is used, which is MSDE.
Implementation Effort:	The AKBPM – method was adapted from the eBusiness method and intensive changes were necessary to integrate EPC. The Workflow and Service Pool were implemented and a set of user specific adaptation based on D 3.2.2 (The Business Modelling Framework) were realised. This method was mainly implemented new and nearly all parts of the method had Akogrimo relevant adaptations. Investigations were necessary to make this method Web-modelling enabled. The UML™ – method was upgraded from UML 1.3 to UML 2.0 and translated into English. User specific adaptations in the Deployment and Class Diagram had been made. Investigations were necessary to make this method Web-modelling enabled. PROMOTE® was translated into English, only minor user requirements were necessary. Investigations were necessary to make it Web-modelling enabled.

The following table identifies some key-figures of the model languages.

Modelling method	Size	Lines of code of method
AKBPM	2.8 MB	43.555
UML™	1.9 MB	39.048
PROMOTE®	8.4 MB	41.948

Table 9 Key Modelling method Figures

Roadmaps

Functional Description:	In this use case the models are used as roadmaps. All models that have been generated are included in the repository, which provides the usual manipulation features for generating, deleting changing the models.
Technical Glimpse:	The models are stored in a database using the meta-modelling mechanism. In the current setup an MSDE free licence database is used to store the models that can be accessed by SQL statements.
Implementation Effort:	There were no changes necessary in the database itself. The models have been manually generated during the workpackage using the model editors.

Document Repository

Functional Description:	This repository collects documents that are referenced from the models. In this scenario it is used as an extension to other document repositories that are directly accessed like the Akogrimo Homepage and the BSCW server for Akogrimo.
Technical Glimpse:	A document repository has been implemented using an Oracle database for storing documents in the database.
Implementation Effort:	This repository was implemented for the Akogrimo GET.

This layer is seen as the collection of the data. The interpretation following the meta-modelling paradigm is performed in the next layer the semantic layer.

4.3 Semantic Layer:

The semantic layer consists of the meta-modelling platform that is the major component of the system that interprets the different modelling methods and establishes a communication with the model repository to higher layers.

Meta-Modelling Platform / Meta-Model Core

Functional Description:	The meta-modelling platform accesses the repository and interprets the semantic of the models using the modelling methods. This platform manages all models, model types, modelling objects and attributes and keeps track of the versions. There are model functionalities to import, export, manage, maintain and analyse the models. The import and export functionality enables the exchange with other tools in various formats. The management functionality enables versioning, access rights configuration and model group management. The maintenance is a key functionality to generate new models and model objects, to delete or change all entries. There are various modes and functionality available to make the modelling user-friendly. The analysis enables queries of the models. The above features are either directly used by BOC client software, or a list of these functionalities is provided via a script language with about 400 functions as a public interface to the Web-Service of the higher layer.
Technical Glimpse:	The interpretation of the roadmaps is performed from a commercial meta-modelling platform written in C++ byBOC that has not been adapted in Akogrimo. The installation size is approximately 54.8 MB.
Implementation Effort:	No adaptations for the Akogrimo project.

4.4 Distribution Layer:

The distribution layer is concerned with providing distributed access to the lower layers by Web-Services.

ADOWeb-Service

Functional Description:	The ADOWeb-Service provides access to the meta-modelling platform using a gSoap Server. It enables external components to use the public interface of the meta-modelling platform by translating Web-Service requests into script language of the Meta-Modelling Platform. In the Akogrimo scenario this Web-Service is used to view, change, import, export, transform and to integrate documents into the roadmap.
Technical Glimpse:	The ADOWeb-Service is implemented in C++ using gSoap to provide a Wrapper that translates SOAP and communicates with the meta-modelling platform.
Implementation Effort:	A prototype of the ADOWeb-Service was upgraded and implemented as first release for the GET, as it is the first installation of such a complex Web-Modelling infrastructure.

This distribution layer was intensively used during the Akogrimo project and a list of initial bugs has been found and fixed to guarantee a stable and reasonable performance.

4.5 Graphic Evolution Portal Layer:

The Graphical Evolution Portal consists of the **Management Component** that provides all portal basic features for the Graphical Evolution Portal and the **Web-Components** that have been identified in the previous chapters and provide the functionality to the user.

Management Component

Functional Description:	The portal management component provides basic user and access management enabling the configuration of access rights to components and provides user specific Workbenches.
Technical Glimpse:	This component consists of about 20 Java Classes that directly access the configuration files for the portal in XML format.
Implementation Effort:	Existing management components have been used and upgraded to better support the Akogrimo scenario.

Beside the management component there are four Web-Components, which are described in the development scenario.

Web-Modeller

Functional Description:	The Web-Modeller provides full functionality to model roadmaps via the Internet using an Internet Browser. The user can model roadmaps by adding, changing or deleting models where the changes can either be stored in the repository or stored locally on the file system of the client. The Web-Modeller understands different modelling methods and provides modelling features for the users like navigation trees, modelling tool bars, zoom features, grid and snap features, basic maintenance and model navigations.
Technical Glimpse:	This component is the most complex Web-Component as it deals with intensive user interaction, with the interpretation of the modelling method and with the management of models. There were about 900 Java Classes, as one result of the EU-Project PROMOTE that has been continuously improved and commercially released. The Web-Modeller uses Java Applet technology.
Implementation Effort:	An existing prototype was used and adapted to execute the three graphical modelling methods in Akogrimo. The major effort was necessary to adapt the rudimentary parts for English modelling methods. It is the first time that the Web-Modeller is used for English modelling methods. Effort was necessary to adapt internal concepts (like model expressions, inter model object references, handling of hidden model properties etc.) to fulfil the needs of the complex AKBPM, UML™ and PROMOTE®. It was the first time that such complex modelling methods have been used in the Web-Modeller. Effort was necessary to identify and deal with integration issues such as dealing with more than one Web-Modeller per user (session handling of the Browser), with the integration of stand alone modeller and the Web-Modeller (conflicts of model ids) and the integration of within the GET platform. It is the first time that a Web-Modeller is integrated in such a Web-platform.

Web-Documentation

Functional Description:	This component is the main component to view the roadmaps in Akogrimo as it provides a simple user interface to navigaten within models. The Web-Documentation understands all three modelling methods and visualises the roadmaps in an Internet Browser via HTML that is dynamically generated from the roadmap repository. Basic user features are implemented for a user-friendly navigation within the roadmaps and navigation to integrated documents.
Technical Glimpse:	This component consists of about 35 Java classes that dynamically generate html pages. An XML stream representing the models is provided by the ADOWeb-Service and transformed by the Web-Component using XSLT. A load balancing, caching and fault resistance has been developed to provide fast and stable roadmaps.
Implementation Effort:	The Web-Documentation started from a prototype and has been massively improved during the project. It is the first time that BOC provides a dynamic Web-Documentation in conjunction with a Web-Modeller. The major effort was necessary to adapt the translation files (XSLT) to

	produce a reasonable Web-Documentation for the three modelling methods. Effort was necessary to adapt the initial style to a user friendly interface providing a set of new features, which were requested during the modelling workshops (refresh, link window, model history navigation, attribute filter). All performance relevant parts have been re-implemented as Akogrimo is the first demonstration of the Web-Documentation. This includes the Web-Service Caching, the Web-Service Load Balancing and the Fault-Resistance. This Web-Component was intensively changed during the modelling workshops of the workpackage.

Web-Import/Export

Functional Description:	This component enables the up- and download of models in various formats. Partners of the Akogrimo project use also stand alone editors to maintain their models. The models can be exported from the stand-alone version and uploaded into the portal. The UML™ diagrams can be up- and downloaded in various XMI formats using transformation scripts. The import of EPC is possible using the upload feature in combination with model transformation. The download of BPEL is also possible by transforming the workflow into a standardised format.
Technical Glimpse:	This Web-Component consists of about 20 Java Classes, which deal with the import, export and transformation of models. In case own model formats are up- or downloaded the component transmits the model file to the ADOWeb-Services. In case external model formats like EPC, XMI or BPEL are required, the import and export component uses XSLT and a model transformer program to transform the models.
Implementation Effort:	The EPC interface was first implemented for the stand alone version by writing translation scripts for first a format transformation from the EPC format to the ADONIS® format. Second a semantic transformation from the EPC semantic to the AKBPM semantic and finally a language transformation to transform the used terminology of the EPC models to the terminology of the AKBPM models. The XMI interface was also first implemented for the stand alone version by writing translation scripts for the Rational XMI dialect, the Eclipse Visual Paradime XMI dialect and the Poseidon XMI dialect. The interface was then transformed into the Web-Component. The relevant parts of the BPEL interface were implemented for Akogrimo defining the relevant concepts in the AKBPM method and developing the translation files for a format transformation into the Active BPEL engine format. Finally effort was necessary to build a Web-Component that covers all individual interfaces and can be integrated into the GET. This flexibility in integrating different modelling formats and methods is unique and is a result of the Akogrimo Roadmap requirements.

Web-Document Repository

Functional Description:	The Web-Document Repository enables the up- and download of documents in a Web-Repository. This is provided to enable the integration of documents in the roadmaps by using an integrated Web-Repository. It is also possible to integrate a document that is already maintained by another repository like the BSCW or the Akogrimo Homepage. In case a single document should be integrated, it can be uploaded in the Web-Document Repository. Simple usability features for a better integration into the roadmaps are provided.
Technical Glimpse:	This component consists of about 15 Java Classes that enable the up- and download of the documents in the database. The files can be accessed using a path within the repository or by the file id.
Implementation Effort:	This repository was implemented for the Akogrimo GET.

4.6 User Interface Layer:

The User interface layer is concerned with the generation of Web-User Interfaces for the components described above.

The Web-Modeller requires a high sophisticated User Interface to provide feasible mechanisms to model. This component is implemented as a Java Applet 1.42 or higher on an MS-Internet Explorer. There were only minor adaptations because of Akogrimo specific requirements.

The other components are using the JSPs and the Struts-Framework that run on all common browsers. For some minor enhancements some simple JavaScripts have been used that are supported by the major browsers.

The implementation effort is identical to the previously described components as currently there exists only one user interface per Web-Component.

5 The IT-Infrastructure

This section describes the IT-Infrastructure of the first version of the GET that has been installed at the BOC Server.

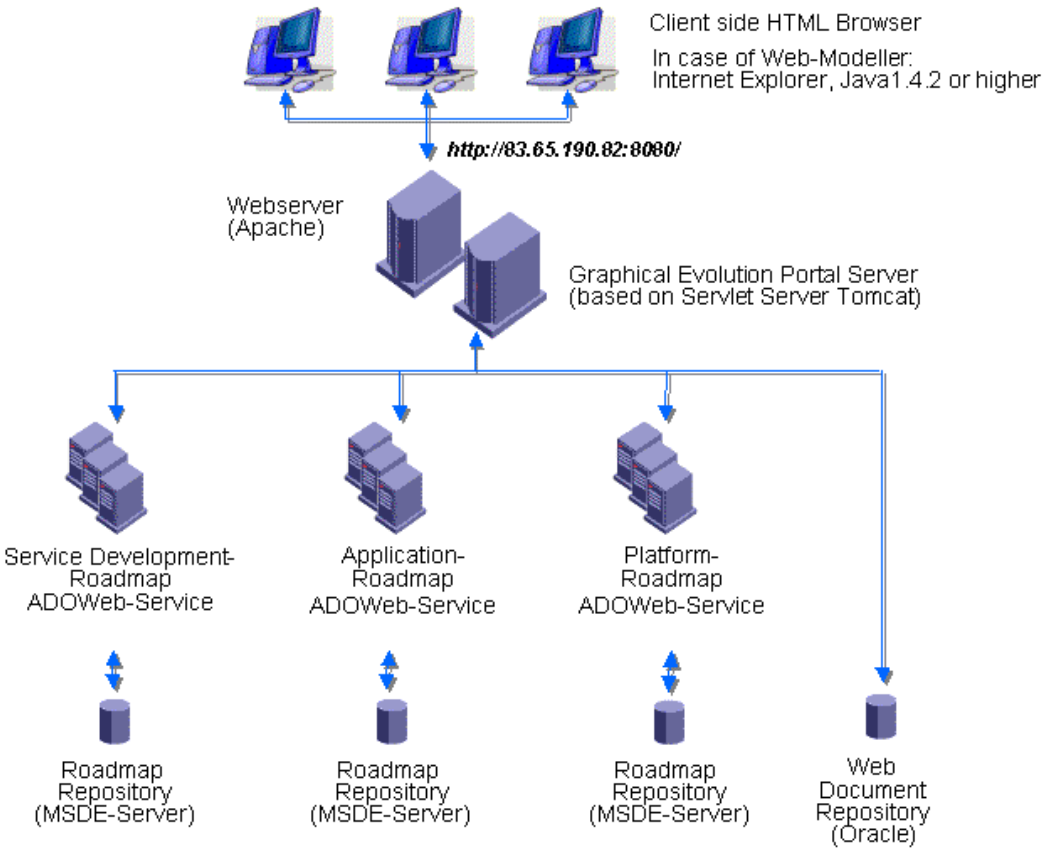


Figure 7 IT-Infrastructure of the Graphical Evolution Tool

Figure 7 indicates the IT-Infrastructure of the GET that runs on a Windows XP Server for the Akogrimo scenario.

The user client requires an Internet Browser to access the GET and to view the roadmaps. The user interface is provided in HTML format with some additional Java Scripts, except the Web-Modeller. The roadmap has been tested with the key Internet Browsers on Windows operating systems.

The Web-Modeller requires MS-Internet Explorer with the Java Plug-In 1.42 or higher. The Web-Modeller deals with intensive user interaction and requires therefore a more sophisticated client side user interface than the other Web-Components.

The client connects to the Web-Server located in Vienna behind a Hardware Firewall using http://83.65.190.82:8080/. The port is observed by the Apache Web Server that forwards the requests for the GET to the Servlet Server using a default Web-Server configuration.

The first version of the GET uses Tomcat 5.1 to interpret the Java Server Pages and run the Web-Application. The Tomcat Server contains the Web-Application of the GET using common Tomcat configurations.

The Web-Application contains the previously described Web-Components like the Web-Modeller, the Web-Documentation, the Web-Import/Export and the Web-Document Repository.

Each of the Web-Components communicate with the ADOWeb-Services, which are Web-Services using gSoap. These ADOWeb-Services, which are listening to identified ports, currently run on one physical machine. Each of the ADOWeb-Services establishes a communication with a meta-modelling platform and ensures the availability of the meta-modelling platform for each request. This means that several instances of meta-modelling platforms (one for each Web-Service) are started that connect to the same database.

The ADOWeb-Service continuously listens to the specified port and in case of a request communicates with the meta-modelling platform. Currently there are three ADOWeb-Services running in parallel per modelling method to enable load balancing and fault resistance, which results in nine Web-Services that run in parallel.

The meta-modelling platform runs in the same environment like the ADOWeb-Service to guarantee a fast and stable communication. The meta-modelling platform communicates with the database to manipulate the models. Most of the requests have low workload at the meta-modelling platform but a high workload at the database.

The GET has a database for each modelling method, as each roadmap uses its own modelling methods. So there is a database for AKBPM, a database for UML™ and a database for PROMOTE®.

In case the workload from the GET is too heavy for the server, it is planned to distribute the three databases for each modelling method physically on a different machine. This results in two servers, where one is managing the Web-Servers and the Web-Application of the GET and the second server is dealing with the three databases.

6 User Guide

This section introduces the Web-Components of the GET by presenting screenshots and by explaining key features of the user interface.

The target is to provide an impression on the usability and functionality of the prototype and to act as brief user guide for users of the tool.

6.1 The Workbench

This section provides an overview of the functionality and usability of the Workbench. Starting the Akogrimo Workbench the user will be asked to login. Figure 8 presents the login page.

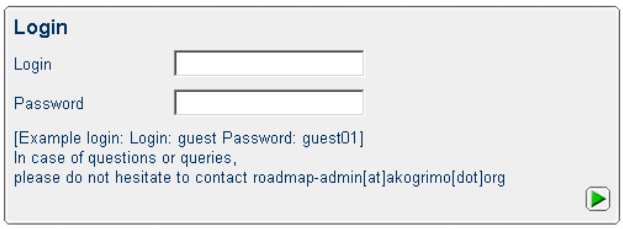


Figure 8 Login Page

After logging into the tool, the user is directed to a specially composed workbench that provides tabs for main navigation sites, and a list of Web-Components on each site.

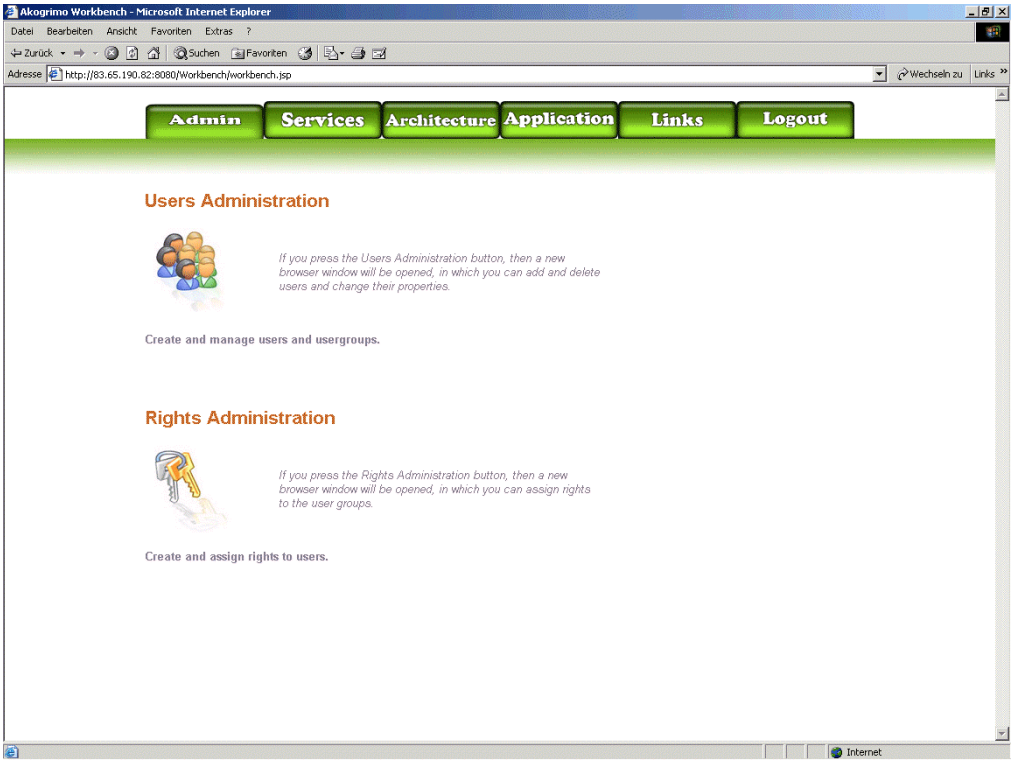


Figure 9 General Workbench Overview

Figure 9 points out the general workbench overview presenting the tabs of the administrator of the GET. The Workbench consists of six tabs. Each tab allows access to different components. All these components will be described in more detail in the following sections.

The following list is an overview of the six tabs and the components, which can be accessed using the different tabs:

- **Admin tab:** The Admin tab consists of the Management Component to administer users and access rights.
- **Service tab:** The Service tab deals with the service development that is modelled with PROMOTE®. It includes access to the Web-Modeller, to the Web-Documentation and to the Akogrimo Web-Document Repository.
- **Architecture tab:** The Architecture tab deals with the platform development that is conceptually supported by UML™. Similarly to the service tab, this tab enables access to the corresponding Web-Modeller, Web-Documentation, Akogrimo Web-Document Repository and to the Web-Import/Export Interface.
- **Application tab:** The Application tab deals with the application development that is conceptually supported by the AKBPM Method. Like the other tabs it allows access to the corresponding Web-Modeller, Web-Documentation, Akogrimo Web-Document Repository and in addition to a Web-Import/Export Interface.
- **Link tab:** The link tab lists some useful web pages.
- **Logout tab:** The logout tab closes the user session.

The following screenshot depicts a list of all Web-Components.

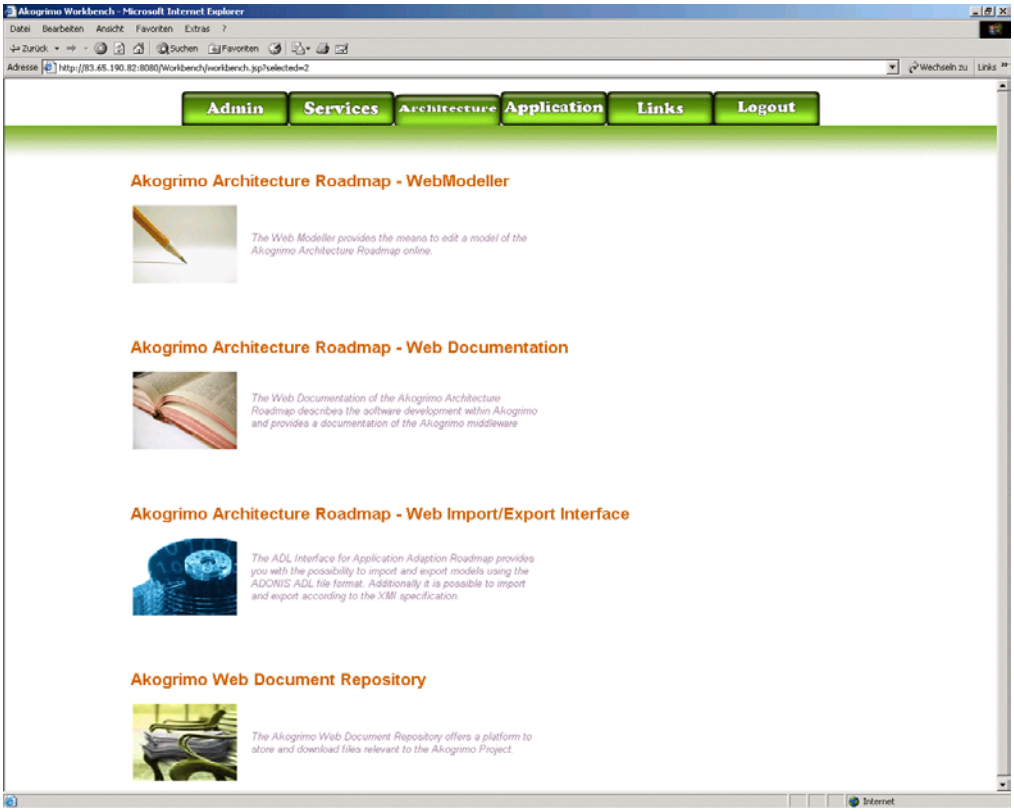


Figure 10 Web-Components in the Workbench

Figure 10 indicates how the Web-Modeller, the Web-Documentation, the Akogrimo Web-Document Repository and the Web-Import/Export are provided in the workbench.

The user can select a Web-Component that is afterwards opened in a new Browser Window. The Web-Modeller requires an additional login of the user, as this Web-Component establishes a direct connection to the roadmap repository. So the user of the Web-Modeller can delete, change and update all models to which he has access rights. This means that the users might have the same access rights on the workbench (e.g. two Service Developer) but have different access rights to the Web-Modeller (e.g. two Service Developers from different partners, where each partner can only change their own models).

The Web-Documentation and the Web-Import/Export have no additional access rights, as these components can only view the roadmaps (depending on the access rights of the models) but cannot change or delete the content.

The Akogrimo Web-Document Repository also has no additional access rights, as every partner should be able to up- and download documents.

Each Web-Component has its own icon and a short description, whereas the icon is the same for each modelling method (roadmap) but the description varies depending on the different uses of the Web-Component.

6.2 The Toolbar

This section describes the toolbar items, which support the usage of the Web-Components. Therefore Table 10 describes every tool and shows the corresponding icon, as it is represented within the GET components.

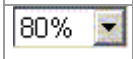
Tools	Feature Name	Description
	**Navigation Tree Narrower**	The vertical minimizer reduces the tree navigation window and enlarges the model view window. Each time the button is pressed, the windows are resized by a fixed step-width.
	**Navigation Tree Wider**	The vertical wider enlarges the tree navigation window and reduces the model view window. Each time the button is pressed, the windows are resized by a fixed step-width.
	**Model backwards**	This button indicates the model history and appears when more than one model has been viewed. The backwards button loads the previous viewed model.
	**Model forward**	This button indicates the model history and appears when more than one model has been viewed and the backwards button was used. The forward button loads the viewed model from which the backward button has been used.
	**Details Mode on/off**	The details button switches the details mode on or off. If the details mode is on, the details window is placed below the navigation tree to show a detailed description of each selected object. If the details button is switched off, there is a pop-up window instead of the details button. The pop-up window only views navigation links for a fast navigation within the models and a comfortable reference to documents. The details view is customised in the modelling method. This means that attributes that are not necessary to be seen can be hidden. The usage of the details mode will be described in more detail in the following section.
	**Refresh**	Usually the Web-Documentation runs with cached images that are stored for 1 min. In case this 1 min cach is too long, the user can refresh using the refresh button. If this button is clicked, the model is again loaded from the database.
	**Help**	A short help page will be provided when clicking on the help button. In general the frames are explained.
	**Print**	This button prints the model view window. In order to have a nice printed page, it is advisable to use the print view button beforehand to change the zoom of the model till the model fits on the printer page.
	**Print View**	The print view enables a preview of the printed page. It is recommended to first use the print view mode before printing.
	**Model Zoom**	The model zoom reduces the size of the model picture. It can be used to shrink large models to fit on the screen or to fit on the printing page.

Table 10 General Tool Icons

6.3 The Web-Documentation

The Web-Documentation component visualizes the models and therefore enables the user to access the roadmaps. The tool bar at the top provides some usability features like zooming, model navigation or details options that are described in the tool list of section 6.2. When looking at the Web-Documentation the user will find a model navigation tree at the left part of the window. Here all the previously designed models according to model groups can be viewed. Model groups are similar to directories within the model repository. Choosing a model by clicking on it will open the corresponding model at the right part of the window.

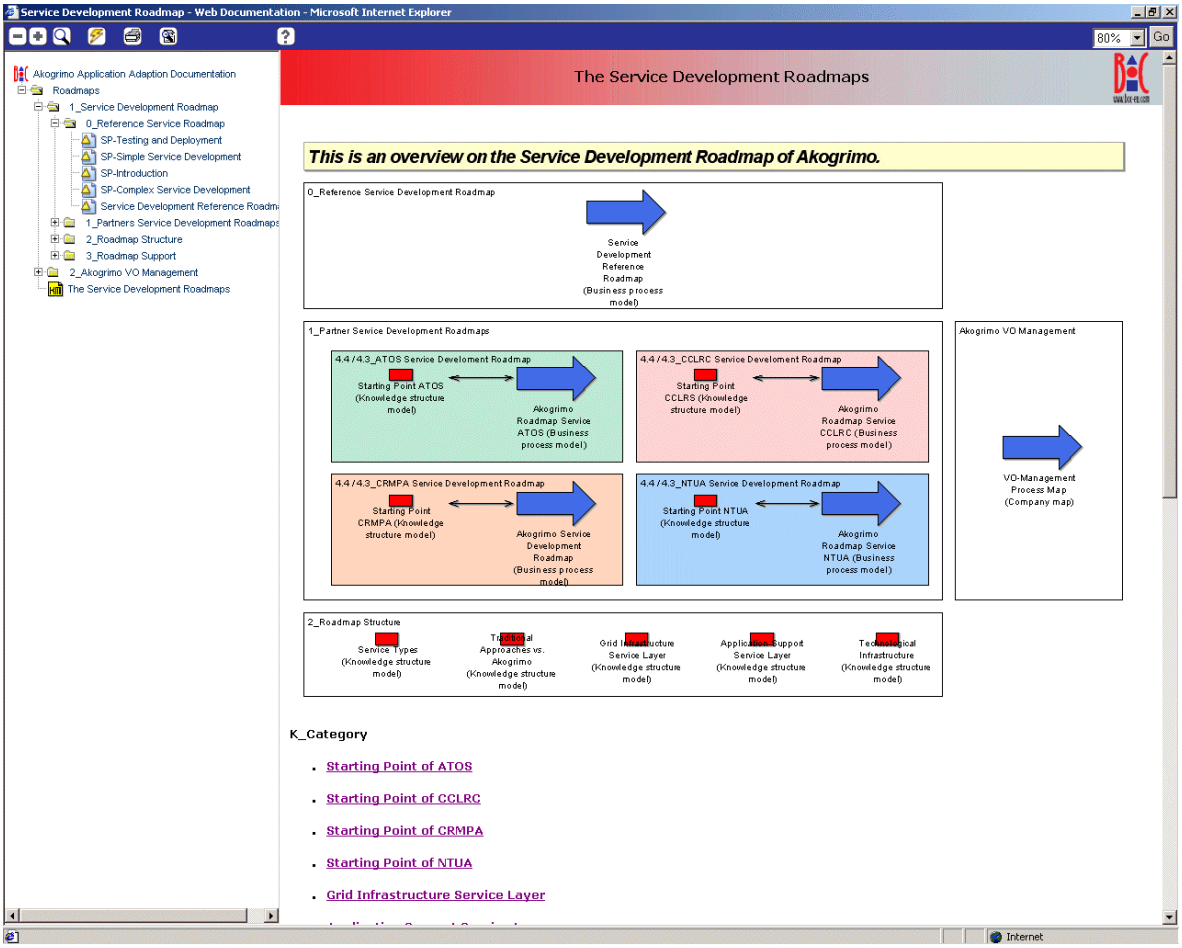


Figure 11 Screenshot of the Web-Documentation

Figure 11 shows a screenshot of the overview on the Service Development Roadmap. The right part displays the selected model "The Service Development Reference Roadmap" by presenting a picture and by providing information about the objects.

A detailed view of each object is possible either by switching to the details window, which appears at the left bottom corner, or by using the quick link window.

6.3.1 The Quick Link Window

Using the quick link window is a very efficient way of getting information about an object and to navigate through the different models. By clicking on the symbol, of which the user wants to get information, a window (called the quick link window) will open. The quick link window shows

references to other models, which will describe the object in more detail. Also links to resources, like documents, which should support the usage of the roadmap, are listed.

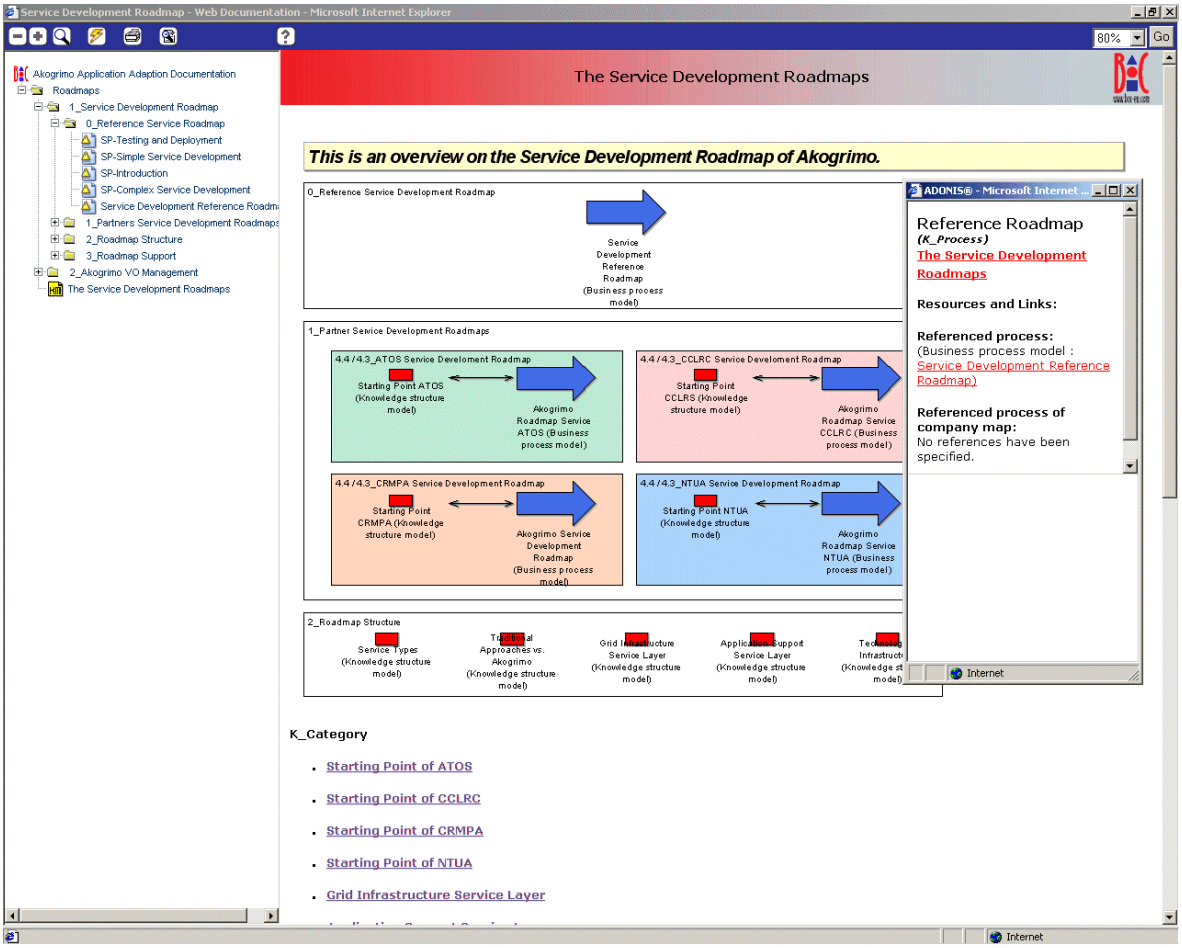


Figure 12 Quick Link Window

Figure 12 shows the quick link window, which opens by clicking on the process symbol called "The Service Development Reference Roadmap". The links include a reference to a business process model, which will describe the Service Development Reference Roadmap in detail. Clicking on the red underlined link named "Service Development Reference Roadmap" opens the corresponding process as Figure 13 shows.

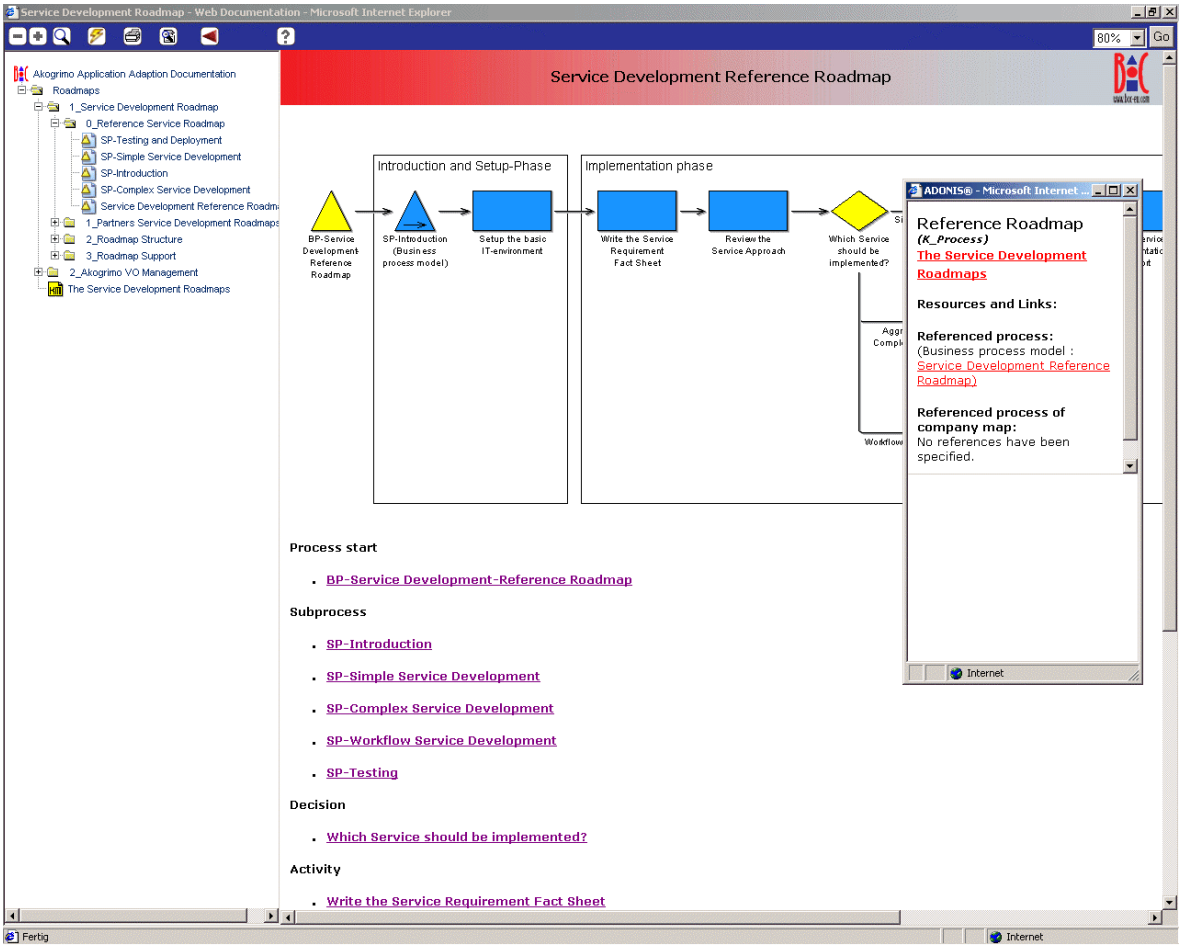


Figure 13 Following a Link

Following this example the user can easily navigate through the roadmap and get information about certain aspects as depicted in Figure 13.

6.3.2 The Details Mode

The second possibility to get information and to navigate through the roadmap is the usage of the details mode. A click on the details button opens the details window, which is placed on the left hand side of the window below the model navigation tree.

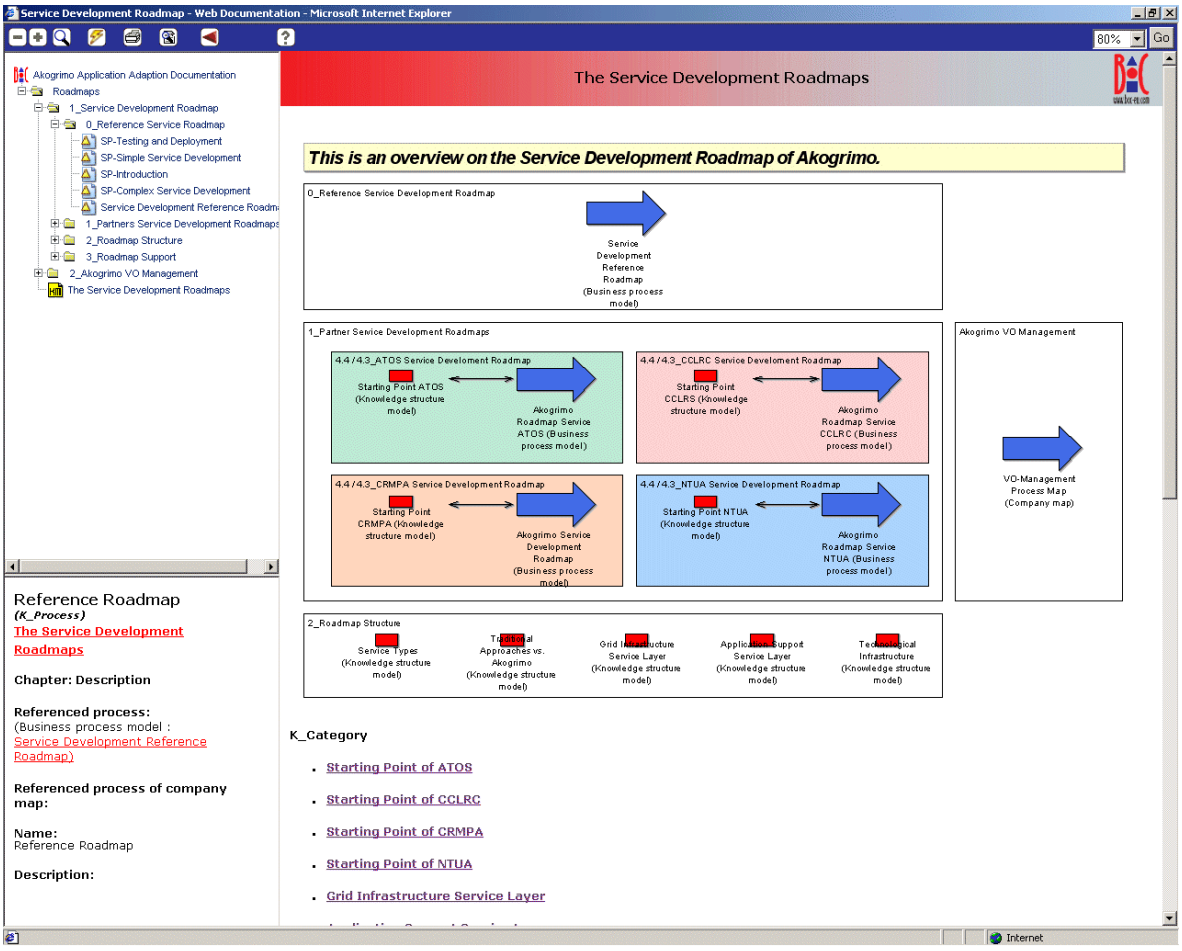


Figure 14 Details Window

Figure 14 shows the details window on the left hand side below the navigation tree. The details window contains information about the chosen object, like e.g. characteristics or references to other models. In case of this example a click on the process symbol called "The Service Development Reference Roadmap" shows details about this process, e.g. a link to the referenced business process model. Following references works similar to the quick link window. Clicking on the link will open the referenced model. In contrast to the quick link window, the details window contains not only links and resources, but also additional information, like e.g. a description of the object.

6.4 The Web-Modeller

The Web-Modeller is the most complex and highly sophisticated Web-Component that enables the maintenance of the roadmaps via an Internet Browser.

In the following some key features are pointed out by some screenshots.

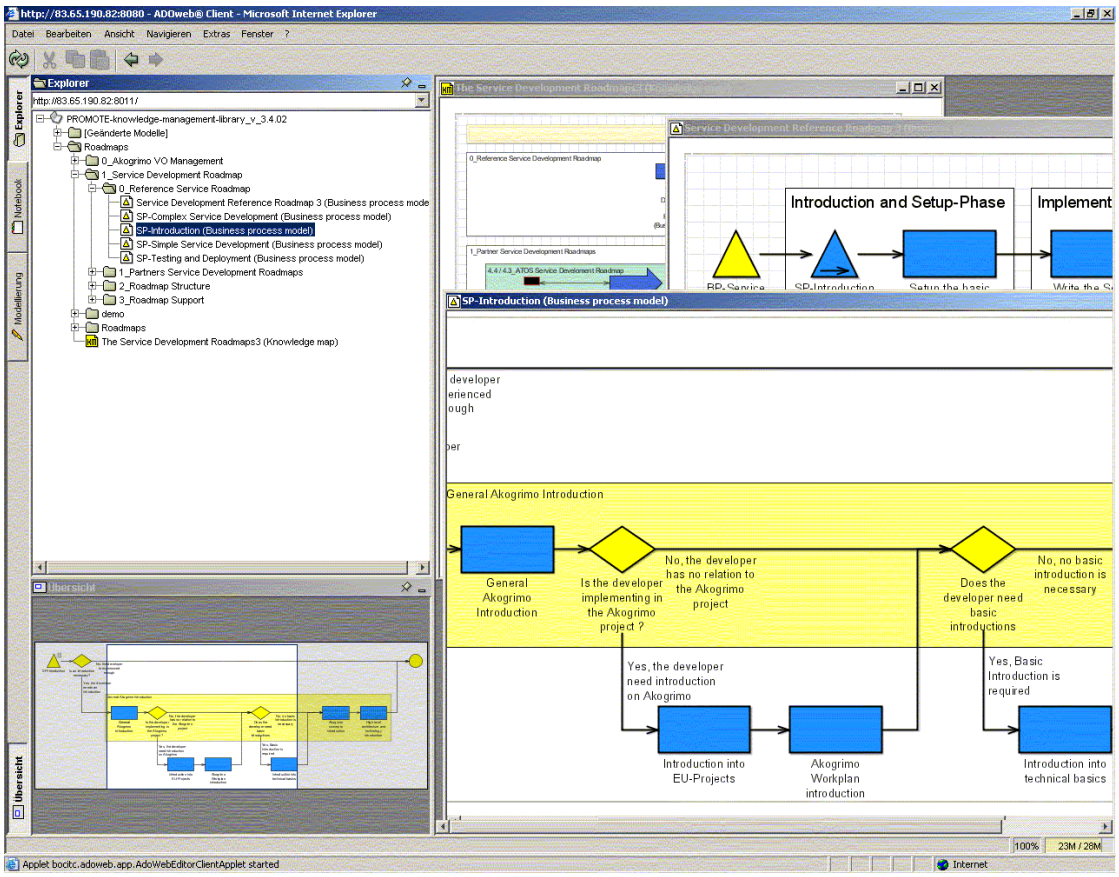


Figure 15 Screenshot of Web-Modeller

Figure 15 shows some of the feature of the Web-Modeller in a screenshot.

Basically the Web-Modeller has a modelling space that is by default maximized. The screenshot above shows the following:

The model-window keeps three different models, which have been scaled down, so three different models, are represented in standard window style. The selected process is in front.

The zoom window in the bottom left corner shows the currently selected process that is front of the model window and indicates the area that is currently visible. The zoom can be changed to see the whole model, to scale to the original size or to scale a specific selected area. An area can be selected with the right mouse.

The model explorer at the top left corner shows all model groups and models from the roadmap repository. Using the right-mouse click provides basic maintenance features to generate, delete or rename models.

The menu at the top provides additional functions like the visualisation of a grid, support in editing the models, changing the language for the user interface or navigate within the loaded models using forward and backward features.

New models can be generated by a right click on the folder (model group) to which the model should belong. Then one after the other the options "New" and "Model…" have to be selected. Now a window opens which contains all available model types. After selecting the respective model type, a name for the new model can be entered. This procedure is shown in Figure 16.

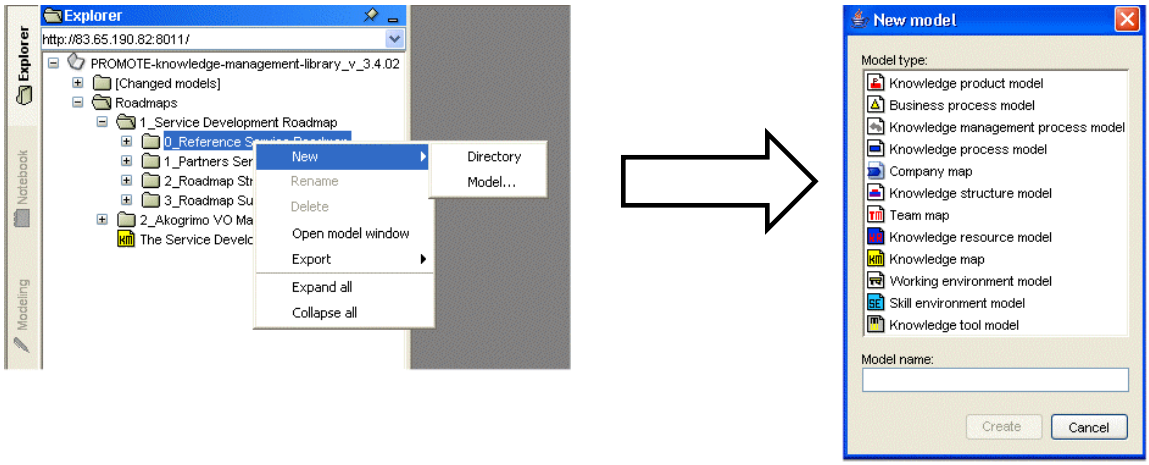


Figure 16 Generation of New Models

The next screenshot (Figure 17) shows the modelling view of the Web-Modeller. The user can easily switch to the modelling mode by using the tabs on the left side of the window. The modelling view enables the user to create or edit models.

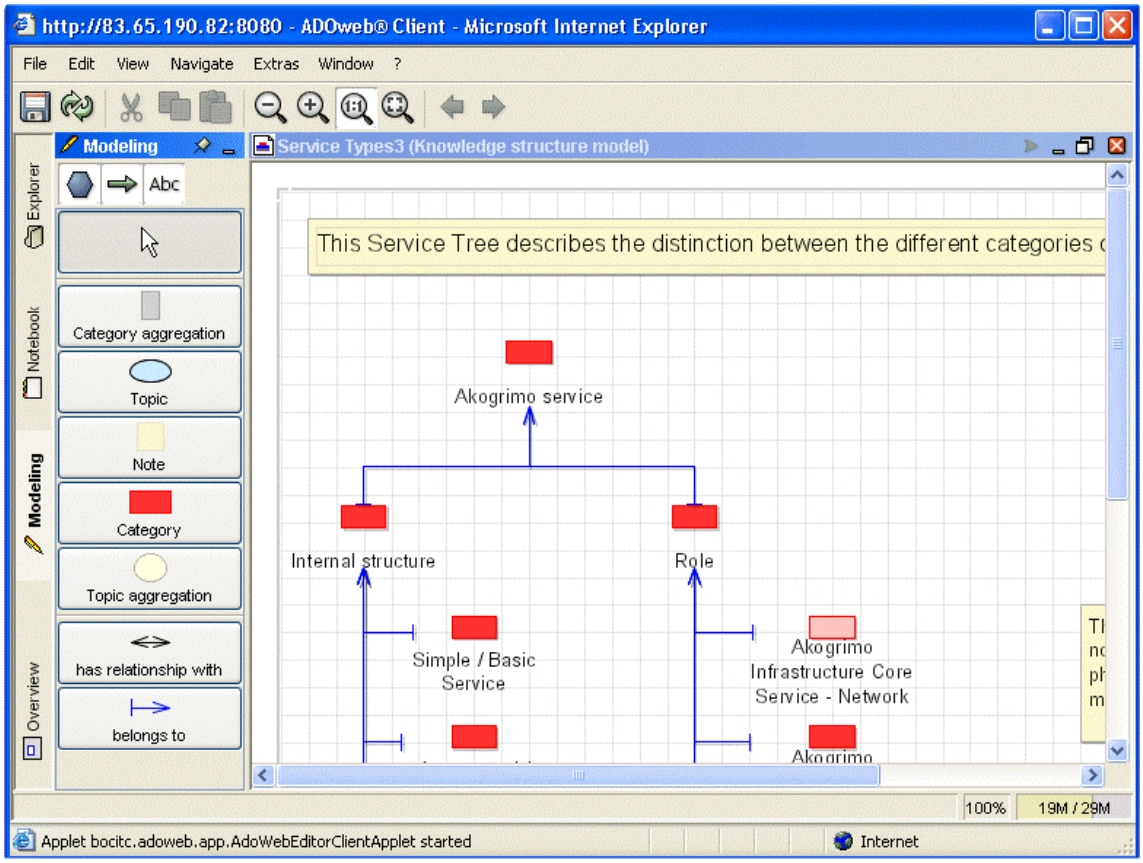
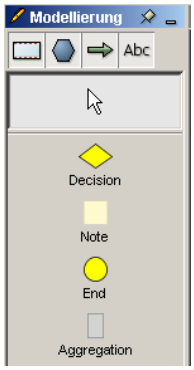


Figure 17 Modelling View

Figure 18 shows how new objects can be added using the modelling list.



The modelling list enables the user to add new modelling objects using drag and drop.

The symbols at the top indicate whether the modelling objects, the relations or the description are displayed.

Figure 18 Modelling list

The next screenshot introduces the notebook, which includes detailed information on each modelling object. By selecting a modelling object – in the example below it is the "Akogrimo Infrastructure Core Service – Network" – the according notebook is presented on the left hand side.

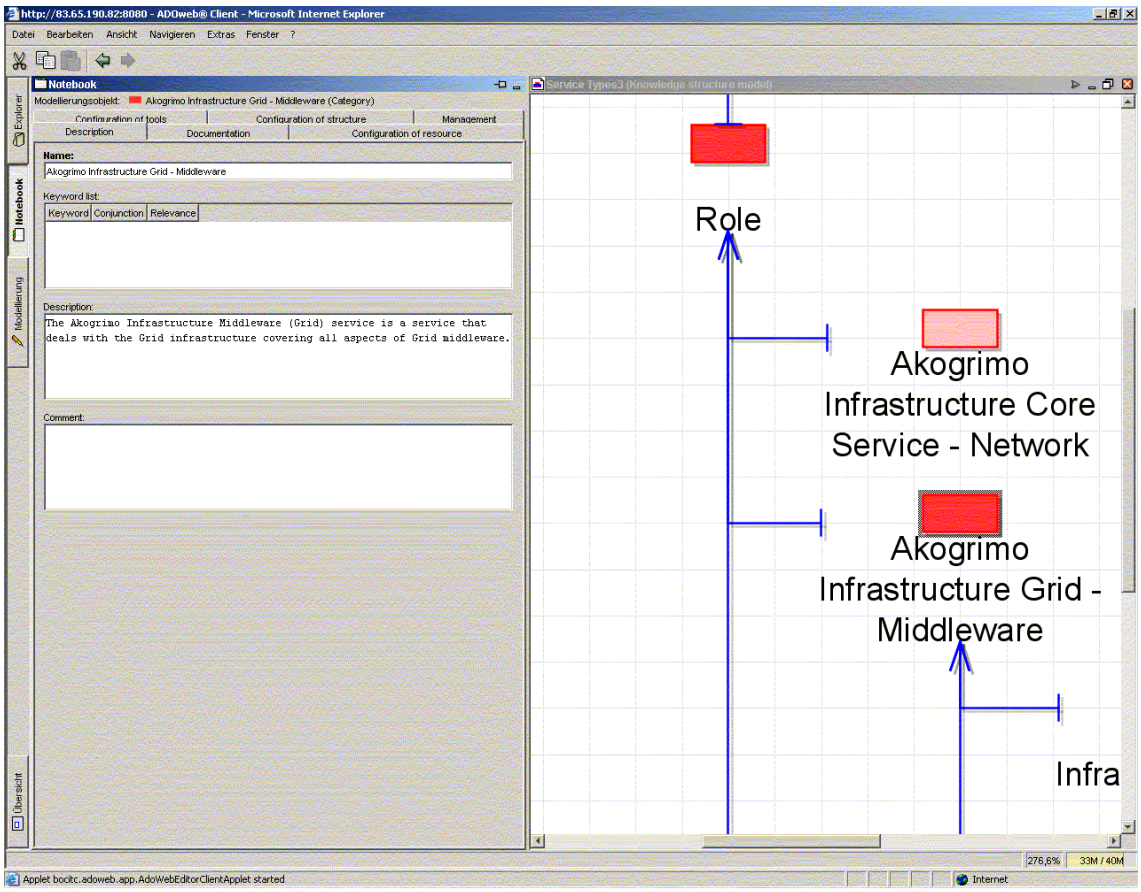


Figure 19 Screenshot of Web-Modeller Notebook

Figure 19 introduces the concept of the notebook. The chapters and attributes of the modelling objects are defined in the modelling method and displayed in the Web-Documentation using the details window.

The attributes of a modelling object are either:

- a text or number, that can be typed in,
- a reference to another model, where models or objects are referenced,
- a link, that refers to a document or program or
- a table that consists of a set of cells from the above attributes.

The information in the notebook can be updated by overwriting the existing text and by saving the model using the save option.

6.5 The Web-Import/Export

The Import/Export of models depends on the used modelling method. In the following section each Import/Export component is introduced.

6.5.1 UML™ Web-Import/Export

The UML™ Method provides two different import and export formats:

- Import and Export of ADONIS ADL files
- Import and Export according to the XMI specification

The usability of this Web-Component is similar to the other Web-Import/Export components. The following figure (Figure 20) shows the user interface of the Web-Import/Export window.

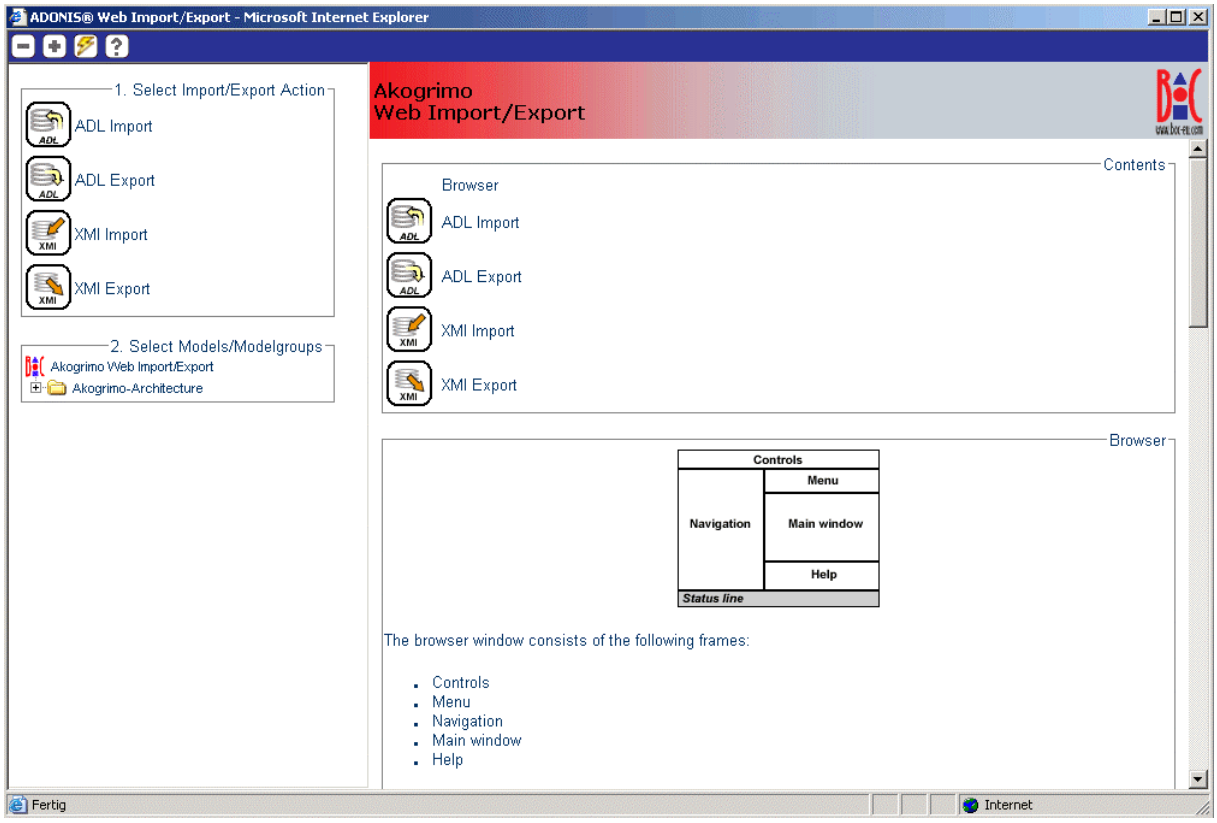


Figure 20 Web-Import/Export User Interface

The left part of the window enables the user, first, to select an Import/Export action. As already mentioned the interface provides ADL Import/Export and XMI Import/Export . After the user has chosen the action he wants to execute, the models or model groups can be selected. In case an import action is chosen the user has to select the group into which he wants to import his models. Otherwise, in case of an export action, the user has to choose models or model groups, which should be exported. The presentation of the model tree depends on the selected action. If ADL export is selected, every model will be displayed. In the case XMI export is selected only valid models will be presented. The welcome page also provides Import/Export support, since all Import/Export action and all necessary steps are explained. In the following Import/Export actions will be described.

6.5.1.1 ADL Import

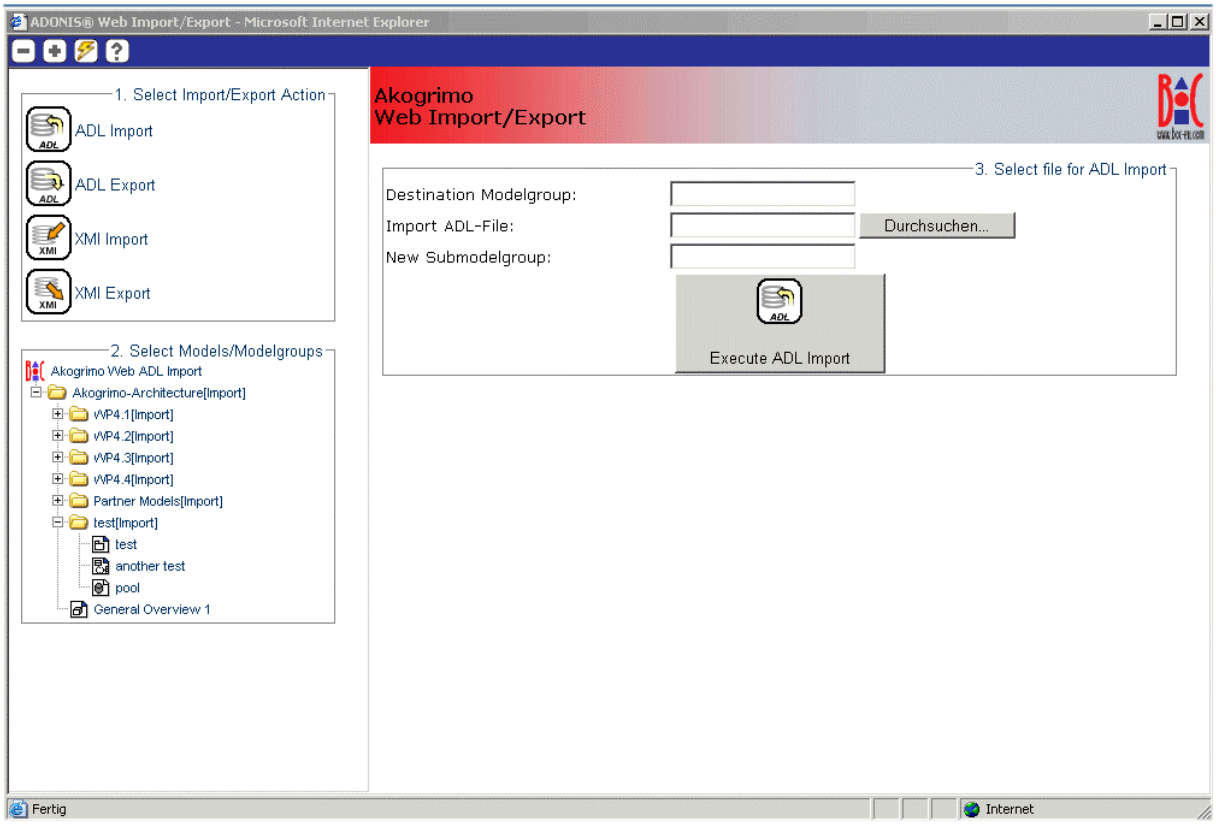


Figure 21 ADL Import

Figure 21 shows the ADL Import user interface. After the destination model group has been chosen, the user can select the file he wants to import. If desired a new submodel group can be created. The import will start after clicking on the "Execute ADL Import" button.

6.5.1.2 ADL Export

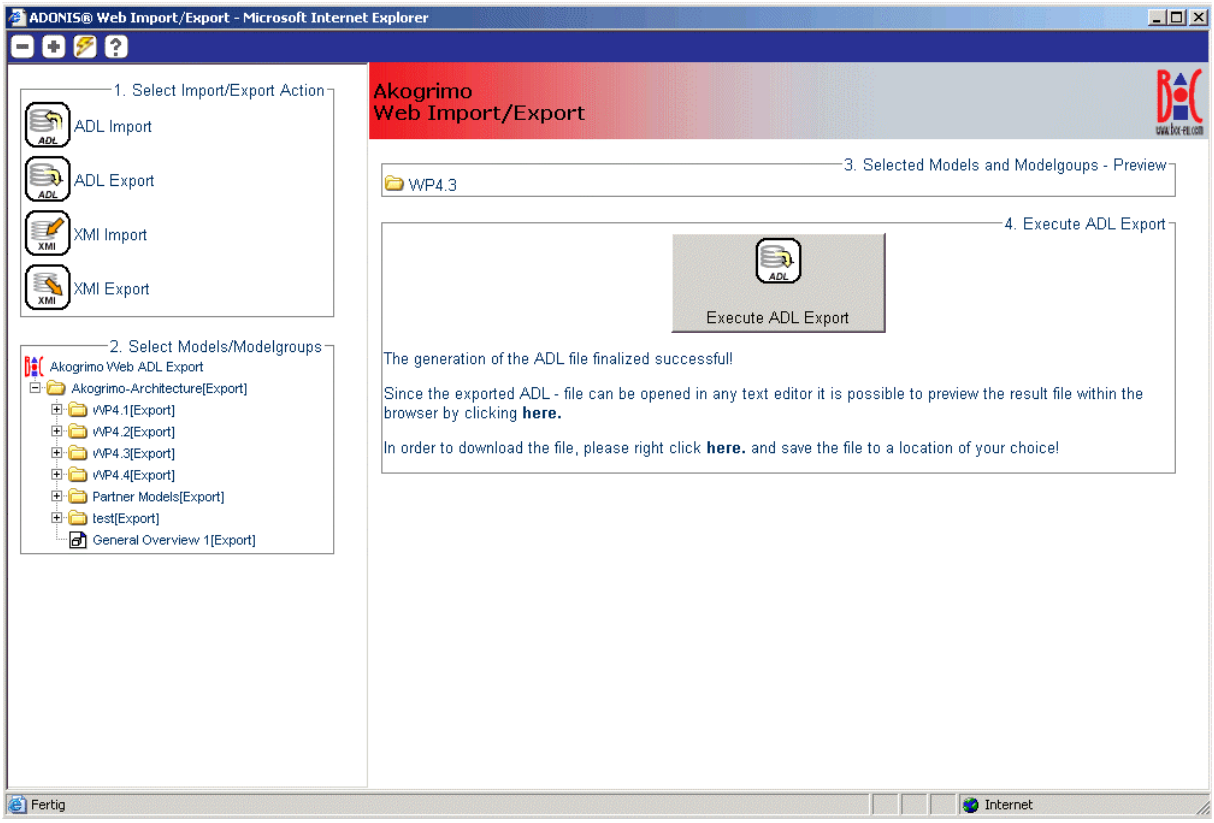


Figure 22 ADL Export

Of course also ADL files can be exported, as Figure 22 shows. These files can be imported into ADONIS®. First the user has to specify the models or model groups he wants to export using the model navigation tree. Clicking on [Export] next to the model or model group he wants to export will add the models to the selected model preview. After the selection of the models only a click on the "Execute ADL Export" is necessary to generate a corresponding ADL file. Finally the models can be previewed and downloaded using the standard Browser feature.

6.5.1.3 XMI Import

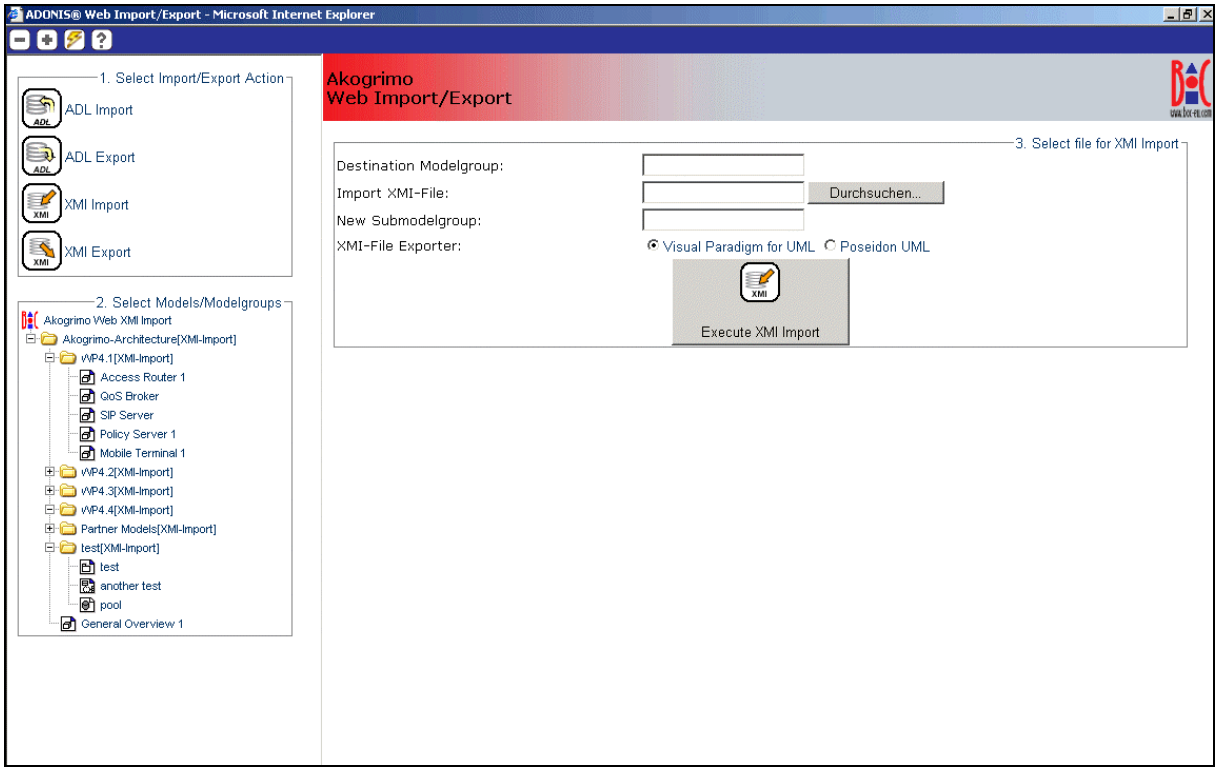


Figure 23 XMI Import

Figure 23 shows the user interface, corresponding to the XMI import. After the destination model group has been chosen, the user can select the file he wants to import. If desired a new sub-model group can be created. Then the correct XMI-File Exporter has to be selected, either Visual Paradigm for UML™ or Poseidon UML™. Finally, after a click on the "Execute XMI Import"-button, the import will be executed.

6.5.1.4 XMI Export

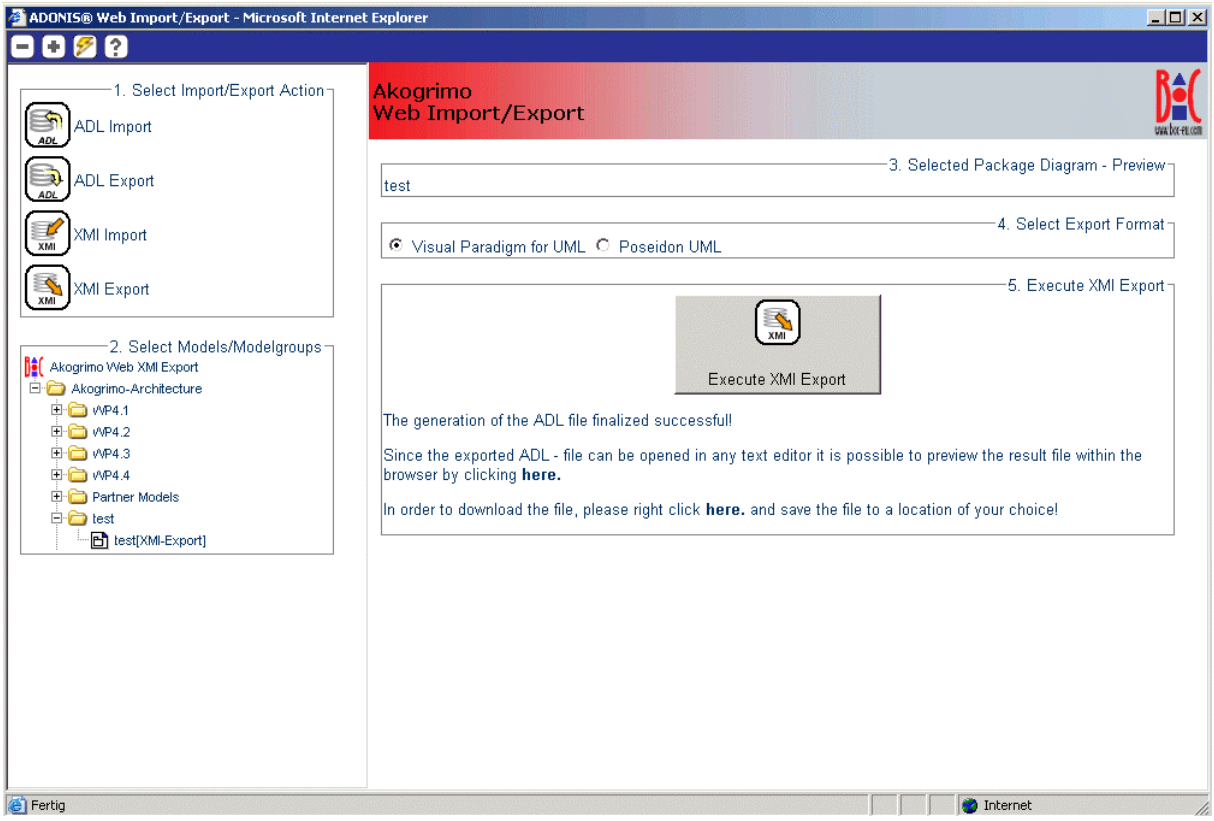


Figure 24 XMI Export

To export XMI-files the user has to choose the models he wants to export. Then the export format has to be chosen. After a click on "Execute XMI export" links will be provided to view and download the file, using the standard browser features. Figure 24 shows a screenshot of the XMI export user interface.

6.5.2 AKBPM Web-Import/Export

The AKBPM Web-Import / Export provides three different import and export formats:

- Import and Export of the ADONIS ADL files,
- Import of the EPC files and
- Export of the BPEL files.

The usability of this Web-Component is similar to the other Web-Import/Export components. The user can select models and model groups from the navigation tree on the left, by using the check boxes or by using the arrows.

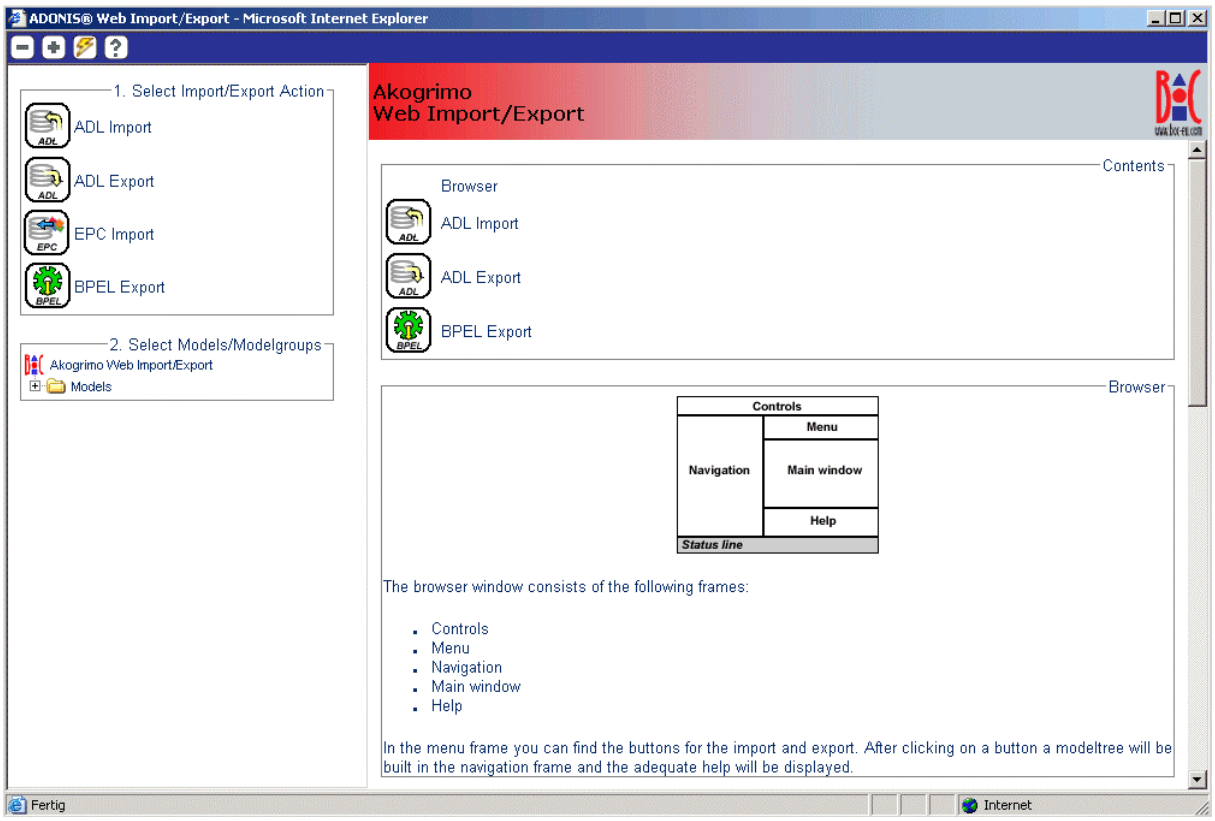


Figure 25 AKBPM Web-Import/Export User Interface

Figure 25 shows the AKBPM Web-Import/Export user interface. The AKBPM Web-Import/Export works basically like the UML™ Web-Import/Export. On the left side there is the navigation tree that enables the selection of model and model groups.

In the following all Import/Export actions, excepting ADL Import/Export, will be described. As the ADL Import/Export works exactly like the ADL Import/Export presented in section 6.5.1.1 and 6.5.1.2, the user is referred to this section for detailed information.

6.5.2.1 BPEL Export

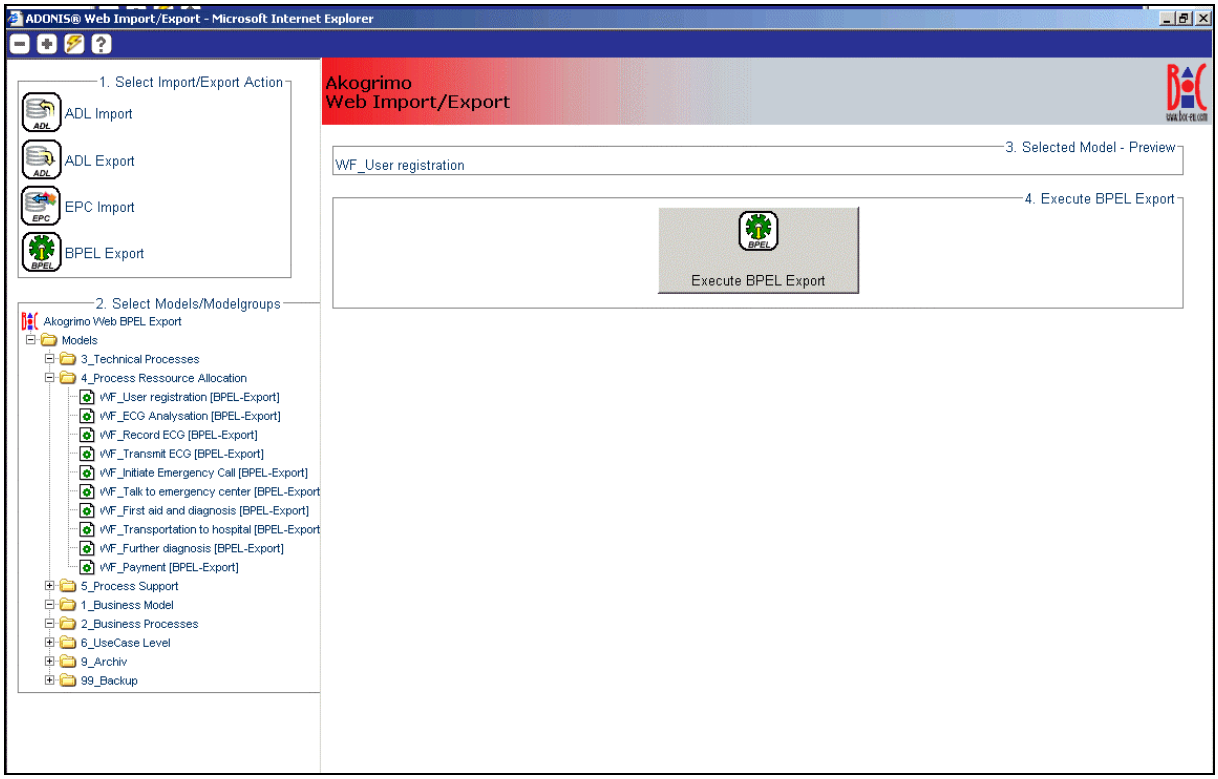


Figure 26 BPEL Export

After selecting the action BPEL Export on the left side of the window, Figure 26 shows the BPEL Export user interface. The models to be exported can be selected using the model navigation tree. In the case of BPEL Export, only the valid model types are listed.

Before the exporting the models, the user has to confirm the selection in the preview window. After confirmation the export file will be generated and provided as a link. The standard Browser feature can be used to download the file.

6.5.2.2 EPC Import

Another feature of the AKBPM Web-Import/Export is the possibility to import EPC models, which have been designed using other modeling tools.

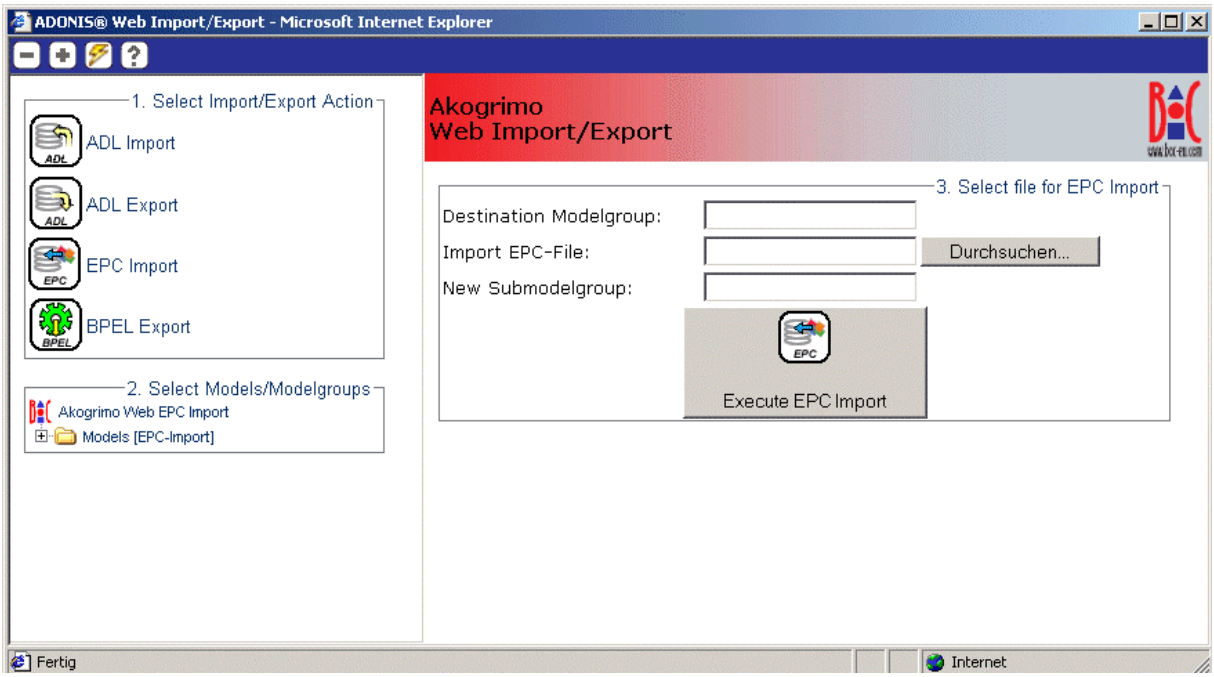


Figure 27 EPC Import

Figure 27 shows the EPC Import user interface. The functionality is the same as the functionality of the other import actions. After the destination model group and the EPC file, which should be imported, have been selected, the EPC import can be executed.

6.6 The Web-Document Repository

The Web-Document Repository collects documents that are required in the roadmaps. One of the key features of the roadmap is to navigate through the existing documents and provide recommended information.

These documents can either be kept:

- in an external document repository, like the BSCW and the Akogrimo Homepage,
- as an URL to a web-document or
- in case the document is not published in another repository, or it should be controlled by the Roadmap modeller, the document can be uploaded into the Web-Document Repository.

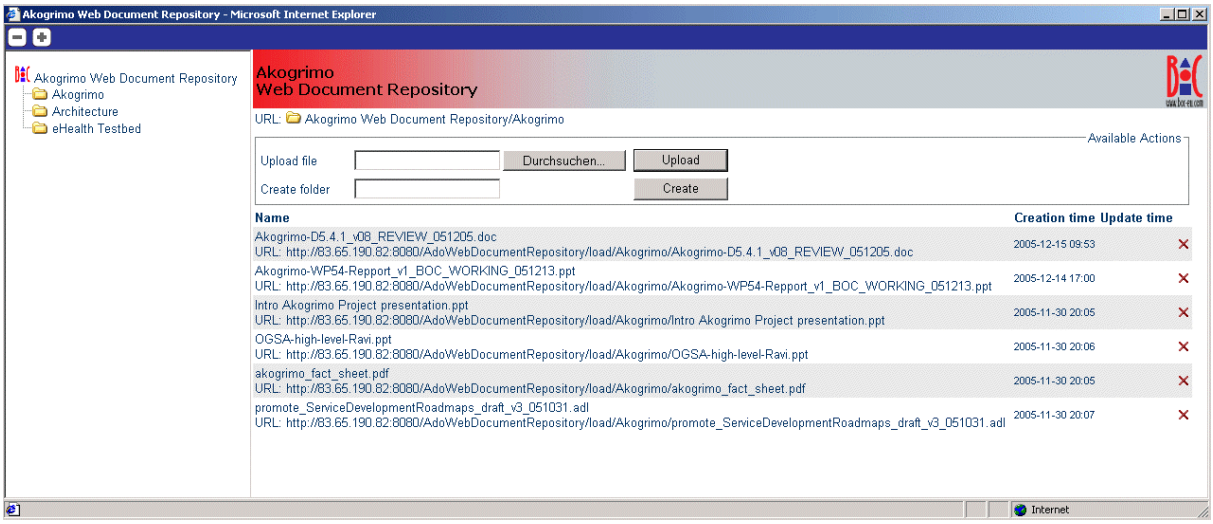


Figure 28 Screenshot of the Web-Document Repository

Figure 28 indicates the user interface of the Web-Document Repository.

The left side provides a navigation tree, to navigate in the directories of the repository, whereas the right side provides the list of files in the repository.

The files can be deleted, downloaded using the standard Browser functionality to save files or using the file dialog to upload files.

The link to the file is provided to enable to copy the link and use the link in the Web-Modeller by pasting the link. The document can therefore be either addressed via its file id, but preferable via the repository path, that enables the upload of new versions and guarantees stable links although the file id may be changed.

To upload a file the user has to select a folder into which he wants to import its file. If desired a folder can be created to ensure clarity.

The global tool icons are the same than in the general introduction.

7 Summary

The Graphical Evolution Tool (GET) is a new generation of a build time environment that supports the distributed specification, implementation and testing on various layers in a service-oriented architecture integrating classic software development with knowledge management in an extremely challenging scenario by supporting a complex and highly sophisticated software research project. The GET is flexible and is able to guarantee cooperation between partners, because of the included components, e.g. the Import/Export component. Moreover the GET is intuitive to use and different access rights can be assigned to different users depending on their requirements.

The user requirements have been collected based on proposed Akogrimo usage scenarios and the experience of building modelling environments.

The result was the definition of Web-Components for the first version of the Graphical Evolution Tool. The focus of the first step was to provide a powerful and consistent modelling tool to specify the application requirements that are developed to run on top of Akogrimo and to support application relevant adaptations of the platform. The service development of these requirements is supported by using a process-oriented knowledge management approach in making expert know-how from Service Developers explicit via a graphical representation.

Therefore the tool has three key objectives that are listed according their priority:

First the provision of a powerful Web-Documentation Component to view the roadmap and to enable the navigation within the Akogrimo knowledge that exists in form of documents at various repositories.

Second is the provision of a distributed Web-Modelling environment that integrates stand-alone versions of the ADONIS® model editor and the Web-Modeller of Akogrimo by a wide range of partners. This integration includes transformation of the generated models to be used in external tools.

Third to encourage the Akogrimo Roadmap user to think and analyse the service in a wider context including the application scenario specification on different process layers. Three different graphical modelling methods have been integrated in one single tool to provide a coherent support for all aspects with one platform.

This document is seen to accompany the usage of the Graphical Evolution Tool that runs at:

- **URL:** http://83.65.190.82:8080/
- **User:** AkogrimoRoadmap
- **Password:** AkogrimoRoadmap-FP6-2003-IST-2-004293

8 References

[1] Akogrimo Deliverable 5.4.1, Roadmap to Akogrimo convergence

[2] Projektmanagement: http://www.kfunigraz.ac.at/inmww/Projektmanagement_SS04.pdf, access: 01.09.2005

[3] Die Welt des Projektmanagements: http://www.rkw-bw.de/Info/files/p05-0067.pdf, access: 01.09.2005

[4] Software Engineering Projekte als Grundlage für Qualitätsmanagement: http://qse.ifs.tuwien.ac.at/courses/skriptum/download/02P_SEPM_wid_2004009.pdf, access: 01.09.2005

[5] IST-2002-004293 Akogrimo – Annex I, pp 104

[6] Akogrimo Internal Deliverable 5.4.1, Initial Roadmap to Akogrimo Convergence

[7] Akogrimo Internal Deliverable 3.2.2, The Business Modelling Framework,

[8] Process-oriented Knowledge Management System based on KM-Services: The PROMOTE® approach, International Journal of Intelligent Systems in Accounting, Finance & Management Published online in Wiley InterScience (www.interscience.com) Int. J. Intell. Sys. Acc. Fin. Mgmt. 11, 253-267, 2002

[9] Larman Craig, UML 2 und Patterns angewendet – Objektorientierte Softwareentwicklung, mitp Bonn, 2005

[10] UML, http://www.omg.org, access: 01.12.05

[11] Kühn H., Junginger S., An Approach to use UML for Business Process Modeling and Simulation in ADONIS®, Proceedings of the 13the European Simulation Multiconference (ESM 99), Warsaw, Polan 1.4, 1999

[12] Jeckle M, UML 2.0 Die neue Version der Standardmodellierungssprache, http://www.jeckle.de/files/umltutorial.pdf, access: 23.0905

[13] PROMOTE, http://www.boc-eu.com/bochp.jsp?file=WP_582571cc1ed802de.b05236.f5980937ac.-7fdc, access: 01.09.05

[14] Woitsch R., Karagiannis D., Proceedings of the I-Know 2004 from the Special Track BPOKI'04 30th of June, Graz Austria 2004

[15] Woitsch R, Process-Oriented Knowledge Management: A Service Based Approach, PhD Thesis, University of Vienna, 2004