D 5.4.1

Roadmap to Akogrimo Convergence

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 5.4.2	Definition of Roadmap to Akogrimo convergence
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 5.4.2 and 5.4.3.

Contributors:	***Reviewers:***
Aniello Rovezzi (CRMPA) Antonios Litke (ICCS/NTUA) Giuseppe Laria (CRMPA) Josep Martrat (ATOS) Julian Gallop (CCLRC) Margit Schwab (BOC) Robert Woitsch (BOC) Wilfrid Utz (BOC)	Burkhard Stiller (UniZH) Christian Loos (UHOH) Josep Martrat (ATOS) Mario del Campo Melgar(TID) Martin Hafner (UHOH) Robert Woitsch (BOC)

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

Version	Date	Authors	Sections Affected
0.0	November 02, 2005	Robert Woitsch	All – initial document
0.2	November 08, 2005	Robert Woitsch	ID 5.4.1 Integration
0.3	November 18, 2005	Robert Woitsch	First Complete Working Document
0.4	November 23, 2005	Robert Woitsch, Margit Schwab, Julian Gallop, Josep Martrat, Aniello Rovezzi, Giuseppe Laria	Integration of Partner Input, WF – Annex added, Tool – list integrated, Chapter 3 definition completed
0.5	November 25, 2005	Antonis Litke, Robert Woitsch, Josep Martrat	Skill Profiles, Content adaptation in Chapter 3, VO-Management Draft in 6.5, Method integration
0.6	December 02,2005	Robert Woitsch, Margit Schwab, Julian Gallop, Josep Martrat, Antonis Litke ,Aniello Rovezzi, Giuseppe Laria	Review and comments of all chapters
0.7	December 05,2005	Robert Woitsch, Margit Schwab, Josep Martrat	Merge of Innovation list of WP 6.3 and Feature Map, Skill Input adaptation, final formatting
0.8	December 21,2005	Robert Woitsch, Margit Schwab, Josep Martrat, Julian Gallop, Antonis Litke ,Aniello Rovezzi, Giuseppe Laria	Adaptations required by the internal review process in all chapters, according the review reports.
0.9	December 30,2005	BOC team	Final Review of all sections, consistency checks, spell checks.
1.0	December 30, 2005	Robert Woitsch, Margit Schwab	Finalisation and approval

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 that want to develop services or applications on top of Akogrimo by providing methods, tools and knowledge. 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.

Four phases have been identified that are supported by the roadmap:

1. The **Requirement Specification Phase** is conceptually supported by a business process oriented method that enables the derivation of requirement specification.
2. The **Implementation Phase** is supported by providing knowledge for developing services in Akogrimo. A graphical representation of knowledge supports the navigation within Akogrimo and offers recommended documents.
3. The **Testing Phase** is supported by indicating the reference testing procedure.
4. The **Roll Out Phase** is supported by indicating the VO-Processes.

The Akogrimo Roadmap uses three modelling methods to support the above four phases.

For the Requirement Specification a Business Process Management Method has been selected to describe the Business Model and the Business Processes on a high level to enable the derivation of system requirements in the form of so-called Business Service Sheets. A new modelling method has been developed for the special requirements of Akogrimo called the **Akogrimo Business Process Management (AKBPM)** modelling method.

For the architecture and design of the Roadmap the well-known Unified Modelling Language was selected to support the specification of the requirements and enable the design of the system. There is a synchronisation between the above introduced requirement modelling method AKBPM and the system design in **UML™**.

The implementation of services or components for, or on top of Akogrimo is described using the knowledge management method **PROMOTE®** that builds on the experience of Akogrimo partners in the application support layer and in the grid infrastructure layer of the Akogrimo architecture explicitly by modelling best practice processes and visualising the knowledge structure. The available knowledge in Akogrimo is structured according to:

1. The **Traditional Approaches vs. Akogrimo Map**, which describes the gap between traditional approaches and the Akogrimo approach by pointing out the innovative elements of the Akogrimo platform and identifying the differences of advanced platform features.
2. The **Technology Map**, which describes the key technology in Akogrimo and points out the relevant IT-Infrastructure, standards or tools that are necessary to develop an application on top of Akogrimo.
3. The **Application Support Layer** and the **Grid Infrastructure Layer**, which describes the relevant elements of the Akogrimo platform that have to be considered when developing an application on top of Akogrimo.

Four individual development processes of Akogrimo, services were introduced that were then integrated into a reference roadmap that presents **the roadmap to Akogrimo convergence**[1].

[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

AAA	Authentication, Authorization and Accounting
AC	Activity (Phase of the project)
AKBPM	Akogrimo Business Process Management (Method)
ATTRACT	Method out of the FP5 – IST project "Advanced Teleworking techniques & Tele-services for insuRance Agents & CusTomers" (Method)
BMF	Business Modelling Framework
CASE	Computer Aided Software Engineering
CMS	Content Management System
CVS	Concurrent Version System
CVS	Concurrent Versions System
D	Deliverable (official)
EPC	Event Driven Process Chain (Method)
GlT	Globus Toolkit
HTML	Hyper-Text Markup Language
ID	Internal Deliverable
MSCs	Message Sequence Charts
NGG	Next Generation Grid
OASIS	Organization for the Advancement of Structured Information Standards (http://www.oasis-open.org)
OGSA	Open Grid Service Architecture
PROMOTE®	Process Oriented Methods and Tools for Knowledge Management (Method)
SAML	Security Assertion Markup Language from OASIS
SDL	System Definition Language
SLA	Service Level Agreement
UML	Unified Modelling Language
VO	Virtual Organisation
WP 4.x	Workpackage 4.1, 4.2, 4.3 and 4.4
WP 5.x	Workpackage 5.1, 5.2, 5.3 and 5.4
WP	Workpackage
WSRF	Web Service Resource Framework
WSRF.NET	Web Service Resource Framework and Microsoft .NET (http://www.cs.virginia.edu/~gsw2c/wsrf.net.html)
XMI	XML Meta Data Interchange
XML	eXtended Markup Language

1 Introduction and Problem Statement

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

This document describes a new approach – the so-called Akogrimo Roadmap - to support the development of applications on top of Akogrimo in the application development scenario and the integration of services into the platform in the platform development scenario.

The Roadmap has been developed by a set of graphical modelling approaches that has been adapted to enable an Akogrimo specific integration of model-based methods supporting the development cycle of applications or services in Akogrimo.

Four phases of the software development process - Requirement Definition Phase, Implementation Phase, Testing Phase and Roll Out Phase - have been identified to demonstrate the support of the Akogrimo Roadmap.

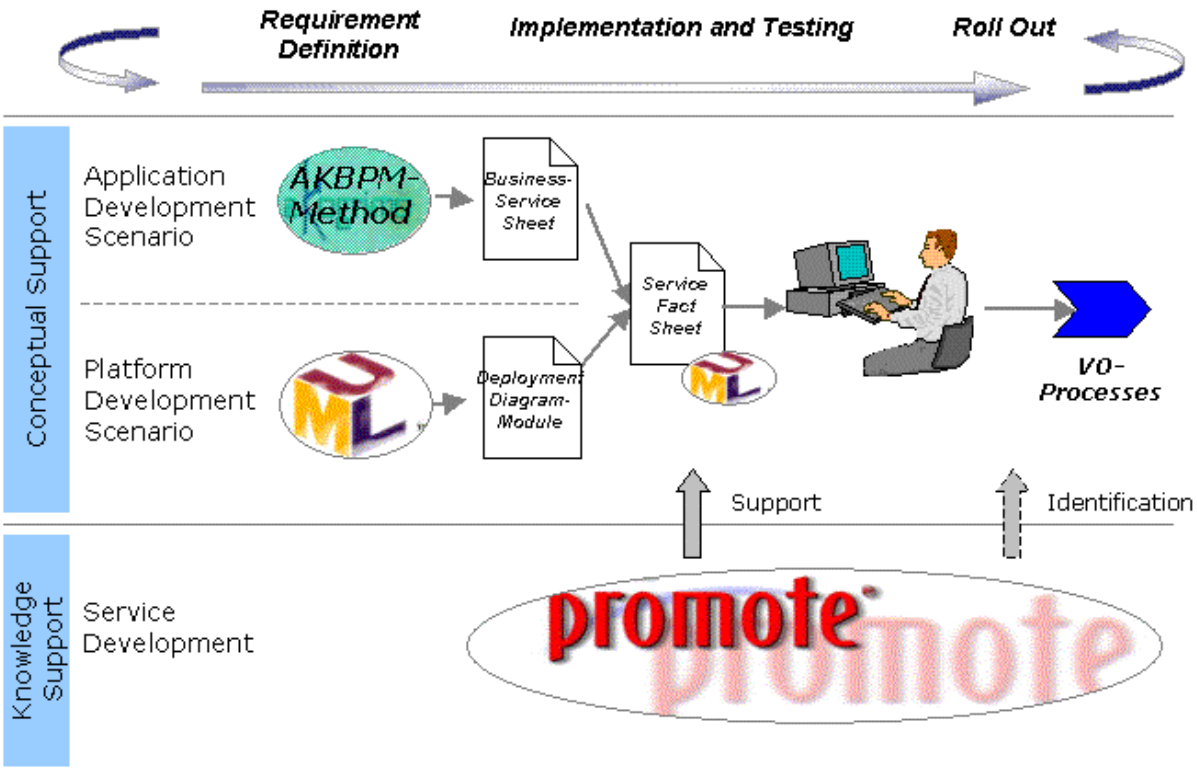


Figure 1 The Roadmap Overview

Figure 1 identifies how the Akogrimo Roadmap supports the application or service development.

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. This method has been applied for the eHealth scenario to demonstrate the procedure. 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 the UML™ model types to describe the platform and the services. The deployment diagram identifies the services.

The **Implementation and Testing Phase** starts with the definition of the Service Fact Sheet, which 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 at defined states of the development process. The Akogrimo Roadmap supports this phase by providing knowledge for the service development of services in Akogrimo. For the provision of this knowledge the modelling method PROMOTE® has been selected to describe the experience of partners with the implementation of Akogrimo services.

This part of the Akogrimo roadmap therefore supports the developer by providing a guideline of how to proceed in developing a service and by providing links to documents for relevant topics.

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

In the following the Akogrimo Roadmap is seen as the integration of the three graphical modelling methods – AKBPM, UML™ and PROMOTE® - to support the four phases of software development – Requirement Definition, Implementation, Testing and Roll-Out – in an appropriate way.

The requirement definition phase for applications is supported by a procedure that derives service requirements on the basis of the application scenario. This part of the Akogrimo Roadmap is also called Application Specification Roadmap.

The requirement definition phase for platform development is supported by UML™, which enables the description of the architecture. This part of the Akogrimo Roadmap is called Platform Documentation Roadmap.

The implementation, testing and roll-out phase is supported by knowledge of partners represented in PROMOTE®. This part of the Akogrimo Roadmap is called Service Development Roadmap.

In the following the overall architecture of the Akogrimo Roadmap, the relationship between this deliverable and deliverable 5.4.2, the goals of the workpackage, and the identification of the different aspects of the Akogrimo Roadmap are discussed.

1.1 The Overview of the Roadmap

This deliverable is closely related to D 5.4.2 (Graphic Evolution Tool)[1] that has been generated in parallel. The following section briefly distinguishes these two deliverables and points out their relationship. The following text focuses on:

- the explanation of roadmaps,
- the analysis for enabling applications to use Akogrimo and
- introducing methods to enable a graphical model based support of application and service development.

The related D 5.4.2 (The Graphical Evolution Tool) focuses on:

- usage scenarios applying the roadmaps,
- identification of user roles and required access rights and
- introducing the components of the Graphical Evolution Tool, the technical infrastructure and Akogrimo relevant adaptations of the Web-Modelling environment.

There are three conceptual layers in this workpackage as indicated in Figure 2.

The content layer deals with the knowledge that is necessary to develop services and Akogrimo relevant topics. The method layer is concerned with the selection, adaptation and provision of graphical representation methods to generate roadmaps. The technology layer provides the necessary technological environment to generate and use roadmaps in Akogrimo.

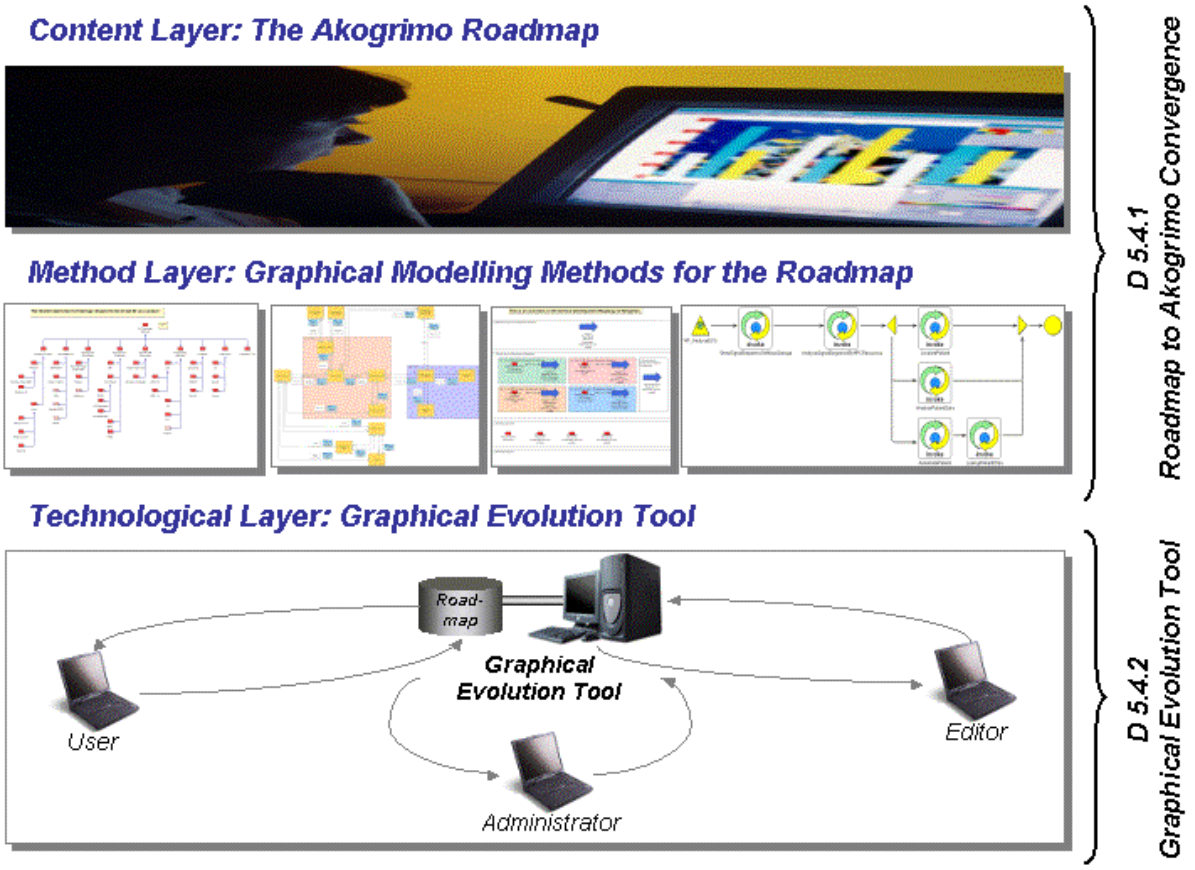


Figure 2 Identification of Deliverables in WP 5.4

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

1.2 Goal of the Workpackage

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 [2]. 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) [3].

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) [4] 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® [5] 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 Structure of the Document

This deliverable is concerned with the roadmap to Akogrimo convergence that consists of a set of graphical models, documents and guidelines. The Akogrimo roadmap is published in the Internet via the Graphic Evolution Tool to enable a navigation through the Akogrimo content, and this document describes:

- the overall structure of the Akogrimo Roadmap,
- describes the analysis of approaches on how to use Akogrimo,
- discusses the selected modelling methods and
- introduces relevant models of the Akogrimo Roadmap.

Both deliverables are based on the ID 5.4.1 (The Initial Roadmap to Akogrimo Convergence) that provides a draft description on the overall structure, an initial view on analysing approaches for enabling Applications to use Akogrimo, a draft description of the used methods, an initial description on the Graphic Evolution Tools as well as initial models for the Roadmap.

In the following the procedure of the Roadmap generation is briefly introduced and each step is allocated to a chapter to introduce the relation between the chapters.

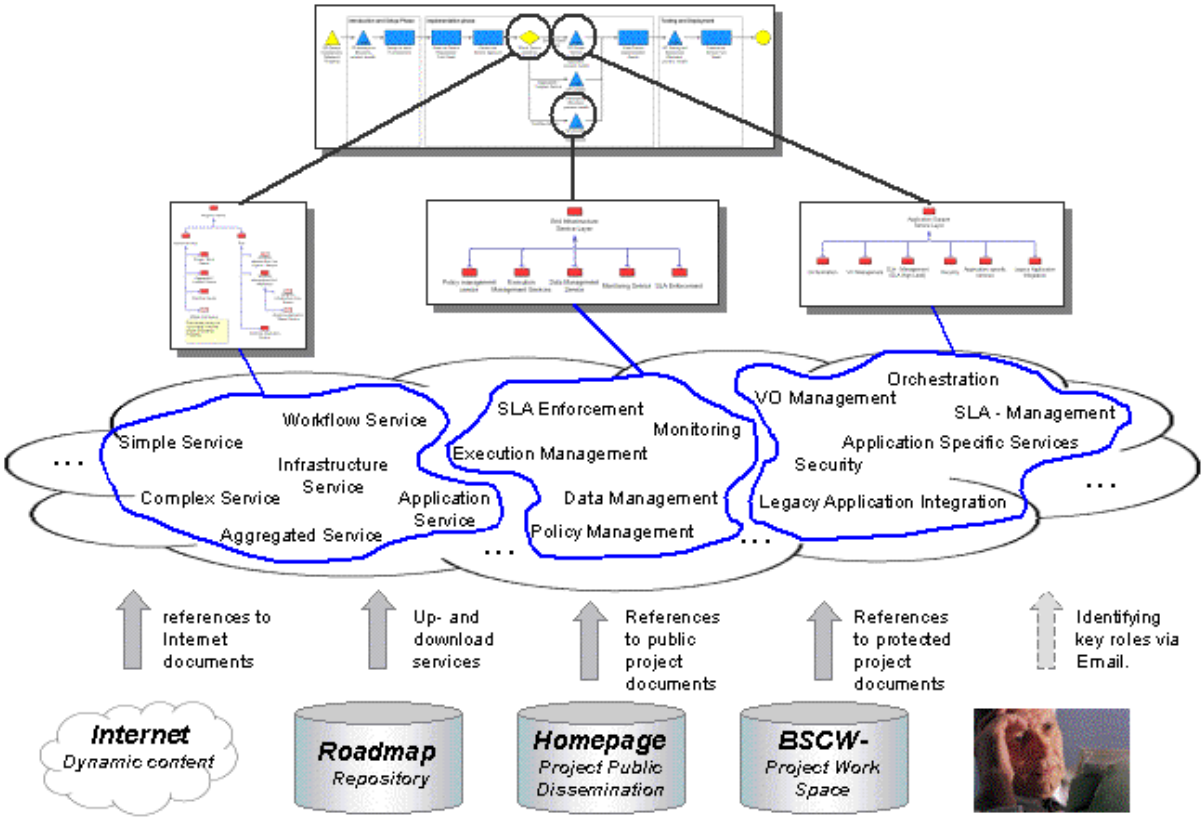


Figure 3 Overview on the Roadmap Usage Concept

Figure 3 identifies the major aspects of the Akogrimo Roadmap to support the development of applications or services in Akogrimo with knowledge that has been gained from partners experience.

The identification of the major aspects and the overall structure of the Akogrimo Roadmap is described in Chapter 2. The relevant concepts of the Akogrimo Roadmap are defined, the overall architecture is introduced and the usage of the Roadmap is pointed out. The above figure will be

explained in more detail and in a more formalised way to introduce how the Akogrimo Roadmap can support the different user-groups during the different phases of software development.

The identification of the major aspects of Akogrimo includes the identification of so-called knowledge sources – the source from which the knowledge is currently available. As depicted in Figure 3 there are five knowledge sources that contains the current existing knowledge about Akogrimo.

To start the acquisition of knowledge, the so-called knowledge object – the object whose knowledge is considered – has to be identified. Here the Akogrimo platform has been identified as the knowledge object that has to be analysed.

In Figure 3 the knowledge object "Akogrimo" is indicated as a cloud covering a set of Akogrimo related terms. The cloud-symbol was used to point out the fact, that the knowledge object is dynamic and fuzzy. The dot symbols were used to point out that the terms mentioned in the figure are only indicative, a complete list of the identified terms are represented in Chapter 3.

Chapter 3 describes the content of the roadmap, by first analysing the relevant starting points before developing an Akogrimo service. Second the relevant content of the Akogrimo architecture is pointed out. Finally the technological infrastructure of the Akogrimo platform is identified including recommended documents and links. This analysis has been made under the view of an application developer that is interested in using the Akogrimo platform to build or adapt an application on top of it. Therefore the WP-layer 4.3 and 4.4 were described during the content analysis.

The identified terms in Akogrimo are briefly described, justified and related to one of the identified knowledge sources.

Figure 3 depicts that the identified terms of Akogrimo are grouped and represented in the form of graphical models. Before theses terms can be graphically represented to enable a simplified navigation through the Akogrimo content, the appropriate graphical modelling method has to be selected. Chapter 4 introduces an appropriate graphical modelling method. A more detailed view on the selection of the appropriate modelling methods and an introduction in the modelling is introduced as an excurse in the appendix.

As indicated in Figure 3 the Akogrimo content described in Chapter 3 is then presented in the form of graphical models using the graphical modelling methods of Chapter 4 to generate the graphical representation of the Akogrimo Roadmap. The results of modelling the Akogrimo Roadmap are described in Chapter 5. First the individual roadmaps are introduced that have been modelled on bilateral face-to-face meetings and have been reviewed via telephone interviews. Second the identified content, that has been analysed in Chapter 3, is presented using the knowledge structure model. Finally the Akogrimo Reference Roadmap that has been generated out of the individual roadmaps via workshops is displayed to give a complete picture on the service development.

The Akogrimo Roadmap is currently published via the Graphic Evolution Tool at:

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

The corresponding deliverable 5.4.2 (The Graphic Evolution Tool) introduces the tool, the architecture and the IT-infrastructure that enables the Akogrimo Roadmap to be published.

2 The Usage Concept of the Roadmap

2.1 Definition of terms

2.1.1 Roadmaps

A Roadmap is often used to designate a (simplified) policy, strategy or action plan.

- A roadmap is a structured (and often graphical) representation of a plan to reach a goal. The goal of a roadmap is to identify (critical) milestones and place them on a timeline.
- Roadmaps are used in all kinds of fields for example economics, politics, technology etc.
- Technology roadmapping is often used in practice to support strategic and long-term planning.

Technology roadmapping [6] is a (technology) management tool that attempts to plan and forecast the necessary steps toward achieving one or more (technology) goals.

The value of a technology roadmap includes:

- communicating vision,
- encouraging collaborative thinking,
- garnering necessary resources to solve technology challenges,
- creating contingency approaches and
- consensus view for decision making.

One of the most common extensions of the technology roadmap is to link it to product roadmaps and market roadmaps to provide the complete picture of "What, Why, and How" in relation to the achievement and delivery of a particular technology goal.

That means beside the important questions of “What is the goal?” and “What is the way to reach the goal(s)?” the concept of a roadmap also tries to focus on supporting topics like necessary resources, and a common mind on decision making and collaboration.

During a partner discussion the term “roadmap” as it is used in this workpackage has been defined as:

“The **roadmap** that collects all steps – based on previous partner experiences – necessary to develop services/components of the Akogrimo infrastructure. Because in Akogrimo we deal with different technologies (grid oriented and network oriented) there are some differences in the steps on the basis of service nature (it could be a web service or a network service). Some of these steps depend on the underlying framework we plan to use and the programming language adopted, because the experience gained is related to the platform as well.

The roadmap should represent in a schematic way (flow blocks) the steps necessary to develop a service into the candidate development environment. Moreover it should summarise:

- experience requirements and how to get them if necessary (e.g. material, tutorial, links…)
- for each block depicted in the roadmap schema a detailed description to understand and to study in depth the arguments if necessary.”

2.1.2 Modelling Method / Language

The language for representing a model is described by its meta model. The concepts of a modelling method are described by elements such as classes, relationships, attributes, and behaviour. A model type describes a concrete modelling method, e.g. a business modelling method. Linking and relating the model types, forms a set of interrelated modelling methods which describe a certain domain under consideration. [7]
In this document the term modelling method and modelling method is used synonymously.

2.1.3 Method

A method consists of three essential elements: the modelling techniques that should be applied, the techniques and algorithms that should be used and the procedure model. The method-engineer is responsible for the definition of the method. [8]

2.1.4 Graphical Evolution Tool

The Graphical Evolution Tool is a distributed Web-Modelling environment that is specially configured and adapted to the needs of Akogrimo. The distributed Web-Modelling environment consists of graphical model editors that are locally installed as stand alone versions with the possibility to upload models into the web-portal and of web-modellers that run within an Internet Browser. To support the modelling environment, there are components to view the models, to upload and download the models and a repository to integrate documents into the models.

2.1.5 Service

The term service is generally used within the Roadmap as software parts of the Akogrimo platform. A detailed distinction is introduced in chapter 3 and shown in chapter 6.

For a better readability of the text, phrases like “development of Akogrimo services” or “development of services within Akogirmo” are used instead of writing the more precise phrase “development of either Akogrimo Infrastructure Services or Akogrimo Application Services that run on top of the Akogrimo platform as defined in chapter 3”.

2.1.6 Roadmap User

In this context the term “user” refers to the “roadmap user” that is seen as the person that interacts with the Akogrimo roadmap models to better navigate in, learn about , apply or develop for Akogrimo.

Roles like Application Provider, Application Configurator, Platform Provider, Platform Configurator, Service Developer and Service Developer Consultant have been identified to interact with the models to either specify application requirements, describe the platform requirements or implement services for Akogrimo.

2.2 Roadmap Support for Users[2]

This section introduces a simple user concept for Akogrimo Roadmap users based on different phases of the service development. First the development phases are allocated to generic user roles to generally introduce different roles and tasks during the development process. Second the generic concepts are mapped to a simple Akogrimo user role concept that is used for initial configuration of the Akogrimo Roadmap.

The goal of this section is an introduction of different phases of the Akogrimo roadmap, different user roles and to point out different user support.

Four development phases have been identified as:

1. Requirement Definition Phase
2. Implementation Phase
3. Testing Phase
4. Roll Out Phase

The above phases have been used to provide a simple framework to better classify the different user support from the roadmap independently, if the phases are executed strictly hierarchically, concurrently or looping according to the applied software development methodology. The support of the Akogrimo Roadmap does not require the above phases but can be split, but to show the different levels of support and to demonstrate the completeness of the Akogrimo Roadmap they have been found useful.

2.2.1 The Requirement Definition Phase

Generic Description:

In the Requirement Definition Phase, all business qualifications and requirements are collected and put together in context to define the overall scope and scale of the software to develop. This typically is done by the **Project Manager** and the **Head of Business Specification**.

The next step has to be working out a detailed business concept where the given framework has to be broken down in all needed outcomes and goals of the project. It also includes the rough definition of tasks and critical success factors. Hereby the **Project Manager** and the **Head of Business Specification** are supported by the **Business Specification Team** and the **Head of Development**.

The requirement definition phase is critical because if the definition is complete and comprehensive the implementation will be more accurate. As well the results of the roll out will have a higher rate of acceptance from the application users.

Akogrimo Relevant Description:

The following user roles for the Akogrimo Roadmap have been identified that work in the requirement definition phase.

1. Application Provider:

   The application provider is seen as an interface to external customers by providing domain specific applications. The application provider either consumes existing services or develops new services if necessary to build a complete application.

2. Platform Provider:

   The platform provider is seen as an interface to external customers who configures an Akogrimo platform and provides it to an operator.

[2] The following chapter is based on expert interviews of partners as well as on literature compare [9], [10], [11]

3. Application Configurator / Platform Configurator

   The application configurator is seen as a partner that selects appropriate services and specifies workflows to compose applications or platforms. The main task is to decide if the required functionality can be covered by existing services, in that case a workflow or complex service is composed, or if new services have to be developed, in that case UML diagrams to specify the requirements are defined.

Akogrimo Roadmap Support:

The Akogrimo Roadmap supports this phase by providing methods to define:

- the business model that describes the application scenario on a business view,
- the business process model that enables the detailed description of the application scenario on different process levels,
- the service requirements via Business Service Sheets that are linked to the previous mentioned business process models and
- UML™ models that enable the specification of services and platforms on a more technical level.

The Akogrimo Roadmap users of this phase are supported by graphic modelling methods that enable process-oriented description of the application scenario and the specification of requirements using the so-called Business Service Sheet on a higher level and UML™ diagrams on a technical level.

2.2.2 The Implementation Phase

Generic Description:

The first step of the implementation phase will be to break down the business case into technical parts that identify software interfaces and additional technical description. A more detailed analysis of the mapping from business requirements to software components is described in chapter 4.1. In this phase the **Developer** has to work together with the **Technical Specification Team** (e.g. the workflow modeller) supervised by the **Head of Development**.

The next phase is the transferral of the requirements into code – the programming by the **Developer**. During the development the **Developer** identifies common pitfalls and mistakes as well as new findings to supply it to its co-workers/colleagues.

Akogrimo Relevant Description:

1. Service Developer:

   The service developer is seen as a software developer that gets a service requirement description and implements the service.

2. Service Development Consultant:

   A Service Development Consultant is therefore seen as an Akogrimo core partner that is well aware of the Akogrimo architecture and can provide expert knowledge in specific domains.
   The service development consultant can be contacted by a service developer that needs support in developing a service for Akogrimo.

Akogrimo Roadmap Support:

The Akogrimo Roadmap supports this phase by providing methods to define:

- the linkage between Business Service Sheets and technical Service Fact Sheets,
- UML™ models to enable the specification of services and platforms on a more technical level and

- Knowledge of service developers by describing a reference process how to implement Akogrimo services.

The Akogrimo Roadmap users of this phase are supported by graphic modelling methods that bridge the gap between the business requirement definition and the technical specification to start the implementation of Akogrimo services.

Graphical representation of knowledge via knowledge models support the user by a reference process on how to implement an Akogrimo service.

2.2.3 The Testing Phase

Generic Description:

The testing phase is characterised by three different kinds of tests and corresponding roles.

The first stage is the testing of the defined program packages and their interfaces according to the technical specification out of the implementation phase. The **Business Testing Team** reports directly to the **Technical Specification Team** in case of any malfunction that is found. The focus in this phase is to test the behaviour of the system in the business context, therefore a detailed knowledge of the end user scenario is required. It's also possible that the **Business Testing Team** discovers missing functionality that either was defined but not developed or was not considered in the definition phase.

After the testing of the single packages the **Integration Testing Team** is running overall integration tests. The interplay of the single packages and the interfaces are tested and business cases are stepped through to simulate the behaviour under real world conditions.

In the end the **System Testing Team** is running performance and load tests under different system environments to assure that going live is possible with the expected amount of transactions.

Akogrimo relevant Description:

In Akogrimo Roadmap there are no special testing roles defined, so in this phase are the same user roles that were identified in the previous phase, the Service Developer and the Service Developer Consultant.

Akogrimo Roadmap Support:

The Akogrimo Roadmap supports this phase by providing methods to define:

- Knowledge of service developers by describing a reference process how to test and deploy Akogrimo services.

The Akogrimo Roadmap users of this phase are supported by a graphical representation of knowledge via knowledge models to support the user how to test and deploy an Akogrimo service.

2.2.4 The Roll Out Phase

For the roll out, first the **Organisational Modeller** has to define the organisational structure and peculiarities as necessary input for the adaptation of the program. This is a necessary precondition for a smooth and frictionless roll out.

The **Business Specialists** have to take care of the data-migration on the one hand as well as the dynamic migration. That means the running processes have to be transferred (in real time) to the new procedures and processes.

All the definition work, planning and testing in the beginning was just to assure that the roll out can be worked out successfully. Nevertheless it appears that some errors occur in this phase, possibly leading to a fallback into the implementation phase, or even into the requirement definition phase.

Akogrimo Relevant Description:

In Akogrimo Roadmap there are no special roll out roles defined and due to the nature of the project, this phase is not focused upon.

Akogrimo Roadmap Support:

The Akogrimo Roadmap supports this phase by providing:

- a process map that identifies the VO-Processes of Akogrimo that describe how a service will be registered in the Akogrimo platform.

The Akogrimo Roadmap users of this phase are supported by a graphical representation of the VO-Process and a process fact sheet.

2.3 Usage Concept

This section introduces the usage concept that describes the key components and the usability of the Akogrimo Roadmap. Generally the roadmap supports the user to navigate through the content of Akogrimo and provides a guideline in the form of a graphical model to develop applications or services.

Figure 4 describes the usage concept of the Akogrimo Roadmap in more detail and introduces a more formalised description of Figure 3.

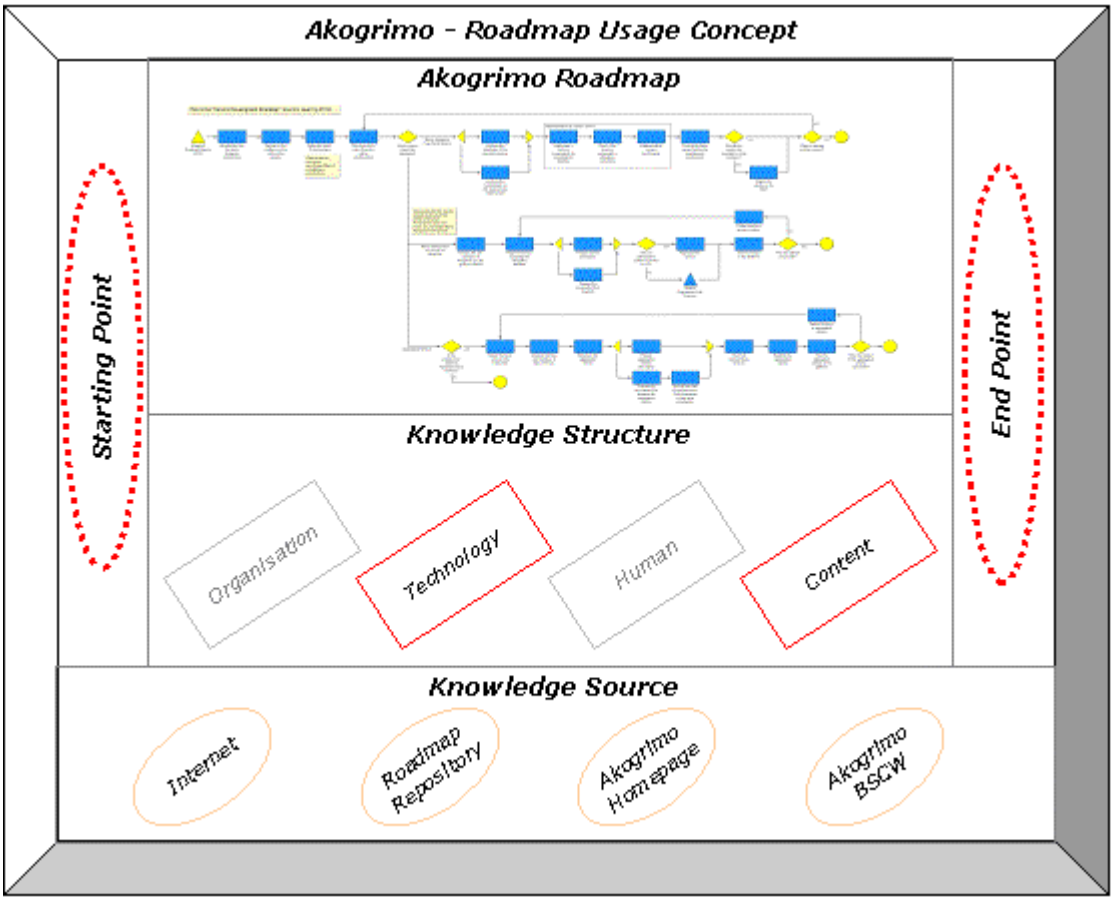


Figure 4 Usage Concept of the Akogrimo Roadmap

Figure 4 describes the usage concept for the Akogrimo Roadmap, this includes the identification of the key components of such a roadmap and the definition of the user, the structure and the sources.

2.3.1 Users of the Roadmap

First the users of the roadmap have to be defined that are supposed to work with the Akogrimo Roadmap. In this context the users are seen as the "market" of the roadmap.

When identifying the users, it has to be considered that there is a distinction between user roles and user groups.

The user roles define the way the user is interacting with the Akogrimo Roadmap based on the supported application scenarios. The Graphical Evolution Tool distinguishes between:

- the Application / Platform Provider,
- the Application / Platform Configurator,
- the Service Developer and
- the Service Development Consultant.

Section 2.2 introduces the above user roles and points out the support of the Akogrimo Roadmap.

In contrast to the user roles, that define the interaction of the user with the Akogrimo Roadmap the user groups define the audience of the Akogrimo Roadmap. The user groups will be enlarged to make the Akogrimo Roadmap public.

In the first step the user will consist of workpackage 5.4 partners as they have been introduced into the Akogrimo Roadmap and are used for the notation. The access consists of mainly read access and partly write access, if required. After a successful test phase, the user group will be enlarged to stepwise involve other partners of the consortium to use the Akogrimo Roadmap.

It is planned to enable the usage of the Akogrimo Roadmap for cooperative EU-projects, interested organisations and related communities. In this phase the participating Universities may provide access to their student courses for a Master and PhD thesis, as well as for teaching issues.

A public provision is under consideration and depends on the feedback of the previous stages.

2.3.2 Starting / End Point of the Akogrimo Roadmap

The starting points indicate different starting positions from where to apply the Akogrimo Roadmap. The individual roadmaps on developing a service for Akogrimo have been analysed to derive the relevant properties that influence the path of the roadmap.

Two properties that influence the development path of the service have been identified:

- The Service Type:

  There is a difference in the development of a service depending on the service type. The different service types have been identified that influence the service development process.

- Skill Profile:

  There are some differences on the skill profile of the service developers. There are different introductions and some limitations depending on the skill profile of the service developer.

Other properties like the used technology or the workpackage layer did not influence the paths of the service development.

The end point of the service development is the deployed Akogrimo service in the platform. The Roadmap identifies the VO-Processes that introduce how to register the service in the Akogrimo environment.

2.3.3 Knowledge Structure of the Roadmap

The knowledge structure of the Akogrimo Roadmap is analysed according to the four different dimensions[3] of knowledge.

Organisation:

The organisational dimension structure items like the budget, project management methods or roles. In the context of Akogrimo this dimension has not been found as relevant.

Technology:

The technology dimension structures items like the key technology, IT-Infrastructure and systems. One technological structure model has been generated that describes the operating system, the used middleware, the used technology, databases and standards.

Human:

The human dimension structures skills, the network of humans, motivation or acceptance. In the context of Akogrimo this dimension has not been found as relevant.

Content:

The content dimension structures the content of Akogrimo. There are three structure models that are used for the Akogrimo Roadmap.

The Akogrimo Feature Map compares the features of traditional approaches versus the innovative Akogrimo features.

The Grid Infrastructure Service Layer and the Application Support Service Layer structures terms that are relevant to develop services in those layers.

2.3.4 Knowledge Sources

The knowledge sources identifies the knowledge sources that can be accessed from the Akogrimo Roadmap. The following list identifies the used knowledge sources for the Akogrimo Roadmap.

- The Internet is used by referencing Web-Pages for the Akogrimo Roadmap. This knowledge source is dynamic as the author of the Roadmap who references the Web-Page usually has no control over the Web-Page. The content can change or be removed without the Roadmap author's knowledge.
- The Akogrimo Roadmap Repository [12], is a knowledge source that collects all documents that are required by the roadmap but are not appropriate to be stored in any of the other containers. In contrary to the Internet this knowledge source is controlled by the Author of the roadmap. In case a document is uploaded the Author is aware of changes or if the document is deleted.
- The Akogrimo Homepage [13] is a knowledge source that is initially constructed to distribute the Akogrimo content to a public audience. The available documents can be used for the Akogrimo Roadmap.
- The Akogrimo BSCW [14] is a knowledge source that is used to share documents with project partners. The documents are restricted and the security mechanism of BSCW has been used for the Akogrimo Roadmap. Documents in the BSCW are referenced from the Akogrimo Roadmap.

[3] The basis structure has been taken from the widely accepted approach in knowledge management considering three dimensions: human, organisation and technology. Only a few authors also mention content as the fourth dimension, this approach had been used in this case, as content became a significant important dimension.

3 Analysis for Enabling Applications to Use Akogrimo

This section is concerned with the identification, description and justification of topics that are relevant for application developers when implementing services or applications on top of the Akogrimo platform. The goal of the analysis for enabling applications to use Akogrimo is to identify all relevant topics that are required to develop applications on top of Akogrimo which entails the problem related to the co-existence and co-operation of different software systems developed on different technologies.

First the relevant topics have been collected based on the individual service development processes. On some steps of the process, there was the need to introduce a set of topics that support the development process. These sets of topics were then identified and grouped in so-called knowledge structures. These knowledge structures were the initial input for the analysis for enabling application to use Akogrimo.

Second, the knowledge structures have been discussed and irrelevant topics were joined or removed. Consistency checks have been performed to map the remaining elements of the knowledge structure with definitions and headlines of deliverables, internal documents or presentations.

Third, the finished knowledge structure has been defined using the simple tables that aims to briefly describe the topic, to justify its necessity for an application developer and to list document references to so-called knowledge sources that provide more detailed information on the topic.

This chapter therefore identifies the necessary topics for an application developer and enables the basis for modelling the knowledge structure models of the Akogrimo Roadmap in Chapter 5.

The following knowledge structures have been identified and are described in more detail:

1. **Starting Point:** First, the definition of the starting point that has been identified by analysing the paths of the development processes will be described. Two variables that influence the development path have been specified:
    a. **Level of Skills** enables an overview of identified skill profiles.
    b. **Service Types** describe the different service types of Akogrimo
2. **Traditional Approaches vs. Akogrimo:** Second, a feature map will be explained that points out relevant features of traditional platforms and Akogrimo specific innovative features. These features are mentioned to indicate innovative topics of the platform.
3. **Application Enabling Concepts:** Third, the key Akogrimo concepts are indicated with references to detailed information to enable the proper usage of Akogrimo key concepts of the application. Two knowledge structures have been found necessary:
    a. **The Application Support Service Layer** that introduces the application relevant Akogrimo concepts.
    b. **The Grid Infrastructure Service Layer** that introduces key Akogrimo concepts that may be required by the application.
4. **Technological Infrastructure:** Fourth, the technological infrastructure that is necessary to develop an application on top of Akogrimo will be introduced.

In the following these dimensions will be introduced in more detail. It is relevant to mention that this chapter focuses on Akogrimo issues and it does not deal with the general approaches to make an existing application Grid-aware, paralleling strategies, etc. In other words, we are not

analysing the strategies to design, develop or migrate distributed applications in a Grid environment.

3.1 The Starting Point

An important aspect when starting the usage or the integration with Akogrimo from an external application is to identify the initial position from which a developer starts. Although there are many aspects that characterise the starting point like technical infrastructure, organisation or architecture related issues, the most important that requires different paths of service development are the definition of the type of service to be developed and the level of experience of the developer.

A different description of the characteristics of the starting point is listed in the individual roadmaps as it seems that they have only little affect on the development procedure.

Based on the analysis of a starting point of Akogrimo partners when tackling the development of a middleware subsystem or an application, two dimensions that strongly affect the development path have been identified:

1. **Level of Skill:** The level of skill distinguishes between beginners that are inexperienced in developing Grid- or Web-services and experienced developers. Beside this distinction there is also the distinction if the developer is part of the Akogrimo community or if the developer is from an external community that wants to use Akogrimo.
2. **Type of Services:** The type of services distinguishable between different internal structures of services that are basic services, complex / aggregated services or workflow services and the roles of the services that are either infrastructure services or customer services.

3.1.1 Identified Skill Profiles

First of all, it is relevant to mention the distinction between the developer skills when we refer to a developer that actively contributes to the implementation of Akogrimo middleware or if it is a developer interested in developing or adapting an existing application on top of this mobile grid infrastructure. In the first case, the middleware developer must know internal architecture of Akogrimo subsystems whereas just a certain level of familiarity is required in the second case (application developer). This distinction must be taken into account in the analysis of required skills.

The implementation of Akogrimo middleware is mainly based on two languages: Java and C# (.NET), although there are also some parts developed in C at the network layer. The chosen reference operating systems are Linux (Obuntu 5 distribution) and Windows (Server 2003). This means that any developer needs to have general knowledge on some of these OSs and the respective developing environments. The baseline IDEs (integrated development environments) in Akogrimo are MS Visual Studio 2003 and Eclipse.

Thus, we identify different levels of skill:

- *Starter or Junior* developer: At this stage, only basic knowledge of developing environments and programming language is requested. In addition, this sort of developer must have some experience in XML notions as well as a basic understanding of WS technologies (mainly the description language WSDL).
- *Advanced or Senior* developer: In addition to previous skills, the senior developer should have experience on implementations using general WS specifications. In this concrete case of being in charge of Grid middleware development, the knowledge of GTv4 or

WSRF.NET frameworks (in general the WSRF specifications) is more than recommended, as well as the familiarity with "Akogrimo architecture".

Senior developers may have special skills related to technological areas on which he/she is focused like:

- Concrete WS specifications: WS-Security, WS-notification, etc
- Service management aspects; state handling, life-cycle control, etc
- Workflow languages and engines (i.e. BPEL4WS).
- Databases (SQL-Server, XIndice, etc)

The roadmap scenario in Akogrimo therefore identified two skill levels, that of the Starter or Junior Programmer and that of the Advanced or Senior Programmer. Although there may be many variations in skill levels, it was necessary to focus on certain skill groups.

Three skill profiles per skill level have been identified:

1. Skill Profile about the *feature* of traditional Grid middlewares and the Akogrimo middleware.
2. Skill Profile about the required *technologies* (XML, SQL, WS specifications, etc)
3. Skill Profile about the key *content of the Grid middleware layers of Akogrimo* covered in WP4.3 and WP4.4, Grid infrastructure serviced layer and Application support layer respectively.

The intersection of skill profiles and levels results in six Skill Profiles that are briefly mentioned in the following table:

	Skill Profile	Starters / Junior Programmers		Advanced / Senior Programmers	
		Application Developer	**Middleware Developer**	**Application Developer**	**Middleware Developer**
Features (refer to Section 3.2 for further details in considered features)	This skill profile contains basic feature indicators of Akogrimo and similar platforms.	Basic skills are required in: 1. Basics features 2. Advanced features, 3. Specific to the part of Akogrimo features on which they are working	Experts skill are required in: 1. Basic features, Advanced skills are required for: 2. Advanced features, 3. Specific to the part of Akogrimo features on which they are working	Advanced skills are required in: 1. Basics features 2. Advanced features, 3. SLA-contract, Security Policy, Business Process of application, VO-Management	Expert skills are required in: 1. Basics features 2. Advanced features, 3. Specific to the part of Akogrimo specific feature on which they are working.
Application Enabling Concepts (see section 3.3)	This skill profile contains the skill of the relevant Akogrimo issues.	Not expected	Advanced	From Basic to Expert, depending on the topic	Expert
Technology (refer to Section 3.4 for further details in involved technology)	This skill profile contains skills on the required technology (XML, WSRF, Operating systems, etc)	Basically there are two technologies that are considered: 1. Java World, that has impact on the programming language (java), the development environment (eclipse), the used middleware (GT4) and the operating system (Linux, Windows). 2. .NET World, that has impact on the programming language (C#), the development environment (Visual Studios) , the used middleware (WSRF.NET) and the operating system (Windows Server 2003). There are common technologies that must be considered in both cases such as the XML, WS specification, etc. It is also necessary to pay attention on more generic WS stacks than exclusively Grid stacks, especially in view of the apparent uptake of WS-* standards by the Grid community, for example the Apache foundation provides offers Java implementations of WS-Security stack and includes implementations of WSRF/WSDM.			

Table 2 Overview of Akogrimo Skill Profiles

3.1.2 Service Types

In a service oriented architecture (SOA) based on Web Services like Akogrimo, the service concept is bringing together the layers of Grid infrastructure middleware services, generic application support services and even lower network services that allow the coordinated and cooperative use of distributed resources (e.g. computation, data sets, applications, mobile devices as PDA, mobile phone, etc.).

However, it is necessary to mention that this document is focussed on the services of the upper layers: Grid infrastructure and Application support layer. The lower network services are not enclosed in the Roadmap document due to two reasons:

i) According to described target audiences, the Akogrimo convergence roadmap aims to capture the necessary processes to develop an internal subsystem (first cycle of the project) and the processes to develop or adapt an application on top of the Akogrimo framework (second cycle of the project) with the testbed development experience.

ii) In fact, the Akogrimo roadmap analyses the aspects for enabling new or existing applications to make use of Akogrimo (in other words to make an application "Akogrimo-aware" so that it can take advantage of benefits and innovation provided by this mobile grid platform). Therefore, it is expected that the Akogrimo platform smoothly integrates network services hiding its complexity to applications that will only interact with some core grid services and the generic application support services.

A set of definitions of different service types has been identified during the modelling exercise. The classification follows two criteria:

- the internal structure of the service itself
- the position or role that service plays in the Akogrimo framework

According to internal structure of the service itself, we have:

Name	***Simple / Basic Service***
Description	We refer to a simple or basic service as a standalone service whose functionalities are implemented in the internal code of the service. This service often offers the functionality via a WSDL interface so that others can use it. In the context of Akogrimo, it is relevant to mention the difference between standard WS (stateless) and the realisation of WS-Resource in a grid environment in which a service has its state and life-cycle to be controlled.
Knowledge Source	www.w3.org/TR/wsdl.html

Name	***Aggregated / Complex Service***
Description	This kind of service implements part of the functionalities in its code and part of its functionality is using other services. It must be taken into account that these "other services" can be simple or, in turn other aggregated services. In another words, we can say that services defining their interfaces and policy can be linked together into a larger composed Web service whose detailed composition need not be exposed to other services invoking the aggregate service. Behind an aggregated service there is "composed" functionality.
Knowledge Source	www.nesc.ac.uk/technical_papers/UKeS-2004-05.pdf

Name	*Workflow Service*
Description	A Workflow service is an aggregated service, which has an associated Workflow template and therefore it needs an engine to orchestrate its functionality. A Workflow is the general way of linking services together and there are likely to be several standards. BPEL is often used as XML-based language for the formal specification of business processes and business interaction protocols.
Knowledge Source	www-128.ibm.com/developerworks/library/specification/ws-bpel/

Name	*Mobile Grid Service*
Description	Mobile Grid Service refers to a service that considers mobility, context and session aspects in its functionally. The concept of "mobile" deals with the "nomadic" concept and not only supports mobile resources (hosts, mobile devices and even users) connected by standard networks and wireless links. This kind of service can be simple or aggregated but they are normally part of a workflow service that enables a distributed application in a mobile grid infrastructure such as Akogrimo.
Knowledge Source	Introduction to mobile grid concepts: http://www.akogrimo.org/modules.php?name=UpDownload&req=-getit&lid=28

Next figure extracted from D 3.1.1 (Overall Architecture Definition and Layer Integration) [15] introduces Akogrimo building blocks of the different layers and it is used to classify service types regarding its functionality in the Akogrimo framework.

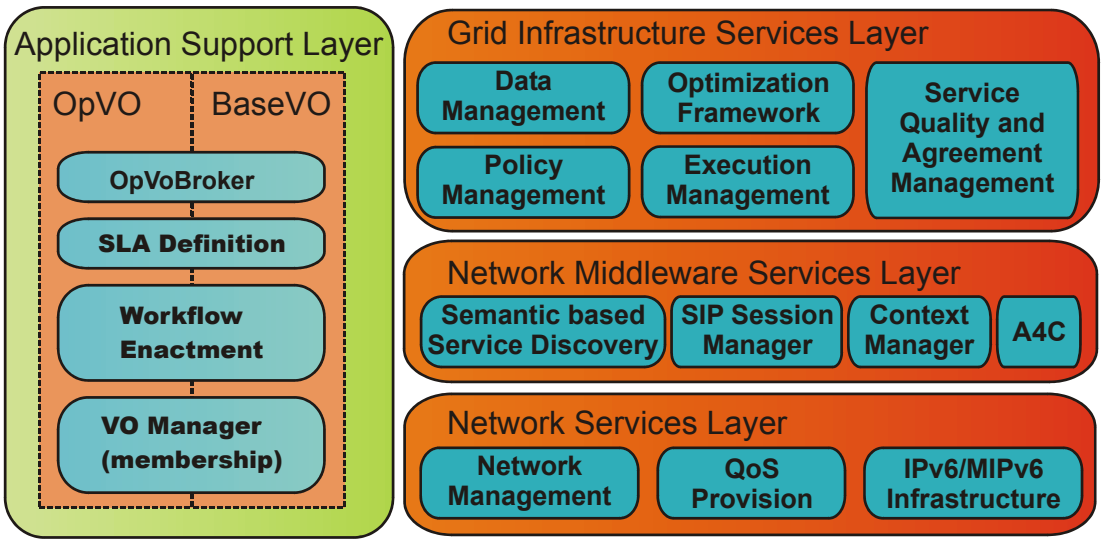


Figure 5 Akogrimo Basic Components

According to the function that a service plays, we identify;

Name	***Akogrimo Core Network Service***
Description	This service deals with network infrastructure layer and covers aspects such as the mobility management in IPv6 (Mobile IPv6 protocol), network security mechanisms, network access, mobile terminal, network signalling, etc The closer relationship with Grid infrastructure subsystem takes place for QoS provisioning since Akogrimo aims to smoothly provide the QoS monitoring of network and grid level. However, the analysis in the roadmap modelling is mainly based on the Grid middleware and application support layer perspective.
Knowledge Source	This is mainly related to WP4.1 of Akogrimo platform.

Name	***Akogrimo Network Middleware Service***
Description	This kind of service encompasses important functionalities on top of core network services, such as the semantic discovery, the session control, the context management and the AAA (Authentication, Authorisation, Auditing) service extended with Accounting and Charging.
Knowledge Source	This is covered in WP4.2 of Akogrimo platform.

Name	***Akogrimo Grid Infrastructure / Grid Middleware Service***
Description	This service deals with Grid core services of Akogrimo infrastructure and embraces functionalities of a middleware based on service oriented architecture (typically OGSA services) such as Grid security services, execution management and scheduler service, policy management, QoS monitoring services according to agreed SLA, etc.
Referenced Support	WP4.3 focuses in this type of services

Name	***Akogrimo Application Support Service***
Description	These services deal with all aspects related to the management of the Virtual Organisation, definition of user profiles, membership management as well as the negotiation of Service Level Agreements and the execution control of business process. This is the upper interface of Akogrimo with the application service, for example the service agent services to interact with the adaptive workflow service, VO management, accounting mechanisms, etc.
Referenced Support	These services are covered in WP4.4 that leverage in functionalities of WP4.2 and WP4.3

Name	***Customer / Application service***
Description	This service does not appear explicitly in the previous figure, encompasses the distributed application services that run on top of Akogrimo platform. This kind of service makes use of exposed platform functionalities (normally the application support services) in its operation. Examples of these services are the EGC Analyser or the visualization of EGC data in the eHealth testbed.
Referenced Support	This service refers to testbeds that will be developed in WP5.2 (2nd cycle of the project).

3.2 Traditional Approaches vs. Akogrimo

One of the key questions during the analysis of approaches for enabling Application to use Akogrimo was the identification of the gap between traditional approaches and the innovative Akogrimo architecture.

The roadmap is seen as an assistance system that guides the user from traditional approaches to the Akogrimo architecture and provides support in implementing services and applications within Akogrimo. The following list shows the identified key elements from traditional approaches and the Akogrimo approach in three categories.

1. **Basic features:** These are relevant features that are common in traditional approaches and also common in Akogrimo. The key elements are listed and briefly described.
2. **Advanced features:** These relevant and advanced features are provided in traditional approaches, but need consideration when using Akogrimo. This category covers all elements that are more sophisticated than the basic features but are not Akogrimo unique. However there are topics that need special attention.
3. **Akogrimo special issues:** These features are seen as innovative parts of Akogrimo. Therefore special consideration is required when implementing a service for Akogrimo. In this list only the technical innovations that need to be considered from an applications development point of view are identified. The Akogrimo innovations are heavily discussed in D 6.3.1 (Marked Study) [16] and was merged with the technical Akogrimo special issues in this list. All technical relevant issues for the application layer 4.4 and 4.3 have therefore been merged in this section.

The gap between the traditional platforms and Akogrimo lies in the third set of features. As many other middlewares, Akogrimo follows the trend of service oriented architectures, relies in open WS standards, addresses globally security aspects, adaptive workflows, etc. The main innovation consists of the smooth integration of technologies related to Next Generation of Grids (NGG) and mobile internet based on IPv6. This aspect leads to the Akogrimo special issues such as the fact of considering the vertical context awareness derived from the mobility of resources and actors, the virtualisation of network resources, etc. to form the Mobile Dynamic Virtual Organisation paradigm as an extension of the VO concept in which the mobility is a must.

In the following the identified topics for bridging the gap between traditional approaches and the Akogrimo platform are described in the above mentioned categories.

3.2.1 Basic Features

Name	***Service Management : State and Life Cycle Control***
Description	Nowadays we have reached the convergence of two worlds, the Grid computing and the Web in the so-called Next Grid Generation (NGG). The key concept is the virtualisation of heterogeneous resources that are exposed as services. This convergence has been gathered in a set of specifications known as WSRF (Web Service Resource Framework) standardised by OASIS in which the services are provided with "persistent state" (derived from the classical concept of "factory" in Grid) and with mechanisms to manage the life cycle of these stated services through properties associated with the Web service. These properties are supported by the underlying Grid framework (in our case Globus Toolkit v4 or WSRF.NET) and some others are managed by the developer/user. The implementation of services in Akogrimo foresees the realisation of WS-Resource. Successful realisation of the Grid vision of a broadly applicable and adopted framework for distributed system integration, virtualisation, and management requires the support for configuring Web services, their deployment, and maintaining their lifecycle management which is gathered in a CDDLM foundation document of Global Grid Forum (GGF).
Knowledge Source	www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf www.globus.org/wsrf/ www.gridforum.org/documents/GFD.50.pdf http://www.cs.virginia.edu/~gsw2c/WSRFdotNet/WSRF.NET_-Developer_Tutorial.pdf)

Name	***Network Communication***
Description	The network communication protocol that will be used in Akogrimo is IPv6. It is determined that the communication will be based in three distinct network service bundles, that will be provided for the communication between the participating entities. These network services will be available to the users and other Akogrimo services through a WS-interface designed for this purpose. By these means we are going to have available network services tailored to the needs for each application on a QoS base, ease accessible as regular Web services.

3.2.2 Advanced Features

Name	***Security Aspects***
Description	In Akogrimo the security features are very important and represent a vertical issue between all layers. The target is to foresee the building of a secure environment, merging secure standards from the upper to the lower layers.
Knowledge Source	D 4.4.1 (http://bscw.hlrs.de/bscw/bscw.cgi/0/75067), WP4.4 internal documentation (http://bscw.hlrs.de/bscw/bscw.cgi/0/36139), D 3.1.1 Chapter 6 (http://bscw.hlrs.de/bscw/bscw.cgi/d64103/D3.1.1_Over-all_Architecture_v1.0_final.pdf)

Name	***Notification Mechanism***
Description	Notification is the mean based on "event-driven" interaction pattern for communication by which "consumers" (as receiver of information messages) can register interest in receiving asynchronous messages relating to desired topics. This notification can occur either via direct communication from a Notification Producer, or through a Notification Broker which acts as an intermediary. In Akogrimo, this is the communication mechanism considered in several situations among subsystems since it gives independency from implementation technology (.NET and Java) and allows decoupling the subsystems.
Knowledge Source	WSN oasis committee is in charge of these specifications: WS-BaseNotification and BrokeredNotification: Latest specifications are available at: http://www.oasis-open.org/committees/wsn

Name	***Adaptive Workflow Enactment***
Description	New opportunities for business enterprises have been created with process management systems to control the interactions and processes of companies. In Akogrimo, workflow management components exactly enact business processes as a composition of services described in a process description language (we use BPEL4WS as a specification language). An advanced feature of Akogrimo infrastructure is its capacity of managing business processes to adapt to special circumstances such as context changes or violations in the QoS provision. The infrastructure is able to dynamically orchestrate a set of services and adapt its operation to changes.
Knowledge Source	D4.4.1 - Architecture of the Application Support Services Layer V1 http://bscw.hlrs.de/bscw/bscw.cgi/0/75067

Name	***Interoperability***
Description	In general, the interoperability is defined as the ability of a system or a product to work with other systems or products without special integration efforts. It is the degree to which a service is connected to and operates with others and obviously here the use of open standards and WS specifications is a must. In Akogrimo, this is an important feature since it specifies the degree to which a middleware service or component is able to be successfully interoperated or integrated with other specified applications or components (e.g., browsers, legacy databases, legacy applications, external systems, and required COTS

	components). Akogrimo considers major WS specifications in its implementation such as WS-Agreement, WS-Notification, WS-Trust, WS-Security, etc
Knowledge Source	www.opfro.org/index.html?Components/WorkProducts/ModelSet/Quality-Model/QualityFactors/Interoperability.html

Name	***Scalability***
Description	Scalability is a qualitative feature about how easy an infrastructure, platform or system is able to be modified to expand its existing capabilities to a higher demanding situations (e.g., to handle more simultaneous users, to control a large number of process interactions, or to store more information in its databases). This is a very important aspect in a distributed environment as the one intended for Akogrimo (mobile and grid). In Akogrimo, the scalability feature is being considered at an architectural design level for the moment and scaling tests shall be defined to determine the degree of linearity of middleware implementation with respect to a system request change with respect to a modification in the components capacities inside the system.
Knowledge Source	http://www.opfro.org/index.html?Components/WorkProducts/ModelSet/-QualityModel/QualityFactors/Scalability.html

Name	***Business Model Support***
Description	One of the advanced features of the Akogrimo project is the Business Model support. The whole Akogrimo architecture is structured so as to allow a Grid support for the provision of commercially exploited services. Various business models can be applied on the proposed architecture making use of the services that are available from the network to the application layers, through middleware. Workflows, orchestration of services, management of users and their identities, policies as well as adequate security schemes are incorporated to enable the adoption of business models for candidate organisations that would like to invest in the Akogrimo concept.
Knowledge Source	D4.4.1 – Architecture of the Application Support Services Layer V1 http://bscw.hlrs.de/bscw/bscw.cgi/d72750/

Name	***Performance / Fault Tolerance***
Description	*Performance*: Akogrimo is a complex platform, addressing large scale problems and service provision across distributed resources and large number of users. For this reason the performance issues are of vital importance for the viability (at least from a business perspective) of the infrastructure. The Grid middleware layer is there to advance the issue of performance and of manageability of these environments, in a manner that will automate various procedures and accelerate the performance based on the knowledge for the execution planning of various jobs. *Fault tolerance*: Fault tolerance is also related to the performance of the platform and the proper operation of the involved resources. Complex and heterogeneous systems (and especially in wireless and ad hoc infrastructures) are prone to failures. Fault tolerance will enable the enterprises that adopt the Akogrimo

	architecture to ensure a minimum set of guaranteed operation, something that will make their business scenarios feasible.
Knowledge Source	Fault–tolerant Grid Services Using Primary–Backup: Feasibility and Performance. http://www.cs.ucsd.edu/~dzagorod/research/pubs/zhang_et_al-ft_grid_services_pb-cluster04.pdf

Name	***Service-Oriented Architecture across Akogrimo Layers***
Description	A Service-Oriented architecture (SOA) is essentially a collection of decoupled services communicating with each other. A service is a function that is well-defined, self-contained, and does not depend on the context or state of other services. The technology of Web services is the most likely connection technology of service-oriented architectures. In SOAs a service consumer is sending a service request message to a service provider. The service provider returns a response message to the service consumer. This approach is adopted in all Akogrimo layers. All modules in Akogrimo (from application services to middleware and network services) are exposing their functionality as Web services. Reference architecture for Grid is the OGSA. Akogrimo is considering the requirements and description of this architecture and it is extending some aspects not gathered in OGSA such as the context awareness and the integration with network layer (see next section – Akogrimo special features).
Knowledge Source	http://www.ggf.org/documents/GFD.30.pdf

3.2.3 Akogrimo Special Features

Name	***Vertical Context Awareness / Pervasiveness and Mobility***
Description	This context awareness is probably one of the main innovations of Akogrimo platform. For the first time, a Grid infrastructure becomes a real mobile Grid that is able to gather the end-user context and adapt to context changes (we should talk about *context of mobile services* which most of them represents the end-user or are part of composed application). The term *Context* encompasses information such as location, availability, device capabilities, schedule of a service. Akogrimo middleware gathers context information from different sources across the layers. The information is detected at the lowest layer in the Akogrimo mobile grid in a raw form but may be required in a partly processed form by an application, hence the reference to "vertical". One of the challenges is to manage the mapping of this information in a useful way. All Akogrimo subsystems can subscribe to context changes of particular types and receive updates of this information in order to react or make necessary adjustments accordingly. The main benefit is the flexible provision of applications since Akogrimo adapts to user context changes (changes of user location, of session due to the use of different devices, of resources availability in mobile Grid, etc) changing data formats, qualities of service, user interfaces, etc.
Knowledge	D4.4.1 - Architecture of the Application Support Services Layer V1

Source	http://bscw.hlrs.de/bscw/bscw.cgi/0/75067 D4.2.2 - Final Integrated Services Design and Implementation Report http://bscw.hlrs.de/bscw/bscw.cgi/0/82972

Name	***Virtualisation of Network Resources***
Description	Akogrimo is encompassing technologies and layers of the Next Generation of Grids and Mobile Internet networks based on IPv6. The key concept of NGG consists in the complete virtualisation of resources. Akogrimo aims to integrate both network and grid layers in the mobile grid infrastructure. Thus, Akogrimo platform virtualises network resources (for example certain bandwidth, assure minimum delay or jitter) as "services" such as the Grid classical resources like storage or computational power, hiding network complexity to applications that will just need to interface with generic support services and grid core services.
Knowledge Source	D4.1.2 – Consolidated Network Service Provisioning Concept http://bscw.hlrs.de/bscw/bscw.cgi/0/86362

Name	***QoS from Network to Application***
Description	An important added value of Akogrimo infrastructure is the tight relationship that establishes between the grid and the network layer in such a way that Grid middleware becomes aware of network capabilities. A clear example of this network and grid functionalities is shown in the management of the Quality of Service. The SLA contract and its negotiation considers QoS parameters that belong to both grid resources (CPU use, Memory disk, etc) and network capabilities (bandwidth, priorities for packet traffic, etc) by means of network bundles or profiles. Thus, the QoS requests of applications are mapped into these QoS parameters. This novelty is completed with the network – grid interactions. Thus, any changes on network performance are taking into account in monitoring of QoS parameters and corrective actions and penalties can be applied according to the defined policy in a per-case basis.
Knowledge Source	http://bscw.hlrs.de/bscw/bscw.cgi/0/66444 D4.3.1 Architecture of the Infrastructure Services Layer V1

Name	***Vertical AAA (A4C) Through Layers***
Description	The A4C system provides the necessary functionality for authenticating users, authorising access to services, accounting and charging for service usage, as well as auditing within Akogrimo. It is based on the generic AAA (Authentication, Authorisation and Accounting) architecture proposed by the Internet Engineering Task Force (IETF). Important Akogrimo components that access A4C include Base VO manager, SIP server, context manager, access router, and QoS broker. By means of the integration of network and grid infrastructure, Akogrimo offers a flexible accounting mechanism that integrates the consume of network resources (Bandwidth), grid resources (CPU/disk usage) allowing at the same time the customer specific definition of accounting measurements.
Knowledge Source	D4.2.2 - Final Integrated Services Design and Implementation Report http://bscw.hlrs.de/bscw/bscw.cgi/0/82972

Name	***Mobility of resources and MDVO paradigm***
Description	Akogrimo is a SOA web-based framework that virtualises as services classical Grid resources, network resources and mobile resources. Akogrimo allows the coordination of these recourses (*hosts, mobile devices and even users*) to enable a Mobile Dynamic Virtual Organisation (MDVO) that is a natural extension of the VO concept with the participation of mobile resources bringing together concepts of mobile computing and cooperative environments for creating a mobile Grid. The MDVO concept will incorporate network-identities based concepts of personalisation, profiling and privacy. By *mobile resources* we refer to "nomadic" resources that can or cannot make use of wireless networks but they are characterised by having special constraints in terms of localisation, availability, etc.
Knowledge Source	D4.4.1 - Architecture of the Application Support Services Layer V1 http://bscw.hlrs.de/bscw/bscw.cgi/0/75067

3.3 Application Enabling Concepts

This section deals with major elements that have to be considered when an application would like to use Akogrimo as a platform. The following discussion is closely related to the architecture of the Grid Infrastructure Service Layer and the Application Support Service Layer but has its focus on the navigation of an application developer through the architectural elements of Akogrimo. The following discussion is therefore seen as an introduction of key terms and as a pointer where to get detailed information.

3.3.1 Application Support Service Layer

The section describes the key elements of the Akogrimo architecture to support an application developer that wants to use Akogrimo as a platform. The goal is to introduce the key elements and points to supply the developer with detailed information.

Name	***Orchestration***
Description	One of the main challenges in an Akogrimo project is to build new services composing available services (components) from different service providers. The challenge of this approach (the aggregator of the composed service can sell this "complex" service to his customers) is not in composition itself but in the dynamic environment that Akogrimo infrastructure is going to manage (component services can not have a fixed geographic location, user of the composed services can be mobile and/or nomadic). In order to achieve this goal Akogrimo leverages on the business process concept. A business process represents a set of one or more linked procedures or activities that collectively realise a business objective goal, normally within the context of an organisational structure defining functional roles and relationships. The business process is available at the application layer and the final consumer wants to buy and use it. Behind the business process there is a workflow that represents the automation of the business process. The workflow coordinates and manages component services or entities involved in the automation of business processes. The view of business processes is a high level vision (application layer), but at low level layers we need to orchestrate one or more services. In this vision we need to publish all the available services inside Grid Service Discovery Server, using templates that identify their features. As service features we mean at least QoS and providers that are able to provide them. In Akogrimo, we follow a service oriented approach and we have the goal of virtualising as a service what is sold in the MDVO. Furthermore, when we talk about service refers to Web Service or WS-Resource. This means that in Akogrimo we have orchestration of Web Services or WS-Resources
Knowledge Source	D4.4.1 (http://bscw.hlrs.de/bscw/bscw.cgi/0/75067) WP4.4 internal documentation: http://bscw.hlrs.de/bscw/bscw.cgi/0/36139 Orchestration: http://devresource.hp.com/drc/technical_white_papers/WSOrch/-WSOrchestration.pdf

Name	***VO Management***
Description	In the Akogrimo project the concept of the Mobile Dynamic Virtual Organisation (MDVO) has been introduced. This concept is based on the classic definition of Virtual Organisation that in Akogrimo project is characterised by two main properties: o Mobility o Dynamicity The MDVO is the environment where customers and providers meet together in order to make a business. The complexity of such environment is due to the dynamic changes of the participants and services that are available. Actually, the actors involved in MDVO can dynamically change and different actors can have join together in order to achieve the MDVO goals. Of course, the participants can span different administrative domains and their geographic mobility contributes to make such environments difficult to manage. VO management should provide typical capabilities of a business VO (subscribing new participants, registering those participants enabling them to sign-in their presence in the VO, supporting the publishing phase of services, checking the access rights for invoking services,…) but everything needs to be adapted to take in account dynamicity and mobility.
Knowledge Source	VO in OGSA (read introduction section): https://forge.gridforum.org/docman2/-ViewCategory.php?group_id=42&category_id=357 MDVO (read DoW): http://bscw.hlrs.de/bscw/bscw.cgi/d35403/-Akogrimo_DoW_v45_EC%20(final).pdf D4.4.1 (http://bscw.hlrs.de/bscw/bscw.cgi/0/75067 WP4.4 internal documentation (http://bscw.hlrs.de/bscw/bscw.cgi/0/36139)

Name	***SLA – Management (SLA High Level)***
Description	Akogrimo infrastructure is going to support a business oriented environment. Then a basilar topic to be considered is the contract agreed between a customer and a provider. The object of the contract will be the sold service and, of course, it will be defined as a Service Level to be respected during the provision of that service according with the price paid for using it. In order to make these capabilities effective the Akogrimo infrastructure includes a SLA (Service Level Agreement) management subsystem. The SLA management at application layer is in charge of executing the negotiation phase to acquire a service and establish a contract between the customer and the provider. The negotiation phase starts taking into account the list of services published by the service providers and the description associated to them (i.e. the Service Level Agreement). When the negotiation is concluded a contract is established, where QoS, charging policies and violation rules are specified. During the execution of acquired services at lower layer (4.3 level) there are specific services that will monitor the execution and will check if the agreement established in the contract is respected or not.
Knowledge Source	D4.4.1 (http://bscw.hlrs.de/bscw/bscw.cgi/0/75067) WP4.4 internal documentation (http://bscw.hlrs.de/bscw/bscw.cgi/0/36139)

Name	***Security***
Description	The dynamic and cross administrative domains Virtual Organisations require a careful management of security. Even if Akogrimo doesn't have a strong focus on security, in a business oriented environment security issues have to be taken in consideration. Of course, in a MDVO the security mechanisms should span across the different layers of Akogrimo then we have to manage it from network to grid layers. But what are these issues? They are the typical aspects related to the establishment of a secure communication channel: • Authentication • Authorisation • Adopt a method or technique to provide means by which data can be transferred from one place or user to another without risk of interception or tampering In a typical Akogrimo deployment environment, a request passes through several distinct channels and these channels can pass over different networks administrated by different entities. Furthermore the channel itself is not fixed but it can dynamically changed during the application use. Summarising in Akogrimo the security mechanisms are strictly related to the participation in BVO and OpVO, and they are in charge of enabling communication between involved parties by satisfying the security policies between requestors and requested services. In general, the VO (we are addressing BVO and OpVO) are viewed as a 'large' (virtual) administrative domain owning its security services for managing security among VO entities and bridging the VO members' security domains
Knowledge Source	Security in Grid environments: http://www.ggf.org/ggf_areas_security.htm Overview on Akogrimo security issues to be addressed (DoW): http://bscw.hlrs.de/bscw/bscw.cgi/d35403/-Akogrimo_DoW_v45_EC%20(final).pdf Akogrimo preliminary security sketch: D4.4.1: http://bscw.hlrs.de/bscw/bscw.cgi/0/75067 WP4.4 internal documentation: http://bscw.hlrs.de/bscw/bscw.cgi/0/36139

Name	***Application Specific Services***
Description	On top of the Akogrimo infrastructure, in principle, many different applications could be executed. Of course, the Akogrimo infrastructure is assumed to be generic enough to be used in different cases and by different applications. In order to do that, it is necessary to implement some application specific services and some specific interfaces that allow the application itself to communicate and interact with the Akogrimo infrastructure. These services and functionalities support the execution of specific applications and then they are not really part of the infrastructure but they can be considered a sort of bridge to be built for each specific application. Of course, some of these functionalities are generic enough to be used in different cases (e.g. for policy management purpose, for security support, SLA management,…). In particular, in Akogrimo we will focus on specific services related to the eHealth and eLearning applications.

Knowledge Source	D4.4.1 section 3.5: http://bscw.hlrs.de/bscw/bscw.cgi/0/75067 WP4.4 internal documentation : http://bscw.hlrs.de/bscw/bscw.cgi/0/36139

Name	***Legacy Application Integration***
Description	We cannot assume that every application executed on Akogrimo infrastructure will be implemented from scratch. Then we have to face the problem of finding some guidelines to integrate legacy applications in the Akogrimo environment. This is the aim of this key topic that will provide input on how to make existing applications Akogrimo-aware. The main issue related to the migration arises in the adaptation of a large number of presently used legacy applications to SOA. One of the key challenges facing this process is a cost-effective migration strategy that has to be devised. Significant corporate investments in the existing high quality validated software needs to be preserved during the transition to the new environment. Legacy systems are critical enterprise assets reflecting key business practices and knowledge acquired over the life of an organisation. From economic perspective, it is unacceptable to discard fully operative software and re-design and re-implement it from scratch using grid-enabled technology. It is essential to ensure that legacy systems which can be incorporated into the grid infrastructure in a smooth, incremental way which does not involve excessive development effort. On the basis of the above consideration, in Akogrimo we are going to individuate and propose cost effective solutions in order to achieve the integration of legacy applications. The results of our investigation will come from the study of Akogrimo testbed, nevertheless, the aim is to provide solutions that are not specific but generic so can be applied in several cases.
Knowledge Source	D4.4.1 section 3.6: http://bscw.hlrs.de/bscw/bscw.cgi/0/75067 WP4.4 internal documentation : http://bscw.hlrs.de/bscw/bscw.cgi/0/36139

3.3.2 Grid Infrastructure Service Layer

The previous section describes the key elements for an application developer that want to use Akogrimo as a platform. This section also describes the key elements of the Grid Infrastructure Service Layer as it is likely that an application developer has also to be aware of the key concepts in this layer.

The goal is to identify and introduce the relevant elements for the application developer and list documents to get detailed information.

Name	***Policy Management Service***
Description	The Akogrimo project foresees different kinds of communications with different kinds of terminals. In this scenario there are many participants involved in the communication and some rules are necessary in order to manage them. These rules represent policies, fixed on the base of participant roles and participant services offered. The first main distinction between participants could be made on the basis of actions that they want to take within the VO. So, some participants may want to offer a service (SP) and some others may want to use/consume them (SC). The interactions between SC and SP are managed by a specific agent (e.g. EMS). This agent has to have the possibility to retrieve policy information about a particular context (i.e. action requested by a SC on a resource deployed by a SP). Agents can change their own behaviour on the basis of the policies received. There is a need for distributed, automated management agents whose behaviour has also to change dynamically to reflect the evolution of the system being managed. Policies are a means of specifying and influencing management behaviour within a distributed system, without coding the behaviour into the manager agents. Furthermore in a large distributed system there will be multiple human administrators specifying policies which are stored on distributed policy servers. Policy conflicts can arise due to omissions, errors or conflicting requirements of the administrators specifying the policies.
Knowledge Source	D 4.3.1 - Architecture of the Infrastructure Services Layer V1. http://bscw.hlrs.de/bscw/bscw.cgi/0/66444

Name	***Execution Management Service***
Description	Execution Management Services (OGSA-EMS) are concerned with the problems of instantiating and managing tasks. Execution management systems support existing grid middleware systems for information gathering (e.g. MDS, Ganglia, and NWS), job execution (Globus GRAM) and data movement (e.g. GridFTP, RFT). Execution management system should make use of advanced reservation mechanisms. By using advanced reservations, grid execution systems enables users to run their applications in explicitly defined timeframe. This functionality is important for interactive applications, where users needs to get a response in real time. In addition, by using advance reservations, parallel applications distributed over multiple clusters can be guaranteed synchronous at the startup of the processes on all clusters. Support for various job types is implemented. Basic types are parallel and serial jobs, job arrays and workflows. Execution management system takes into account data location and movement (data-aware scheduling) and characteristics of network links. An example of data-aware scheduling are mapping tasks to resources closer to data instead of

	moving large data over network. System scalability is very important due to the nature of a grid system. Standard functionalities of job management systems being provided are checkpointing, pre-emption, job migration, fault-tolerance and rescheduling. The functional capabilities of the developed system are: 1. Selecting the set of resources that can be used to execute a submitted job. 2. Assigning jobs to resources and creating an execution plan, trying to balance the workload, optimise the performance and provide QoS. 3. Handling job queues and priorities to meet SLAs or handle crisis situations. 4. Replicating jobs to provide fault tolerance. 5. Providing advanced resource reservation. 6. Managing the job execution (deploy, start, suspend, terminate).
Knowledge Source	D 4.3.1 – Architecture of the Infrastructure Services Layer V1. http://bscw.hlrs.de/bscw/bscw.cgi/0/66444 The Open Grid Services Architecture, Version 1.0 http://www.gridforum.org/documents/GWD-I-E/GFD-I.030.pdf

Name	***Data Management Service***
Description	The Data Management Service provides efficient access to and movement of huge quantities of data, data sharing, archiving of data and data management are essential requirements in science, technology and business areas. Data services are used to move data to where it is needed, manage replicated copies, run queries and updates, and transform data into new formats. Data services requirements include: • *Data access.* Easy and efficient access to various types of data (such as database, files, and streams), independent of its physical location or platform, by abstracting underlying data sources as required. Mechanisms are also required for controlling access rights at different levels of granularity. • *Data consistency.* Consistency should be maintained when cached or replicated data is modified. • *Data persistency.* Data and its association with its metadata should be maintained for their entire lifetime. It should be possible to use multiple persistency models. • *Data integration.* Mechanisms for integrating heterogeneous, federated and distributed data are required. Many different types of data must be supported (flat files, streams, DBMS, catalogues, derivations from other data, data services as data resources, etc.). It is also required to be able to search data available in various formats in a uniform way. • *Data location management.* The required data should be made available at the requested location. It should be possible to allow for selection in various ways, such as transfer, copying, and caching, according to the nature of data.
Knowledge Source	D 4.3.1 – Architecture of the Infrastructure Services Layer V1. http://bscw.hlrs.de/bscw/bscw.cgi/0/66444 The Open Grid Services Architecture, Version 1.0 http://www.gridforum.org/documents/GWD-I-E/GFD-I.030.pdf

Name	***Monitoring Service***
Description	Monitoring tasks are fundamental in every distributed computational system. Monitoring data represents an operational snapshot of the system behaviour along the time axis. Such information is fundamental to determining the origin of the problems or for tuning different system components. For instance, fault detection and recovery mechanisms need a monitoring component to decide whether a particular subsystem or server should be restarted due to the information collected by the monitoring system. In the particular case of Akogrimo grid computing system, the following functionalities have been identified for the Monitoring component: • To monitor quality of services. • To monitor service level agreements. • To check resources status. • To provide monitoring information used for controlling tasks.
Knowledge Source	D 4.3.1 – Architecture of the Infrastructure Services Layer V1. http://bscw.hlrs.de/bscw/bscw.cgi/0/66444 The Open Grid Services Architecture, Version 1.0 http://www.gridforum.org/documents/GWD-I-E/GFD-I.030.pdf

Name	***SLA Enforcement***
Description	The SLA subsystem handles the aspects related to the quality of service (QoS) of all offered services in Akogrimo. When a Service Costumer (SC) wants to use a service published by a Service Provider (SP), they must negotiate the conditions of use which are sealed in the SLA contract. Thus, during the use of any kind of service, the agreed QoS level must be assured. SLA subsystem is in charge of watching over the fulfilment of this QoS, and to know by means of the Policy Manager what recovering actions should be applied in case that the QoS described in the contract is not respected throughout the execution phase. At this point the figure of a trusted third party appears; part of SLA functionalities can be optionally located on an independent third party trusted by SC and SP, who will look after the fulfilment of the agreed conditions. The following functional capabilities have been identified for SLA Enforcement: • Receiving measurements by the Monitoring Subsystem • Evaluating thresholds for the verification of fulfilment of the QoS detailed in SLA-Contract. • Propagating violations • Taking Recovery actions
Knowledge Source	D 4.3.1 – Architecture of the Infrastructure Services Layer V1. http://bscw.hlrs.de/bscw/bscw.cgi/0/66444

3.4 The Technological Infrastructure

The technological infrastructure is an important dimension to be considered when using Akogrimo for an application to identify and avoid interoperability problems, to speed-up the setup of the infrastructure and to enable the share of sources and services.
In the following ,the relevant key technologies for Akogrimo have been identified, introduced and referenced to more detailed documents that are seen as interesting for application developer sthat want to use Akogrimo as a platform.

Operating Systems:

Name	***Windows Server 2003***
Description	One of the main Akogrimo challenges is to have an architecture that allows the interactions between different platforms. On the basis of this precondition, we are going to assume the adoption of at least two OS families: MS and Linux. In particular, with respect to Microsoft we are going to use Windows Server 2003. This OS has been chosen as we are going to implement management services that can profit from using capabilities allow a better performing web platform. Furthermore, in Akogrimo services to be managed will be WS or WS-Resource and Windows Server 2003 provides full support for hosting this kind of services. In fact, WSRF.NET (a middleware to implement WS-Resource on Microsoft platform) can be installed on Windows Server 2003
Knowledge Source	http://www.microsoft.com/windowsserver2003/evaluation/overview/-default.mspx

Name	***Windows XP***
Description	Similar consideration (as for Windows Server 2003) can be taken for MS Windows XP. In particular, the adoption of Windows XP could be reasonable for user sthat are not interested in running management services that need Windows Server capabilities ,but need a lower spec OS ,for example ,solely to run their application/service. In any case, MS Windows XP has been tested to run a Grid middleware (in particular WSRF.NET) for building WS-Resource sor WS (and this is a mandatory precondition to choose an OS because, also in this case, the provided services need to be WS or WS-Resource. Finally, it is worth mentioning that Windows XP (as with Windows 2003 server) support sWSE (Web Service Enhancement),the MS implementation of many WS-specifications.
Knowledge Source	Windows XP: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwxp/html/rtc_enhancerichclient-real-timecomm.asp WSE: http://msdn.microsoft.com/webservices/webservices/building/wse/-default.aspx

Name	***Ubuntu 5 Linux***
Description	All Akogrimo components developed using Linux are being initially integrated on Ubuntu 5.04. We preferred this distribution since it provides support to IPv6 protocol as well as one of the most user friendly interfaces among the Linux distributions. This does not exclude the possibility of testing our components on different Linux distributions.
Knowledge Source	http://www.ubuntu.org

Middleware:

Name	***WSRF.NET***
Description	In Akogrimo, there is an effort to virtualise the resources available in the MDVO. In an Akogrimo context ,virtualise means that the resources (e.g. software, hardware, instrumentation…) will be available as Web Service and WS-Resource. On the basis of this assumption, it has been evaluated that it would be useful to have a framework that implements the basic WS-Resource specifications so that in Akogrimo we could focus on the challenges that are related to Akogrimo without spending effort to provide solution sto low layer issues. These framework sare available for different platforms and, in particular, for MS platforms we have the framework WSRF.NET: an implementation of WSRF from the University of Virginia based on MS .NET.
Knowledge Source	http://www.cs.virginia.edu/~gsw2c/wsrf.net.html, http://www.cs.virigina.edu/~gsw2cWSRFdotNET/WSRF.NET Developer Tutoruak.pdf, http://www.cs.virginia.edu/~gsw2c/WSRFdotNet/WSRFdotNet programmers reference.pdf

Name	***Globus Toolkit 4***
Description	The open source Globus Toolkit is a fundamental enabling technology for the "Grid," allowing people to share computing power, databases, and other tools securely online across corporate, institutional, and geographic boundaries without sacrificing local autonomy. The toolkit includes software for security, information infrastructure, resource management, data management, communication, fault detection, and portability. It is packaged as a set of components that can be used either independently or together to develop applications. Every organization has unique modes of operation, and collaboration between multiple organizations is hindered by incompatibility of resources such as data archives, computers, and networks. The Globus Toolkit was conceived to remove obstacles that prevent seamless collaboration. Its core services, interfaces and protocols allow users to access remote resources as if they were located within their own machine room while simultaneously preserving local control over who can use resources and when. The Globus Toolkit has grown through an open-source strategy similar to the Linux operating system's, and distinct from proprietary attempts at resource-sharing software. This encourages broader, more rapid adoption and leads to greater technical innovation, as the open-source community provides continual enhancements to the product.
Knowledge Source	http://www.globus.org http://www.globus.org/toolkit/docs/4.0/key/GT4_Primer_0.6.pdf

Development Environment:

Name	***Microsoft Visual Studio 2003***
Description	In general, when implementing, it is useful to have a powerful development environment that integrate sdifferent tools and simplifies the task of developers. The use of Microsoft Visual Studio in Akogrimo is a direct consequence of the adoption of MS technologies as a possible deployment and development platform. Visual Studio includes features for simplifying the building of XML base web services.
Knowledge Source	Visual Studio 2003: http://msdn.microsoft.com/vstudio/previous/2003/ Visual Studio 2005: http://msdn.microsoft.com/vstudio/tour/evaluation/default.htm

Name	***Eclipse***
Description	A widely used, comprehensive development environment, available on both Windows and Linux.It has been used within the Akogrimo prototype for developing Java classes which are then used as the basis for services. However its use is optional and any software developed using it can be used without it.
Knowledge Source	The home page provides links to a wide range of resources on Eclipse. The system when downloaded ,contains links to tutorials. http://www.eclipse.org There is also a collection of tutorials for Java development at: https://eclipse-tutorial.dev.java.net/

Standards

Name	***XML***
Description	eXtensible Markup Language - A language for defining markup languages, that now provides the basis for many Grid, Web and Networking specifications – a recommendation by W3C.
Knowledge Source	http://www.w3.org/XML/

Name	***Web Service***
Description	One of the main requirements of Akogrimo is interoperability. Interoperability should be addressed in different areas but in the simplest case we can talk about interoperable communication. In practice, this means that services implemented on different platforms can communicate cross-platform. Of course ,this means a standardized way to define the message exchange is needed but this also means a common agreement on the semantic of the exchanged messagemust be found. In order to achieve this goal many different areas are affected (e.g. interface description, message definition, securityetc.). In past years, the Web Service technologies have becomesynonymous with interoperability and many specification have been defined to cover different areas (e.g. WSDL for service description, SOAP for message exchange, WS-Security for security aspects, …) .It is clear that in Akogrimo we find it useful to utilise these existing standard sand specification sfrom Web Service world.
Knowledge	WS overview:

Source	http://www.w3.org/2002/ws/ http://msdn.microsoft.com/webservices/webservices/ How to create a WS in C#: http://abstractvb.com/code.asp?A=1006 How to create a WS in Java: http://java.sun.com/webservices/docs/1.6/tutorial/doc/ WS specifications: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnglobspec/html/wsspecsover.asp

Name	***BPEL***
Description	The adoption of Web Service technologies in Akogrimo has brought about the cdevelopment of Business Process Execution Language – An XML-based language for describing the execution of business processes using Web Services as the constituent activities. In fact, Akogrimo willsimulate resource s(hardware and software) as a Web Service and it has between the main requirement ,the possibility to carry out at run time ,different services (that is Web Service and WS-Resource). In order to do this, it is necessary to have a language to design the business process esthat describe the different services to be managed between the services and this should be a "standard" or "world wide aacceptance". This consideration caused the development of the BPEL and of a Business process engine to execute such BPEL script
Knowledge Source	BPEL: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel http://www-128.ibm.com/developerworks/library/ws-bpel/ BPEL engine: http://www.activebpel.org/info/intro.html http://en.wikipedia.org/wiki/BPEL

Name	***SAML***
Description	SAML – Security Assertion Markup Language – provides a means of transmitting information concerning authentication and authorisations. It is used within Akogrimo for sending such information to and from internal components that require authentication and user attribute information.
Knowledge Source	Official information about SAML by the OASIS committee: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security OASIS committee introduction: http://xml.coverpages.org/saml.html SAML is also described in the State of the Art of Mobile Grids produced by Akogrimo (deliverable D2.2.1): http://www.mobilegrids.org/modules.php?name=-UpDownload&req=getit&lid=16

Programming Languages

Name	***Java***
Description	The Java programming language is used within Grid software and can be used by Akogrimo applications. It is used in association with the Globus Toolkit 4 for developing new grid services.
Knowledge Source	http://java.sun.com/

Name	***C#***
Description	In Akogrimo we want to adopt a flexible approach. This means that different technologies and platform scan be used. This approach is also valid for programming languages. We have chosen the MS language, C# ,for several reasons: • It is better supported for developers work ingon MS platforms • A WSRF framework exists on the MS platform that allows the implement ation ofWS-Resource susing C# language • It is portable • APIs are available that can sit on underlying MS OS The programming language is recommended for developers usingMS platform and in particular ,MS.NET
Knowledge Source	http://www.c-sharpcorner.com/language.asp Tutorial: http://www.softsteel.co.uk/tutorials/cSharp/cIndex.html .NET framework: http://msdn.microsoft.com/netframework/programming/fundamentals/-default.aspx

Name	***C++***
Description	Although much Grid software in Akogrimo uses Java, some use is made of C++, particularly for accessing some native libraries. Akogrimo applications can also be developed using C++.

Name	***Python***
Description	The Python interpreted interactive object-oriented programming language can be used by an application making use of the mobile grid. Although not used to any significant extent within the Akogrimo prototype and not required for any Akogrimo application, it isa suitable language for preparing small applications that do not have demanding performance requirements. In its Jython form, it can make direct use of Java interfaces
Knowledge Source	Python tutorials and reference material: http://www.python.org Jython tutorials and reference material: http://www.jython.org/

Database

Name	***SQL Server***
Description	The use of a relational database is fundamentalin the implementation of a complex infrastructure ,as in Akogrimo. It is clear that such a database will be based on the standardized SQL language. In Akogrimo we will notimpose the use of a specific product ,but product scan be used depending on the requirements of platforms. The term SQL Server is generally used to refer to the MS product, which provides a server for SQL databases. Can be used in association with .NET.
Knowledge Source	http://www.microsoft.com/sql/2005/default.asp http://www.ispras.ru/~gsql/ http://dev.mysql.com/doc/refman/5.0/en/tutorial.html

Name	***XINDICE***
Description	Due to the strong need for interoperability, in Akogrimo we will use XML based languages and specification as often as possible. This can introduce an overhead to manipulate XML documenst and store them. For this reason ,in some cases it could be useful to use a native XML database that allows the simplifying ofmany operations using XML documents. XIndice is a product from Apache that enables access to a native XML database.
Knowledge Source	http://xml.apache.org/xindice/

Name	***MySQL***
Description	Open source database management system that runs on both Linux and Windows as well as other platforms.
Knowledge Source	http://www.microsoft.com/sql/2005/default.asp

Web-Server:

Name	***IIS***
Description	Akogrimo will use Web Service and WS-Resource. Independently from the adopted platform, these entities will run behind a Web Server. In Akogrimo we willpublish on the network Web Service implemented on MS platform, for which, we need a Web Server. The Microsoft platform provides the Internet Information Server (IIS) that carries out this role for services implemented on MS platform using MS .NET framework.
Knowledge Source	Web Server definition: http://www.webopedia.com/TERM/W/Web_server.html IIS: http://www.microsoft.com/WindowsServer2003/iis/default.mspx

Name	***Apache***
Description	The Apache Software Foundation is a supplier of a wide range of software used for application to application communication which is used widely for Grid and Web Service software, including Tomcat. It is produced in an open and participatory environment and released under the Apache Software Licence.
Knowledge Source	Single stop web page which provides access to every item of Apache software: http://www.apache.org

Name	***Tomcat***
Description	This is the web applications container used in the official Reference Implementation for Java Servlets. It can be also used for deployment of Grid Services.
Knowledge Source	http://jakarta.apache.org

Others

Name	***Legacy Application Integration***
Description	Software support for incorporating prior applications into an Akogrimo framework
Knowledge Source	D4.4.1: http://bscw.hlrs.de/bscw/bscw.cgi/0/75067

3.5 Conclusion of the analysis

The previous section discusses all key topics that are required to build an application on top of the Akogrimo platform. All necessary topics have been identified, briefly described and detailed documents have been indicated to enable navigation through the Akogrimo content.

The above described topics are classified in knowledge structure models named:

- "Service Types" that represents the knowledge structure of the service types identified in the section starting point.
- "Traditional Approaches vs. Akogrimo" that represents the different features of traditional platforms compared to the innovative Akogrimo platform.
- "Application Support Service Layer" that represents the key Akogrimo concepts required by an application that is built on top of Akogrimo.
- "Grid Infrastructure Service Layer" that represents key Akogrimo concepts that may be required by an application that is built on top of Akogrimo.
- "Technology Infrastructure" that represents the technological infrastructure that is necessary when implementing an application for Akogrimo.

The next chapter introduces the graphical modelling methods that are used to generate the Akogrimo Roadmap, whereas the above described knowledge structures are modelled using the PROMOTE® modelling method. The links to the identified knowledge sources are also modelled in the PROMOTE® modelling method to enable navigation within the different knowledge sources.

An important issue is the integration from the requirement specification to the knowledge support based on the analysis of this chapter.

4 Methods for the Akogrimo Roadmap

4.1 Method Integration

This section describes the integration of the three graphic modelling methods in the Akogrimo Roadmap. Figure 6 indicates that the two graphic modelling methods AKBPM and UML™ conceptually support the requirement specification whereas the graphic modelling method PROMOTE® supports the implementation and testing of Akogrimo service by providing knowledge.

Below,the graphic modelling methods will be introduced, the modelling method definition will be briefly described and the proposed procedure in generating the modes is described. A detailed description on the graphic modelling methods is presented in the Annex.

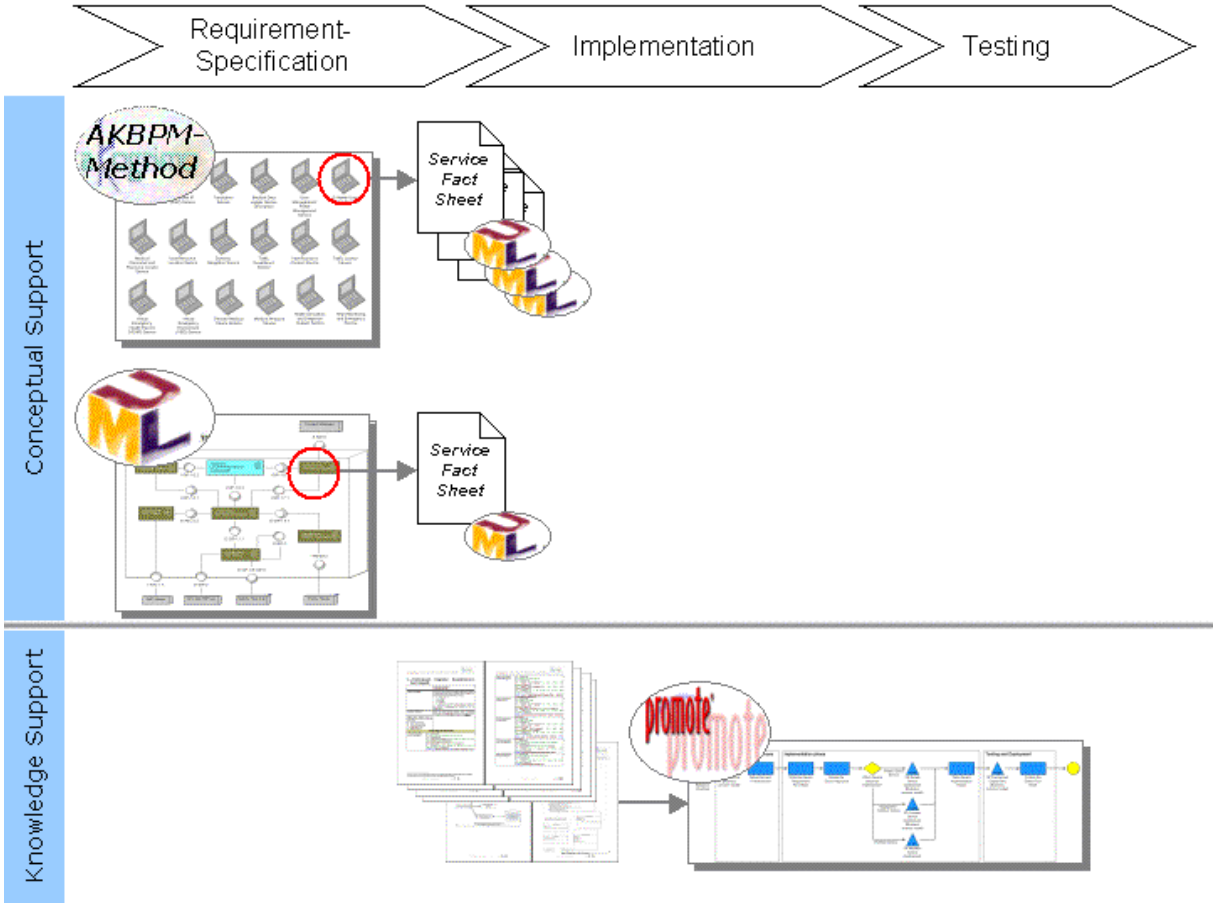


Figure 6 The Akogrimo Roadmap Detailed Method Mapping

Figure 6 describes the already introduced method mapping within the Akogrimo Roadmap in more detail. The Akogrimo Roadmap supports three aspects:

The Application Development Scenario in Akogrimo is conceptually supported by the AKBPM method, a business process oriented approach to derive so-called Business Service Sheets to specify the requirements of an application. Such a Business Service Sheet is a description of the functionality and the responsibility of a service that has to run on top of Akogrimo and is part of the composed application.

The Platform Development Scenario in Akogrimo is conceptually supported by the UML™ method, an object oriented modelling paradigm that supports the requirement specification of infrastructure services in Akogrimo. The specification of the service for the Akogrimo Infrastructure is seen as a deployment diagram that identifies the services and their interfaces.

The specification phase is completed by linking the so-called Service Fact Sheet – a technical description for service developers – either to the previously mentioned Business Service Sheets from the AKBPM method, or to the previously mentioned services identified in the deployment diagram of the UML™ method.

The Akogrimo Roadmap supports this linkage between one Business Service Sheet to many Service Fact Sheets or one service in the deployment diagram to one Service Fact Sheet by providing the UML-Use Case diagram in both of the methods. This enables to specify the Business Service Sheet and the service in the deployment diagram with Use Cases.

The definition of the Service Fact Sheet is seen as a trigger to start the implementation and of the service. This phase is supported by providing knowledge in the form of graphic models that have been modelled using the graphical modelling method PROMOTE®.

4.1.1 Requirement Specification with AKBPM

The first step of the development process of any software product is the collection and the concrete determination of the system requirements. Based on the results of this process we can then proceed to the design and the implementation of the system.

The design of a software system is a complex task. Many people contribute to it, adding different tasks covering different aspects of the system's structure. All these people need a common reference approach coordinating their actions.

In recentyears several modelling methods and methods have been developed to establish such a reference system. An experience report from IBM [17] points out that existing notations such as OOAD (Object Oriented Architecture and Design), EA (Enterprise Architecture) and BPM (Business Process Modelling) only partly support the SOA (Service Oriented Architecture) paradigm and need integration to be used as a reference approach.

Well known principles such as information hiding, modularisation and the distinction between topics are covered in common modelling approaches but additional topics dealing with service choreography, service repository or explicit service modelling are still missing in current modelling methods.

Currently there exists no standardised modelling method that integrates all of the above aspects. Therefore the Akogrimo roadmap is built to combine the various aspects and act as a common reference system coordinating the design and implementation actions between the involved partners.

The roadmap is used for depicting and visualising the foreseeable necessary steps that a group of collaborating people and organisations must follow in order to generate a product with desired features. The Business Process oriented approach is used in the Akogrimo roadmap via the AKBPM method whereas the object oriented aspects are used in the Akogrimo roadmap via UML™.

The AKBPM method has been selected to define the functional requirements as business process oriented methods ,like the listed approaches in the annex B . Thesetypically provide an end-to-end view of activities that can be seen as a set of functional requirements. If business process oriented approaches are used for software development they are mainly concerned in requirement definition as discussed in annex C.

The Akogrimo roadmap integrates three business process formats on three modelling levels:

- The ADONIS® business process format is a generic high level business process format that provides most of the required modelling concepts to define the domain of the system.
- The widely used Event Driven Process Chain (EPC) can be used as a more technical specification to derive the specification of the domain from the higher level process esand identify technical requirements.
- The third process method is related to the standard of the Business Process Execution Language for Web Service (BPEL) that is integrated into the AKBM method to provide an executable process modelling method.

4.1.2 The Conceptual Link from AKBPM to UML™

The above introduced AKBPM method enables the derivation of high level service requirement sfrom business process models. This high level service requirement description is called the Business Service Sheet. This Business Service Sheet can be allocated to a pre-defined service when the functionality is sufficiently covered by already existing services or by allocating a use case model when the service is implemented or revised.

Akogrimo therefore aims to synchronize the AKBPM method with UML™ by introducing Business Use Cases Models that can be modelled in the AKBPM method and define the service on a high business level and by System Use Case Models that can be modelled in the UML™ method to define Use Cases to describe the technical behaviour of a service.

A well-known method to achieve a tight synchronisation is the Feature Oriented Domain Analysis (FODA) [18].

The business processes defined in AKBPM aims to understand and model selected scopes of the applied domain, on a technical level. This results in a derivation of:

- definition of the domain, and it's relation with other domains,
- input and output data,
- identification of responsibilities per functions and
- a description of user expected system features in the context of the users' working scenario.

One of the most common means for depicting commonality and difference within a domain is feature analysis. A feature can be defined as any prominent or distinctive characteristic identified within a domain. One of the most commonly used forms of representation of the understanding derived from domain analysis is object-oriented.

Object-oriented practitioners usually coincide on the argument that a feature equates to a Use Case, part of a Use Case or a responsibility of a Use Case. Furthermore, object-oriented methods can be used in both descriptive and prescriptive forms of modelling.

The descriptive forms examine existing systems and known requirements and attempts to find commonality and difference between what may be seen to be an exemplary system in a domain.

Prescriptive forms represent decisions and commitments made with regard to future functionality. It also allows a conceptual analysis of the problems of the domain that can be used to solve existing problems or requirements in a domain and to identify and address future requirements.

Considering the evolution in the feature domain analysis and object oriented areas, it can be said that it is adequate to work on a one-to-one equation at this level of analysis. That is, a feature equates user requirement and, similarly, a user requirement equates to a system Use Case.

Therefore the linkage between the requirement modelling using AKBPM and the object oriented system design using UML™ can be synchronized by the Use Case diagram that is modelled as a feature list in AKBPM and as a User Requirement list in UML.

4.1.3 Service Development supported by PROMOTE®

The first thing to do, after collecting the system requirements, is to determine the configuration details of the intended platform. Generally, the platform being used is already defined as one of the major characteristics of the Akogrimo infrastructure. However, for the purposes of a specific application some minor modifications could be required.
This procedure also includes the determination of the design of all the software components which the developed services are based on, as well as of the relationships between components and services.
In parallel with this process any pre-existing software component or service developed in the context of the Akogrimo framework should be identified that could be used for the purposes of the new service being developed.
The result of the above procedure is the overall design, that is ,the trigger of the service development.
The service development has been analysed in detail by first modelling the development approaches per partner and second , byderiving a reference service development process to provide knowledge about the service development using the graphical modelling method ,PROMOTE®.
The procedure for developing services for Akogrimo, testing the services and deploying them is described in detail in Chapter 5.
The Akogrimo Roadmap for the service development guides the service developer through the steps involved ,starting from a service fact sheet that indicates the service requirements, the complex process of implementing a service to a full implementation report, describing the task-flow of the overall development procedure and finally, their results.

4.2 AKBPM

4.2.1 AKBPM: Introduction

The Akogrimo Business Process Management (AKBPM) Method is a new modelling method that has been built especially for the requirements of Akogrimo.

The method consists of:

- the Web-Modelling requirement method (for a detailed discussion on Web-based modelling methods please refer to the annex C, Annex: Excurse An Overview on Web-Modelling),
- enriched by a workflow specification (for a discussion on process and workflow standards please refer to the annex B Annex: Excurse on Process and WF-standards)
- enriched by the Event Driven Process Chains
- and completed by a service model

In the following this method is called AKBPM (a specification of this method is described in the annex D Akogrimo Business Modelling Method). 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.

4.2.2 AKBPM: The Modelling Method

The AKBPM (Akogrimo Business Process Management) modelling method is a process-oriented method to specify application requirements in a service-oriented manner. It enables the description of business actors interaction in a process-oriented manner and specifies organisational business processes using various concepts on workflow level. The workflow can be allocated by technical services to configure the application or by use cases to specify required services.

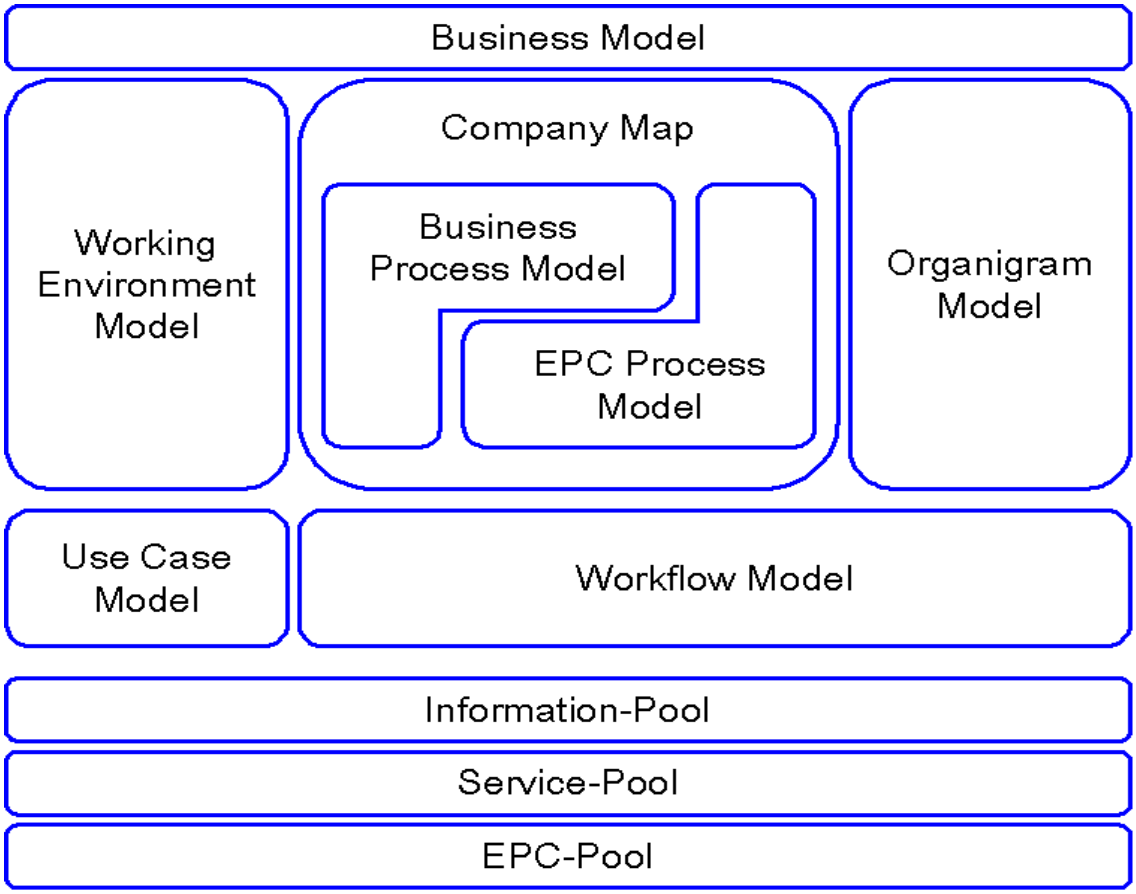


Figure 7 Overview of the Akogrimo Business Modeling Method

Figure 7 depicts the 11 model types of AKBPM, consisting of one overview and three pool models. The overview model company map, describes all identified business processes. It is expected that ADONIS® Business Process Models will be used on an organisational level although technical processes can be modelled as well, and the standard EPC (Event Driven Process Chain) process models will be used on a technical level although they can be used for organisational models. The linkage of the business process model and the EPC process model indicates this. In general ,the modelling layer indicate sa top down approach starting from the business view and approaching layer by layer ,more technical aspects.

The pool models collect information that is used in the above models This includes the collection of documents (Information-Pool), a description of services (Service-Pool) or a collection of all EPC-objects (EPC-Pool).

Below,the model types are briefly introduced:

Business Model:

This strategic model defines the business participants and the information, product and money flow between the different organisations. The relevant products and processes are identified and described on a more detailed level in other model types.

Company Map:

The Company Map is an overview model that identifies all processes within the business scenario. This includes business processes using the application that runs on top of Akogrimo.

Business Process Model:

The relevant business processes of each business participants are identified and specified in the business process model. This includes relevant business activities, business roles, performers and resources.

EPC Process Model (Event Driven Process Chain):

A more technical view on the process can be specified using the Event Driven Process Chain [19]. Although technical specifications can be defined in the business process model, some users especially from the eHealth scenario are familiar withthis kind of representation. An interface has been provided to import EPC Process Models from different modelling tools.

Working Environment Model:

The Working Environment model describes the organisational environment such as organisational units, performers and roles. This model type might be necessary when identifying persons or organisational units that are related with the Business Process Model.

OrganigramModel:

The Organigram Model describes the organisational units , performers and rolessimilar to the working environment. This model type is related to the Event Driven Process Chain.

Use Case Model:

The Use Case model is a specification of requirements of a so-called business case specified in UML. A common approach is to distinguish between business use cases that describe in detail the business case and the more technical use cases that focus on technical specification. Such an approach has been implemented to provide the relation between the business use cases in this model and the technical uses cases defined in UML.

Workflow Model:

The Workflow Model provides modelling constructs to define BPEL workflows. It is a technical model that allocates the required services to form executable workflows. An interface has been provided to export the workflows to Active BPEL engines.

Information-Pool:

The Information Pool collects all relevant documents that should be integrated into the requirement specification. This includes additional documents and links regarding application specification, business cases or guidelines.

Service-Pool:

The Service Pool collects all available services in Akogrimo that can be allocated in a workflow.

EPC-Pool:

The concept of Event Driven Processes requires a help model to store modelling objects that can be referenced. This model type is therefore a help pool to enable the EPC-interface and the EPC Process Model.

4.2.3 AKBPM: The Procedure

This section introduces the proposed procedure that can be used to specify application requirements in a process-oriented manner when defining application requirements for the testbed demonstration [20], [21], [22]. [23], [24].

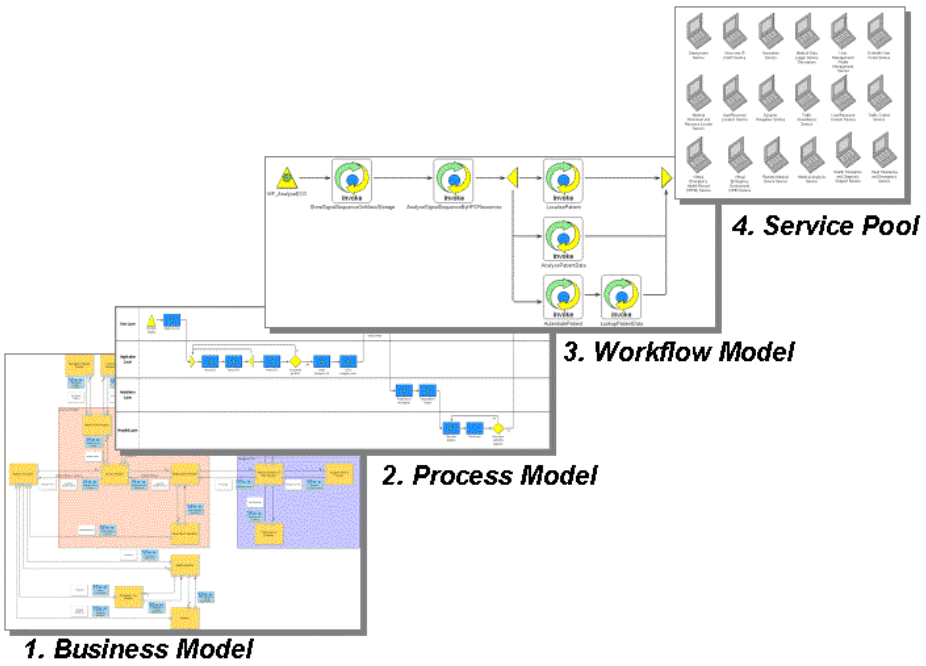


Figure 8 Proceeding of Roadmap modelling with AKBPM method

Figure 8 describes the proposed proceeding to identify application requirements in a process-oriented way. The AKBPM modelling method is provided to support the solution and application configuration of the demonstration test beds.

1) **Definition of the Business Model:**

   The business model enables the definition of the application scenario on a high business level. AKBPM enables the identification of the core business actors, the core product, information and money flows as well as the identification of the relevant business processes.

2) **Definition of the Processes**

   Core elements of this method are the different processes. The company map identifies the relevant business processes that can be modelled either in EPC format or in the ADONIS® format. The processes can be linked to enable a layered approach starting from high level organisational business processes to low level technical processes.

3) **Specification of Workflows**

   The workflows using Akogrimo services to provide an application can be deduced from the technical processes. This model enables the allocation of services to the workflow.

4) **Service Pool**

   The service pool is a collection of all Akogrimo services that can be used for processes and workflows. If additional services are required, these additional services can be identified in the use case model.

Sample models from D3.2.2 (The Business Modelling Framework) that depict the above procedure of the AKBMP modeling method are described in Annex G.

4.3 UML™

4.3.1 UML™: Introduction

The Unified Modelling Language is a well known graphical language to specify system artefacts [25], [26], [27], [28]. Although it is basically a generic modelling method it is widely accepted as defacto standard notation for software blue prints, especially in the object-oriented paradigm.

In Akogrimo UML 2.0 has been used to enable the documentation of the platform and the technical specification of services. It is differently used than the above two methods, as it is not used as a guideline but as a description of relevant software aspects.

The challenge was to integrate the UML diagrams that have been designed in various workpackages into the Akogrimo roadmap and 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.

4.3.2 UML™: Modelling Method

The modelling method UML 2.0 has been selected to describe the software architecture of Akogrimo.

The requirement for the roadmaps was to enable an integration of UML 2.0 with the above two modelling methods that are used for the Akogrimo roadmaps. This integration will be established by the Graphic Evolution Tool ,enabling references between these modelling methods.

The import from external modelling tools will be provided by an XMI interface for UML2.0.

The UML modelling method is not described as it is a well known standard in the software development community.

The procedure and the selection of the relevant model types will be proposed in workpackage 5.1 which is supported by the Graphic Evolution Tool from workpackage 5.4.

4.4 PROMOTE®

4.4.1 PROMOTE®: Introduction

PROMOTE® is a process-oriented knowledge management modelling method developed by BOC in the EU-Project PROMOTE. It has been applied in commercial products to describe the knowledge transfer, competencies and skills between knowledge workers in a process oriented manner.

In Akogrimo ,this modelling method is used to make the implicit experience of service development from workpackage 4.3 and 4.4 partners explicit in a process oriented way. Topic Maps are used to identify the core topics in Akogrimo and to classify knowledge resources like hyperlinks and documents.

The method consist of the process-oriented knowledge management model types of PROMOTE® (for a detailed discussion on process-oriented knowledge management modelling methods please refer to the annex E, Annex: Excurse on Process-Oriented Knowledge Modelling Methods)

It supports:

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

4.4.2 PROMOTE®: The Modelling Method

PROMOTE® is a knowledge management modelling method that describes the knowledge transfer between knowledge workers in a process-oriented manner [29], [30], [31]. In Akogrimo this method has been used to make the expert know-how in service development of the workpackage partners explicit. For a more detailed introduction into process oriented knowledge management modelling methods see annex E Annex: Excurse on Process-Oriented Knowledge Modelling Methods and on the PROMOTE® language annex F The PROMOTE® Modelling .

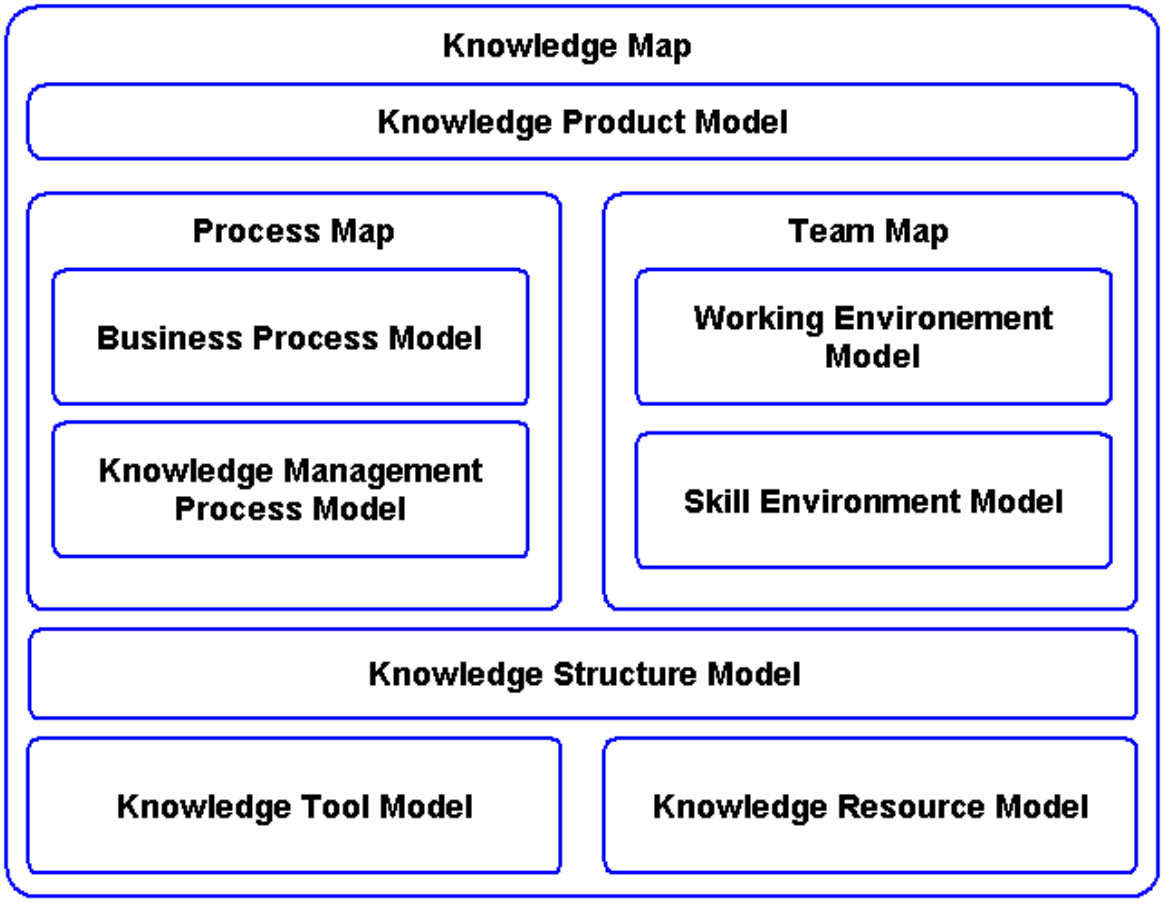


Figure 9 Overview of the PROMOTE® Method

Figure 9 depicts the 11 model types of PROMOTE®, consisting of three overview and two pool models. The overview models like the knowledge map, the process map and the team map enable to describe an overview of underlying aspects. The pool models like the knowledge tool model and the knowledge resource model collect external links. The other models provide constructs to identify certain aspects like sequences, knowledge flow, skills or organisational environment.

Below the model types are briefly introduced:

Knowledge Map:

The Knowledge Map model describes an overview on the whole roadmap. This model will be used to identify the process, structures and additional relevant information.

Knowledge Product Model:

The Knowledge Product Model describes how the information will be provided to the user. This model specifies the type of the product, responsibilities and provides a description o fhow the product will be used.

Process Map:

The Process Map is an overview model that identifies all processes within the roadmap. This includes business processes that in this scenario define the roadmap and knowledge management processes that define the knowledge transfer.

Business Process Model:

The Business Process Model describes a sequence of logical related activities. In this scenario this model type is used to define the roadmap.

Knowledge Management Process:

The Knowledge Management Process defines the knowledge flow within an organisation. In this scenario this model type is not used.

Team Map:

The Team Map is an overview model that identifies all human and organisational relevant aspects. This includes organisational units, departments, roles, employees or skills.

This model type will not be used in this scenario.

Working Environment Model:

The Working Environment Model describes the organisational environment such as organisational units, performers and roles. This model type might be necessary when identifying persons or organisational units.

Skill Environment Model:

The Skill Environment Model describes the skill-profiles of persons and optionally enables the definition of knowledge access rights. This model type will only partly be used in the roadmap scenario.

Knowledge Structure Model:

The knowledge structure model enables the classification of the content. This model type will be used to define the starting point and the end point of the roadmap as well as to define the organisational, technical, human and content structure.

Knowledge Tool Model:

The knowledge tool model is a pool model that collects and describes all relevant tool links.

Knowledge Resource Model:

The knowledge resource model is a pool model that collects and describes all relevant resource links.

4.4.3 PROMOTE®: The Procedure

This section indicates the procedure in modelling the roadmaps with PROMOTE®. During the acquisition of the Akogrimo Roadmap the procedure below had been applied by initially starting with models out of deliverables, then generating questionnaires and so-called acquisition tables. The feedback of the partners was discussed in telephone conferences and the modelling had been carried out during individual telephone workshops. Based on the results ,a new questionnaire or acquisition table was distributed to start the acquisition again. This procedure was repeated five times to end up with the Akogrimo Roadmap presented in Chapter 5.

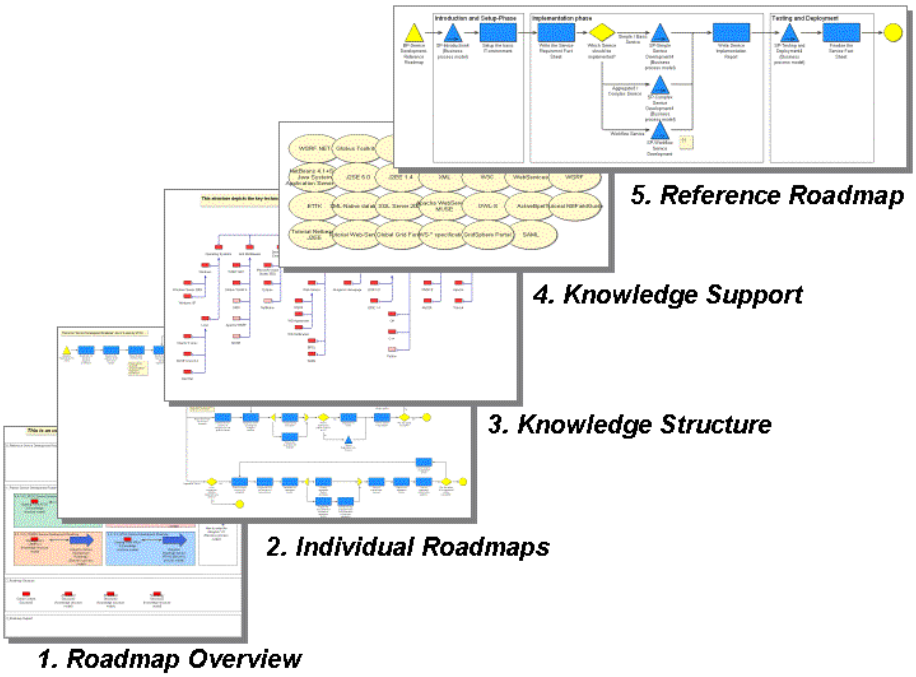


Figure 10 Proceeding of Roadmap modelling with PROMOTE®

1) **Define the roadmap overview:**

   The first model is the knowledge landscape that indicates an overview of the roadmap. It was conceptually drafted and refined according the modelling results. The result is an overview of the roadmap scenario.

2) **Definition of individual roadmaps:**

   First some conceptual drafts of individual roadmaps have been defined as an input of individual modelling sessions. The business process model was used to represent the individual development processes of the partners. The result is a process chain that describes each step from the starting point till the end point that had been modelled in physical modelling workshops and refined in telephone modelling workshops.

3) **Define the knowledge structure:**

   Based on the deliverables ,knowledge structures have been provided as an input to individual modelling sessions. These models have been redefined during workshops and enhanced by aspects that have been raised during the roadmap modelling. The result is a common structure of the relevant content.

4) **Definition of knowledge support (knowledge source) :**

   The next step was the collection of knowledge sources to support the knowledge structure and the roadmap. The result are pool models that collects knowledge resources.

5) **Generation of reference model:**

   The final step was an aggregation of the individual roadmaps to generate a reference roadmap. The aggregation is based on the analysis of individual roadmaps and a final review of the workpackage partners.

5 Roadmap to Akogrimo Convergence

This chapter describes interesting aspects of the Akogrimo Roadmap that are graphic representations of relevant aspects in Akogrimo using the PROMOTE® method. For a detailed description on the modelling method, please refer to the annex to read about the notation. They are seen as an integrative model that enables the navigation in relevant information and a stepwise guidance through the development of services.

The graphical representation is distributed via the Graphic Evolution Tool that currently publishes the Akogrimo roadmap at:

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

The roadmap is stored in a database that can be accessed by modellers to adapt the roadmaps and by users to view the models. They are therefore seen as "living documents" that are likely to adapt according to new experience from the developers.

The roadmaps therefore:

1. provide a graphic representation of key elements in Akogrimo to simplify the content,
2. provide a stepwise guideline in developing a service by pointing out key activities,
3. enable a navigation through the content, by providing links between the graphic representation,
4. integrate other information like documents, hyperlinks or portals into the graphic representation by providing links from the model to the actual document and

The following sections present some screen shots of the models to discuss the graphic representation and introduce the stepwise guidelines. For a detailed view on the navigation within the roadmaps and the integration of information, please use the links and account provided above.

Each of the individual roadmaps have been initially modelled from scratch, so the interviewed partner was not aware of the results of the other initial workshops. This gave a very interesting insight on the different views of service development. Some of the phases were modelled nearly identical but other phases of the model were totally different.

The differences had been discussed in telephone workshops and expert interviews to gain a common understanding. The initial individual roadmaps for example ,distinguished between sixteen different definitions of service types this affected the development phase. The result of the consolidation of the service types is described in Chapter 3.

The goal of this chapter is therefore to introduce the various models and to point out, interesting aspects like the discussion on services.

5.1 Overview and Navigation

The entry point into the Akogrimo Service Development is the "Overview and Navigation" model that uses the modeltype Knowledge Landscape to indicate the structure of the roadmap.

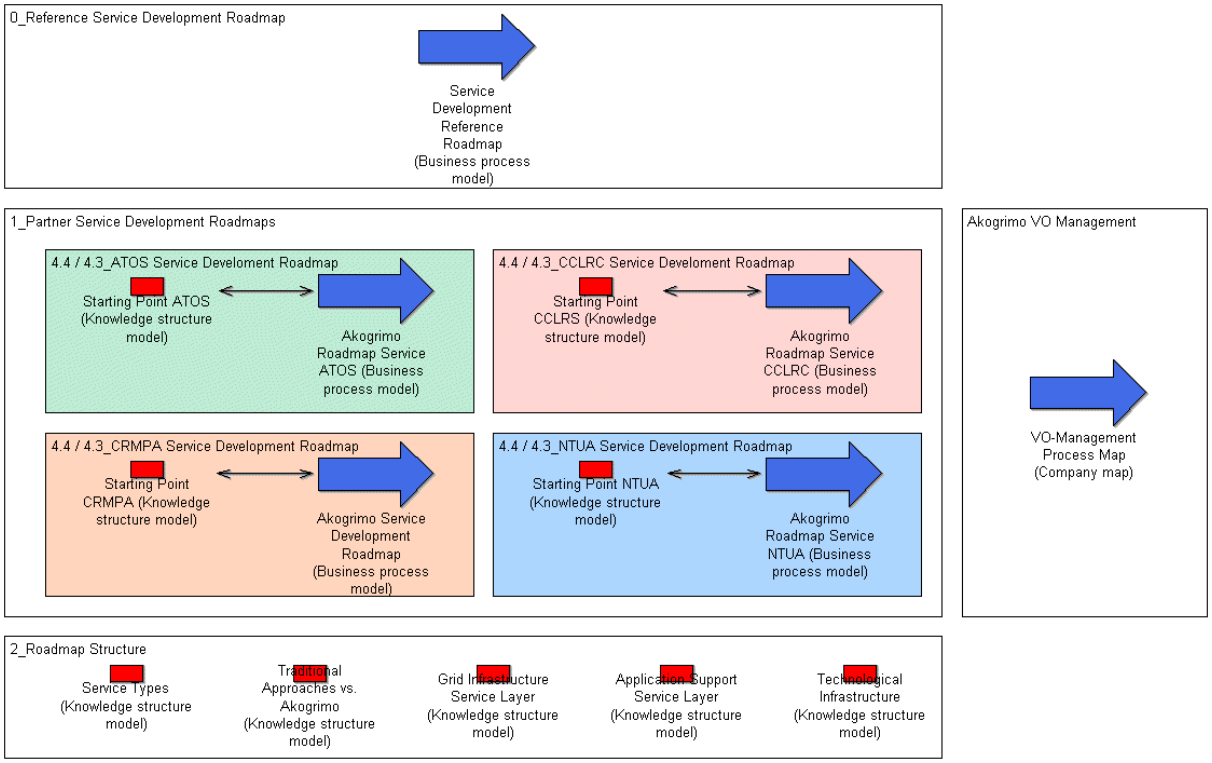


Figure 11 Overview and Navigation of Service Development Process

Figure 11 is a screenshot of the Overview and Navigation model of the Akogrimo roadmap that indicates the four individual Service development processes of the workpackage partners and the different starting points.
Beside the individual roadmaps the four structure models are indicated, the organisational processes within the VO and the Service Development Reference Process.
Each of the symbols (modelling objects) provide a hyperlink to the related model, so it can be used to navigate through the models.
It must be stated, that only the models that are relevant to the navigation are identified and not the supporting or detail models.
Beside the navigation feature of each modelling object there is also a short description inside the modelling object that explains in short term the modelled content.

5.2 Individual Roadmaps

5.2.1 ATOS

This section introduces the individual roadmaps in alphabetical order of the partners. The roadmap of ATOS shows an interesting procedure in the development of a service, after the setup of the IT-infrastructure and a brief introduction. The various tasks (blue rectangle) were derived by developing a dummy service object until the finished service that is deployed in the DMZ had been described, pointing out the different procedures in SLA template generation, testing and deploying. Three service types – basic service, customer service and aggregated service have been identified.

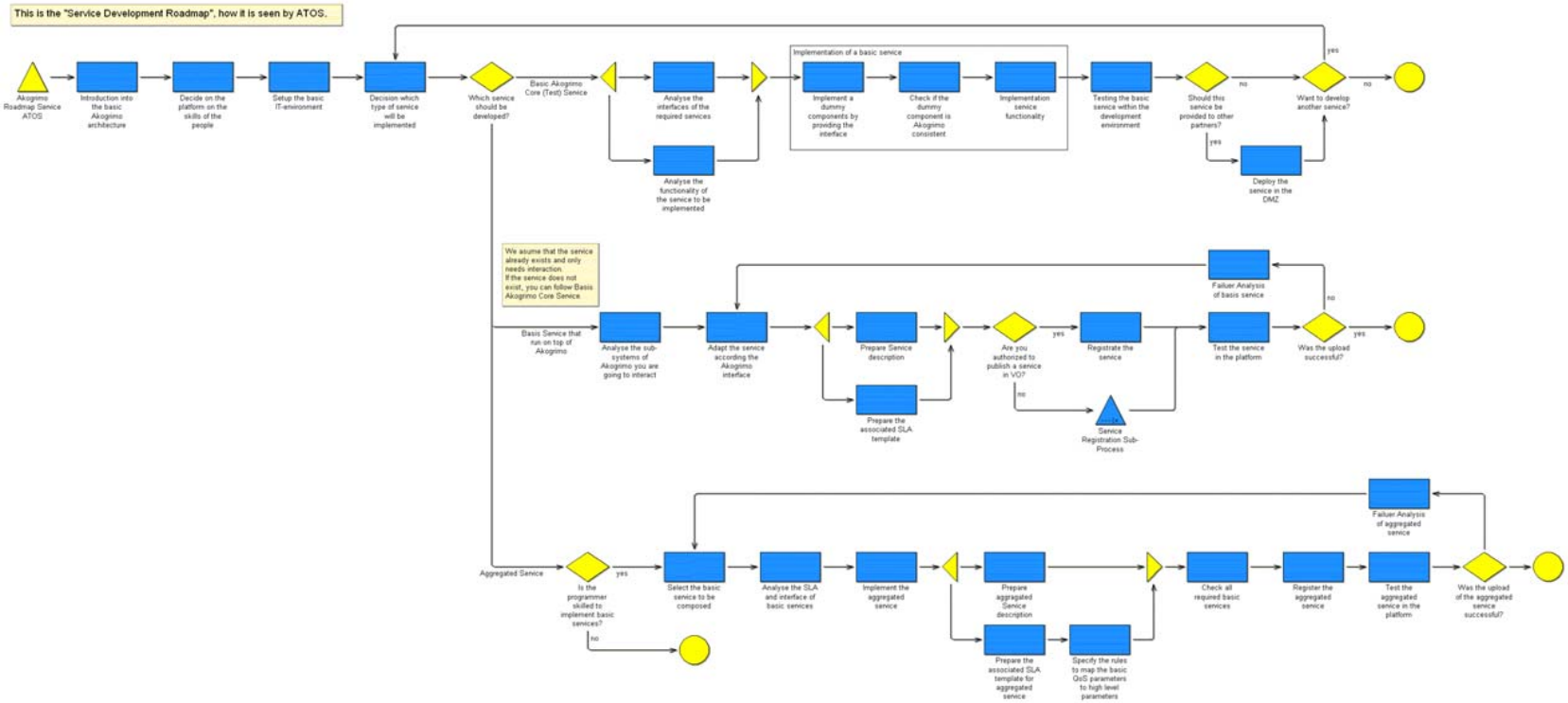


Figure 12 Individual Service Development Process by ATOS
for a detailed view please refer to http://83.65.190.82:8080/

5.2.2 CCLRC

The roadmap of CCLRC discussed a more sophisticated introduction distinguishing between Akogrimo partners and external partners. Two service types have been identified distinguishing between basic services and customer services. Two software development procedures have been modelled :the agile software development and the incremental development, whereas the incremental approach has been favoured.

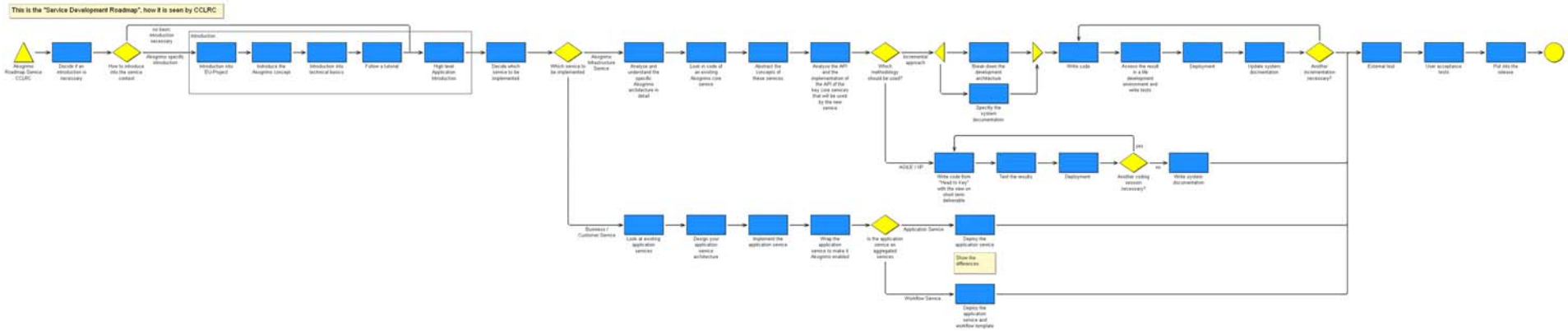


Figure 13 Individual Service Development Process by CCLRC

for a detailed view please refer to http://83.65.190.82:8080/

5.2.3 CRMPA

The roadmap of CRMPA uses four sub-processes that describe the skill check procedure, the implementation of a simple service, the interoperability check and the deployment in more detail. This roadmap created an interesting discussion regarding the definition of the various skill levels and the training concept to guide a beginner or an advanced programmer to the level, where an Akogrimo servicecan be implemented.

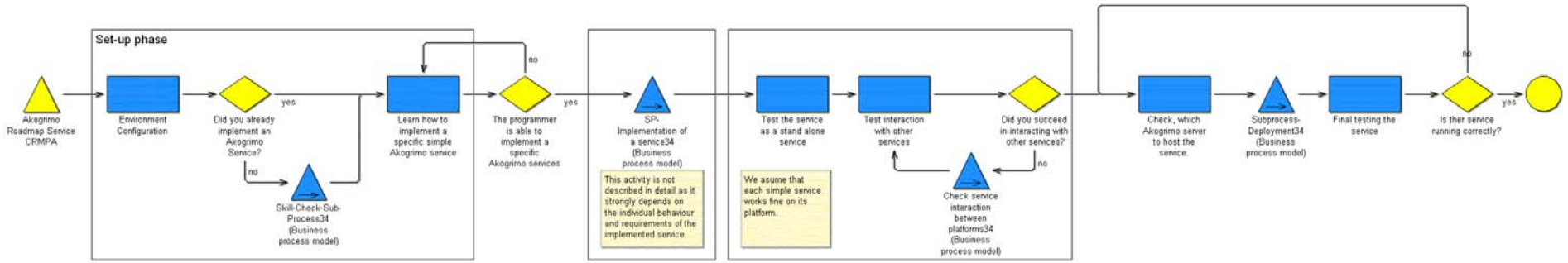


Figure 14 Individual Service Development Process (Main Process) by CRMPA

for a detailed view please refer to http://83.65.190.82:8080/

It has been observed that the setup of the technological environment, the introduction and the skill checking are the first tasks of all individual roadmaps. Although different starting points indicating different technologies have been identified there were no variation of the first activities of the roadmap. The service development phase however showed extreme differences between the partners that range from no differentiation on the various service types, to dependencies between indicated services types.

An interesting approach in training the developer with basic samples are indicated in the skill checking procedure of CRMPA, where a developer must provethat simple service can be implemented before the roadmap indicates further tasks.

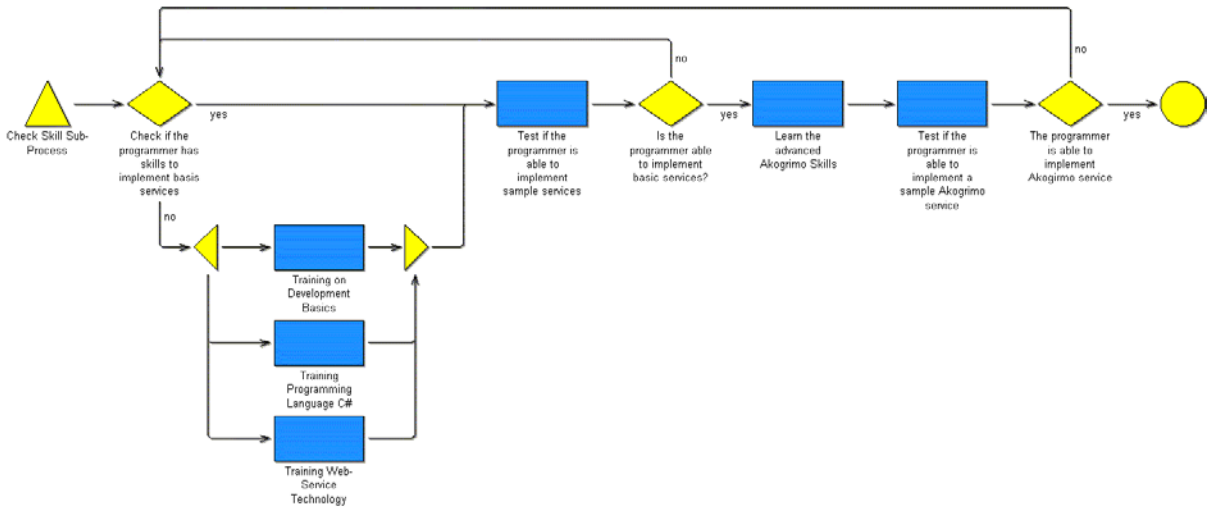


Figure 15 Sub-Process Skill Checking

for a detailed view please refer to http://83.65.190.82:8080/

Figure 15 depicts the skill checking and trainings procedure that has been indicated by CRMPA that proves that the developer is fit to implement an Akogrimo service. Ifthe developer does not havethe sufficient skills, he is looped back to training.

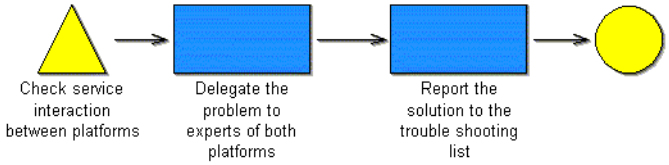


Figure 16 Sub-Process Service Interaction Problem

for a detailed view please refer to http://83.65.190.82:8080/

Figure 16 indicates the first draft of problem solving procedures. Currently there is no experience on problem solving procedures for Akogrimo services. So the process is trivially indicating to directly ask a specialist and indicating to build up a trouble shooting repository. This process is indicated to observe the trouble shooting during the development phases and enhance the process while getting more experience with the most likely Akogrimo traps.

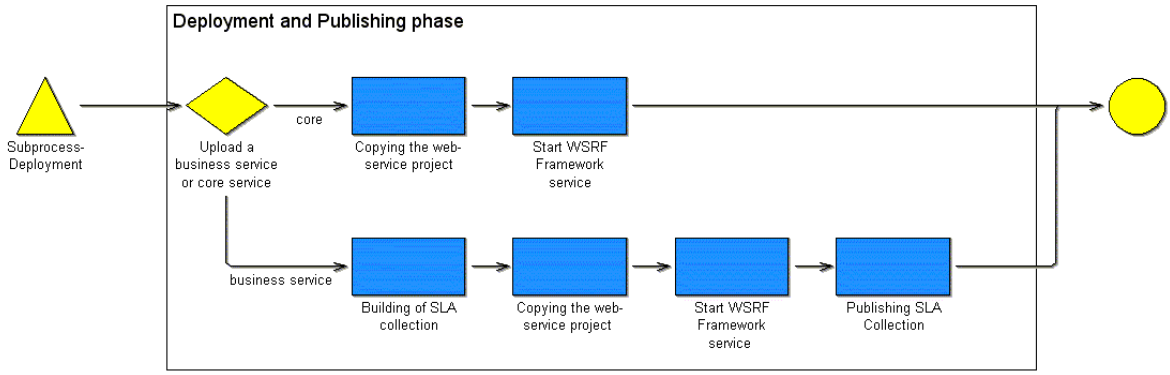


Figure 17 Sub-Process Deployment

for a detailed view please refer to http://83.65.190.82:8080/

Figure 17 describes the deployment phase in detail.

5.2.4 NTUA

The individual roadmap of NTUA indicated an implementation report per services. This concept has been improved to identify a document called "Service Fact Sheet" that indicates three stages. The first page is standardised and lists the service requirements, the second page is also standardised and lists the planed solution and how the service will be developed. This page enables possible reviewerstoleave comments, to describe the design decision in more detail. The third page starts the implementation report that describes implementation, testing and usage issues from the service to be followed by other partners. The last part is not standardised but leaves it up to the developer to document the service.
The procedure of the service fact sheet has been improved during telephone conferences and is currently provided in the layer WP 4.4 to coordinate the implementation of the services.

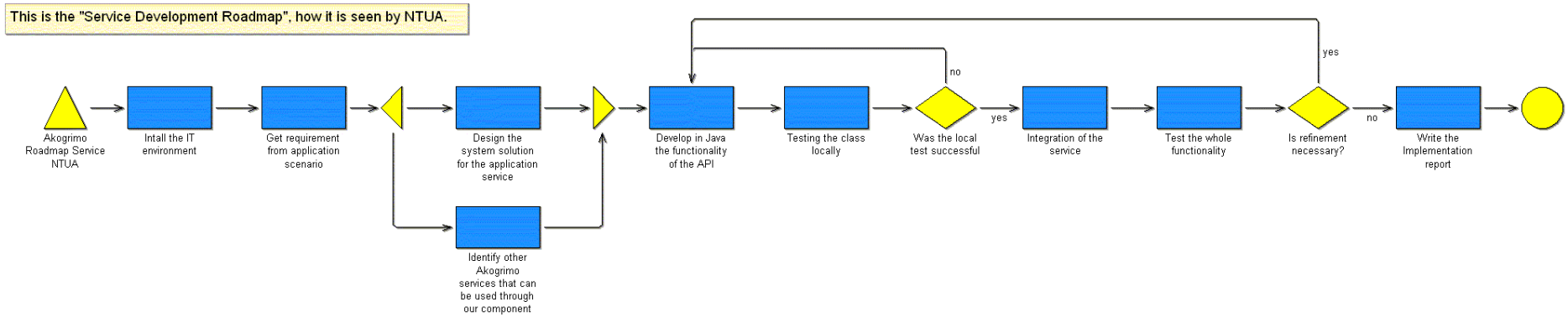


Figure 18 Individual Roadmap NTUA
for a detailed view please refer to http://83.65.190.82:8080/

5.3 Roadmap Structure

5.3.1 Traditional Approaches vs. Akogrimo Map

This structure model identifies the basic features of Grid middlewares that should be known by junior developers, the advanced features that should be known by advanced developers and the Akogrimo special issues that are Akogrimo specific.
The light red colour indicates topics that are identified but have no special meaning in Akogrimo. The red topics indicate that Akogrimo focuses on these issues and improves them.

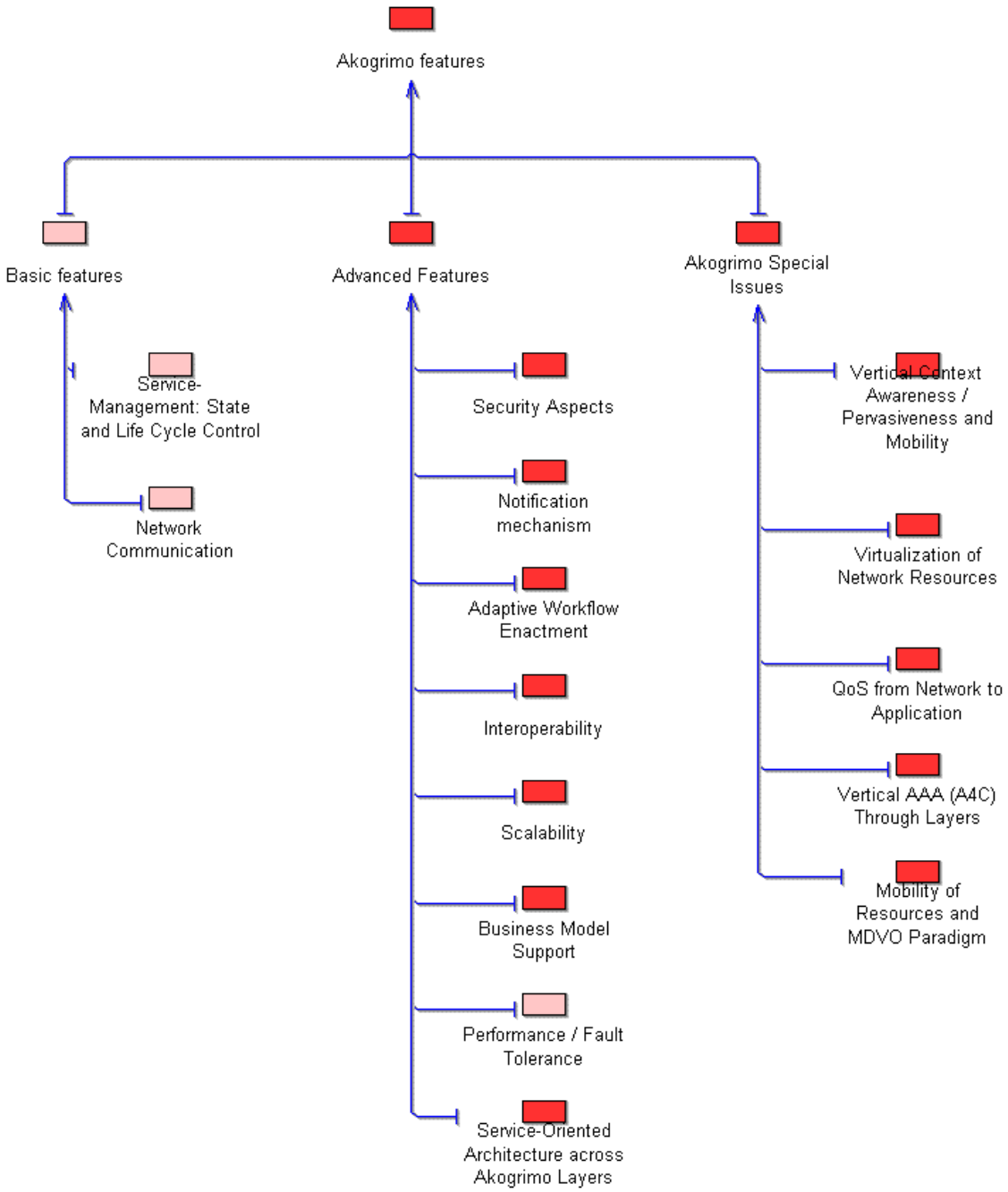


Figure 19 The Akogrimo Feature Map

for a detailed view please refer to http://83.65.190.82:8080/

Figure 19 depicts the model of the feature map that is described in section 3.2.

5.3.2 The Service Definition

As mentioned in the description of the individual roadmaps, there were several different interpretations of service types and service roles. This consolidated version has been developed, as each of the mentioned service types and service roles affect the roadmap differently. To generate a common view on the roadmap it was therefore necessary to have a common understanding.

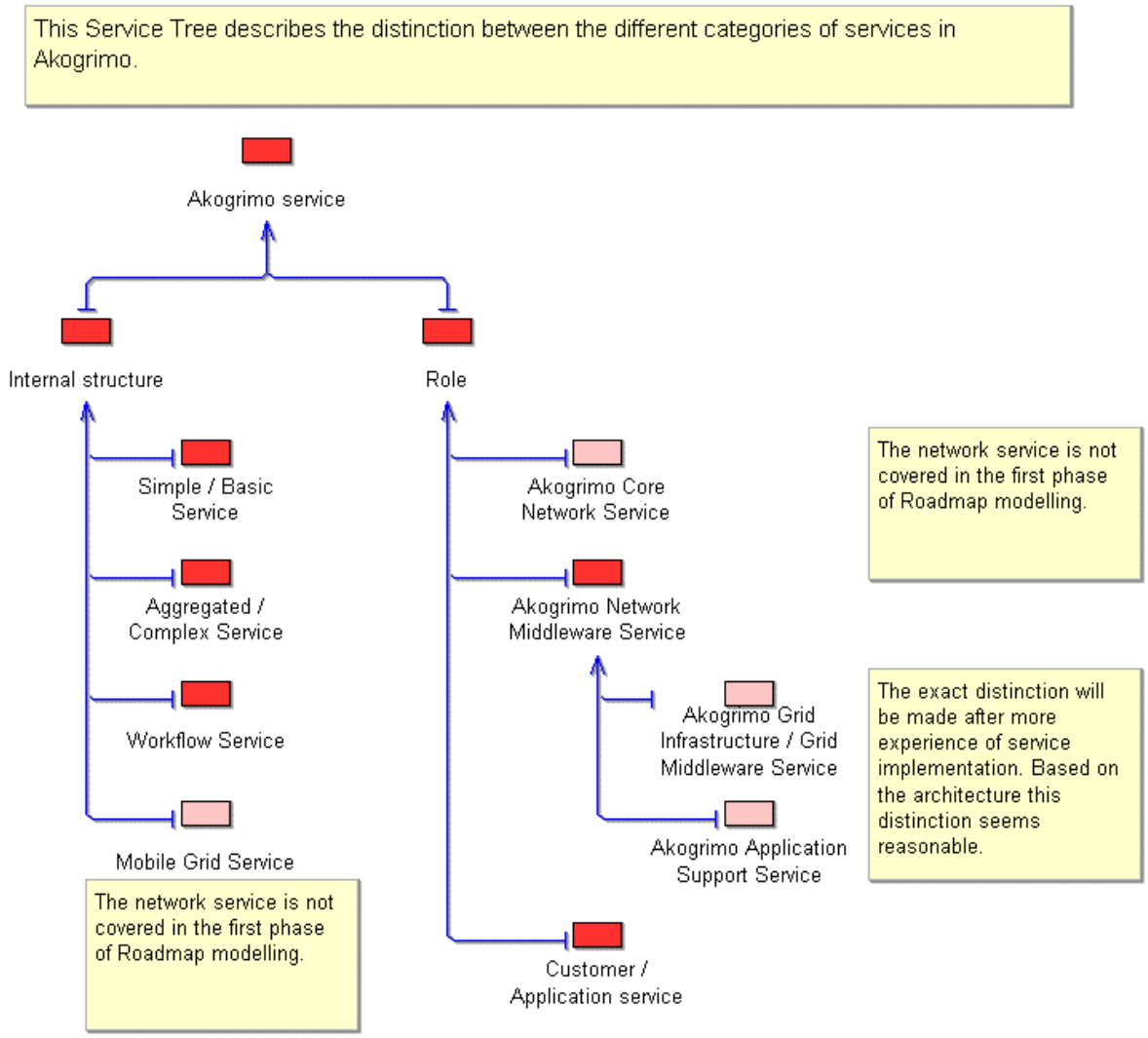


Figure 20 The Service Type Map

for a detailed view please refer to http://83.65.190.82:8080/

The light red service categories indicate that this category has been identified, but not considered in the roadmaps. The "Mobile Grid Service" and "Akogrimo Infrastructure Core Service – Network" category have been identified but not considered, as the focus of the workpackage was on WP layer 4.3 and 4.4. The "Akogrimo Infrastructure Core Service" and "Akogrimo Application Support Service" have been identified but not considered in the roadmap as there is too less experience available in the development of application services.

5.3.3 The Application Support Service Layer

The key content categories have been identified for the WP 4.4. layer in Akogrimo. The initial model has been identified based on deliverables and was refined during telephone workshops.

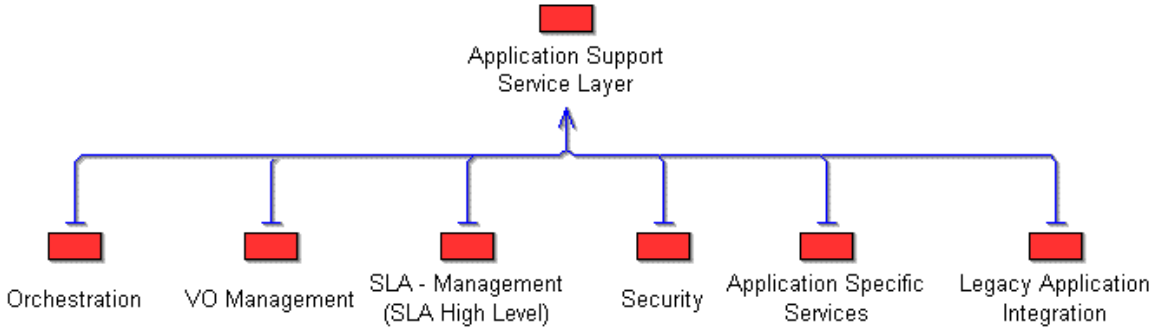


Figure 21 The Application Support Service Layer

for a detailed view please refer to http://83.65.190.82:8080/

Figure 21 depicts the content model for the Application Support Layer that has been introduced in section 3.

5.3.4 Grid Infrastructure Service Layer

The key content categories have been identified for the WP 4.3. layer in Akogrimo. The initial model has been identified based on deliverables and was refined during telephone workshops.

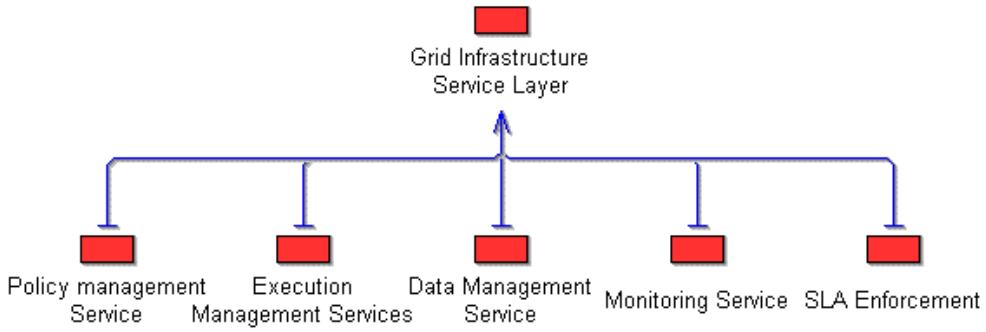


Figure 22 The Grid Infrastructure Service Layer

for a detailed view please refer to http://83.65.190.82:8080/

Figure 22 depicts the content model for the Grid Infrastructure Service Layer that has been introduced in section 3.

5.3.5 The Technological Infrastructure Map

One technological infrastructure has been identified for the WP layer 4.3 and 4.4 that was drafted based on internal development documents and refined during workshops. The light red categories indicate that they have been identified but not considered to be relevant for the core Akogrimo development. The programming language Python for example was identified because it is used for some testing procedures, but coloured light red, as the Akogrimo platform would not have any Python code.

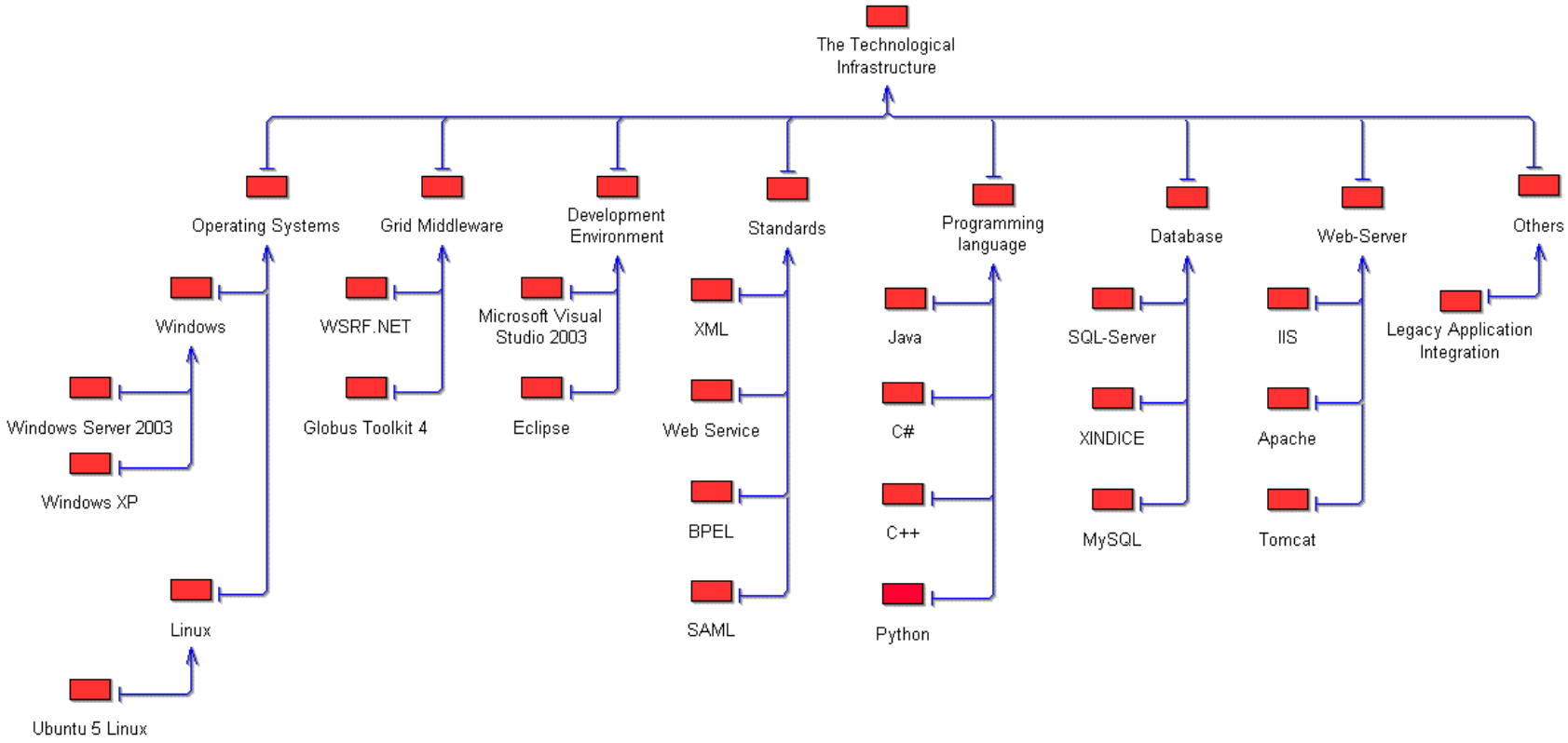


Figure 23 The Technological Infrastructure Map

for a detailed view please refer to http://83.65.190.82:8080/

Figure 23 depicts the model of the Akogrimo technological Infrastructure that has been introduced in section 3.4.

5.4 The Reference Roadmap

5.4.1 Service Development Reference Roadmap

The Reference Roadmap for Akogrimo consists of several sub-processes that describe the service development process in more detail and in four key activities that are described in the main process.

The first key-activity is the setup of the IT-Infrastructure, as it was listed in all individual roadmaps as one of the first tasks. The other three activities that are indicated in the main process are concerned with the writing of the service fact sheet. Before the service development starts, the service requirements are indicated, after the service development the implementation report is written and after the successful deployment, it is finalised.

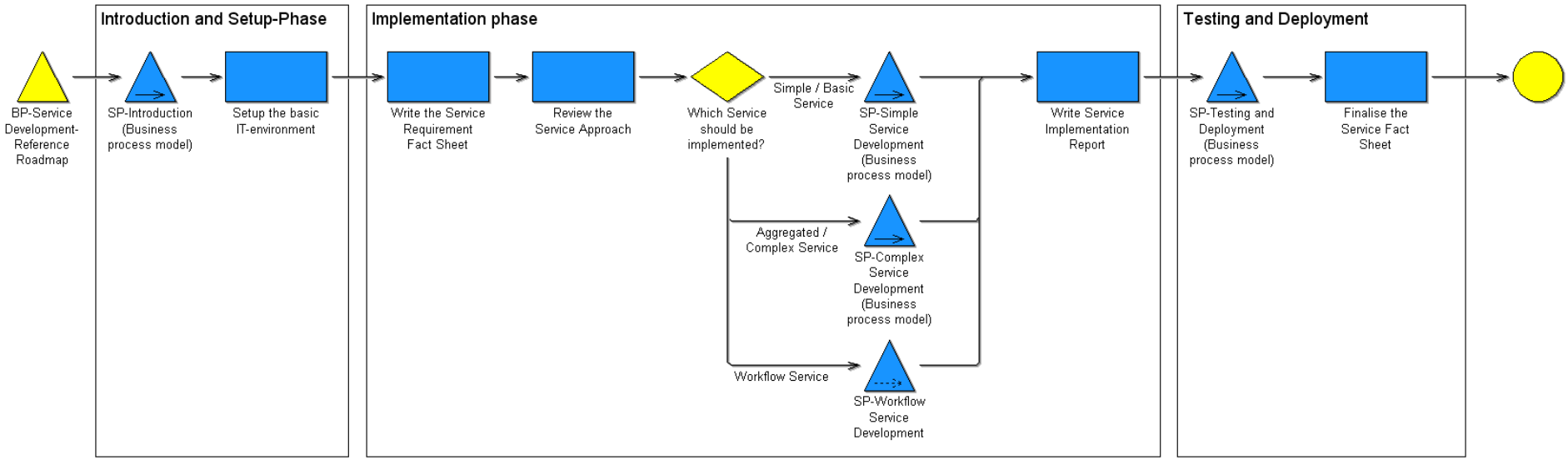


Figure 24 The Reference Service Development Process
for a detailed view please refer to http://83.65.190.82:8080/

5.4.2 The Sub-Process Introduction

The first sub-process of the Akogrimo Roadmap is the introduction process. First a general Akogrimo introduction is proposed where slides, the Akogrimo video and flyers explaining the Akogrimo idea are provided. If the developer is involved in the Akogrimo project itself, there is an introduction in EU-project preperation as well as in the project plan to gain in-depth knowledge of the workplan and the Akogrimo terminology. For junior programmers ,there is an introduction of basic elements. Finally the innovative Akogrimo concepts are introduced and architectural issues are provided on a more technical level.

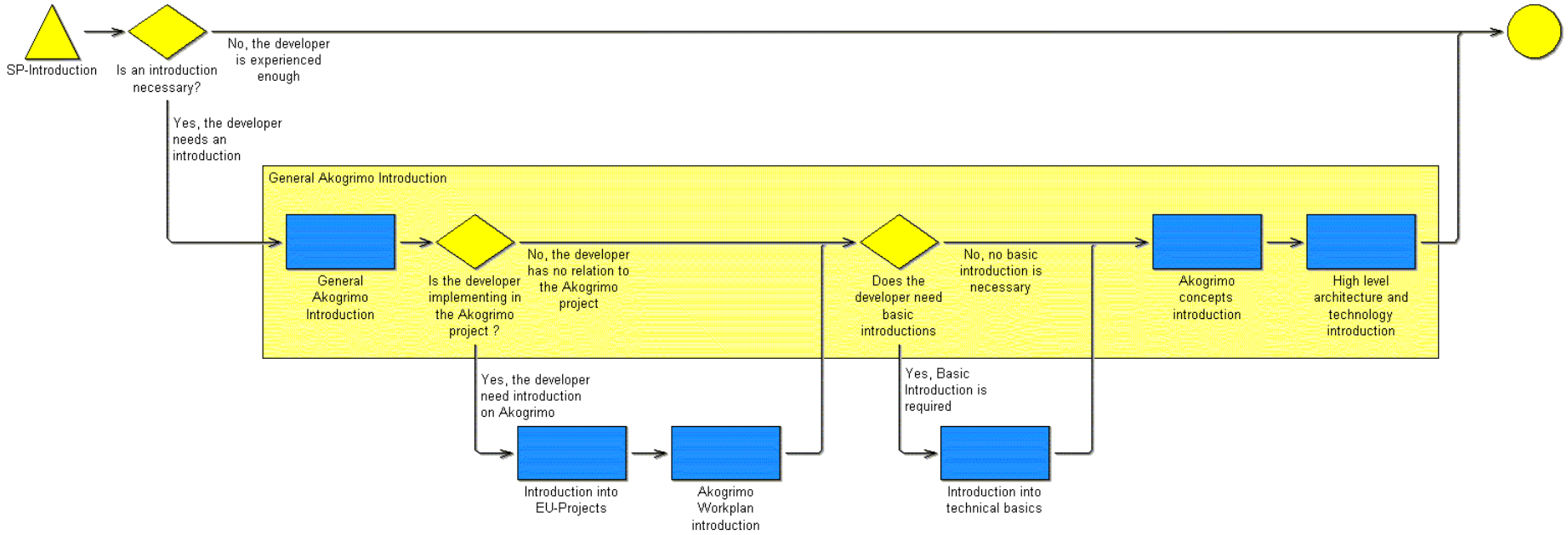


Figure 25 The Reference Service Development Process Introduction
for a detailed view please refer to http://83.65.190.82:8080/

5.4.3 The Sub-Process Simple Service Development

The implementation of a so-called Simple (Basis) Service depends if the service is an Akogrimo Infrastructure Service that provides some functionality of the platform or as a customer application service that implements a feature for an application. The proposed procedure for analysing, implementing and associating the SLA template and testing are slightly different.

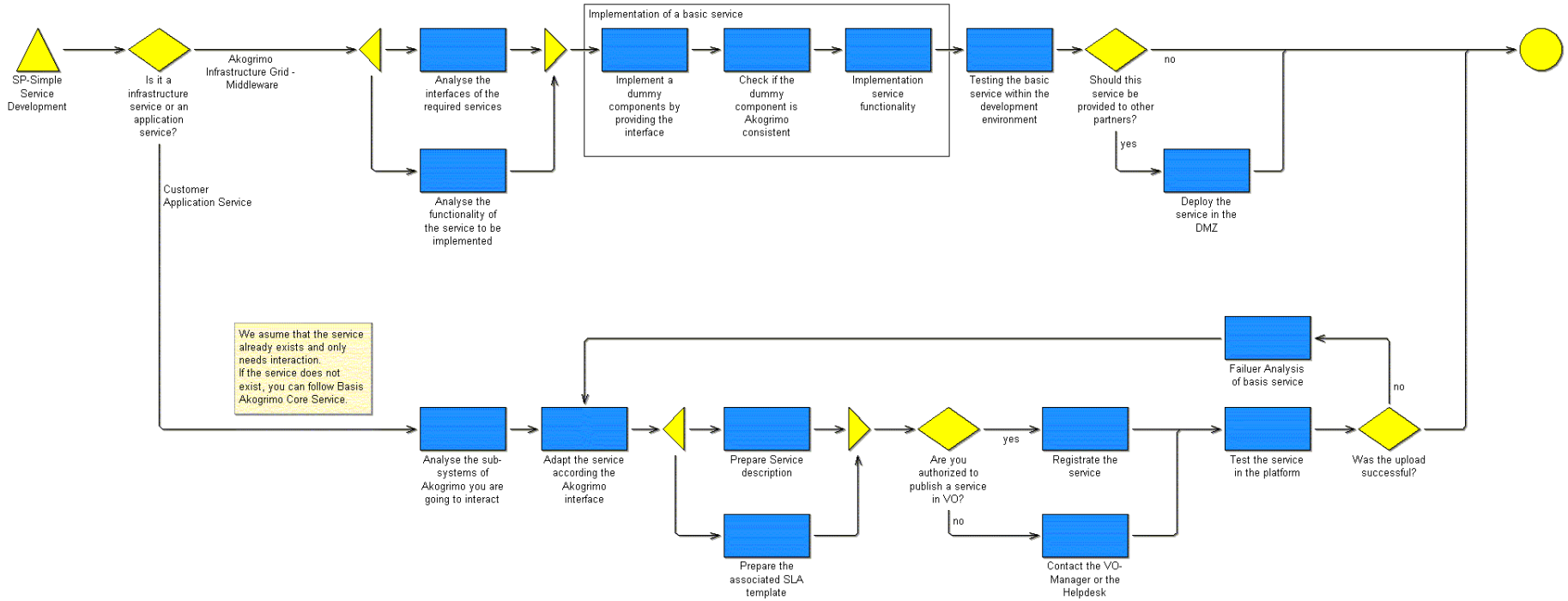


Figure 26 The Reference Simple Service Development Process, for a detailed view please refer to http://83.65.190.82:8080/

5.4.4 The Sub-Process Complex Service Development

The development of a complex service differentiates between complex (aggregated) Akogrimo Infrastructure Services and complex (aggregated) Application Services. In both cases there is a different procedure proposed for the collection of sub-services, the SLA generation and the testing procedure. In case of a complex application service development, the wrapping procedure, to integrate legacy application in Akogrimo is mentioned.

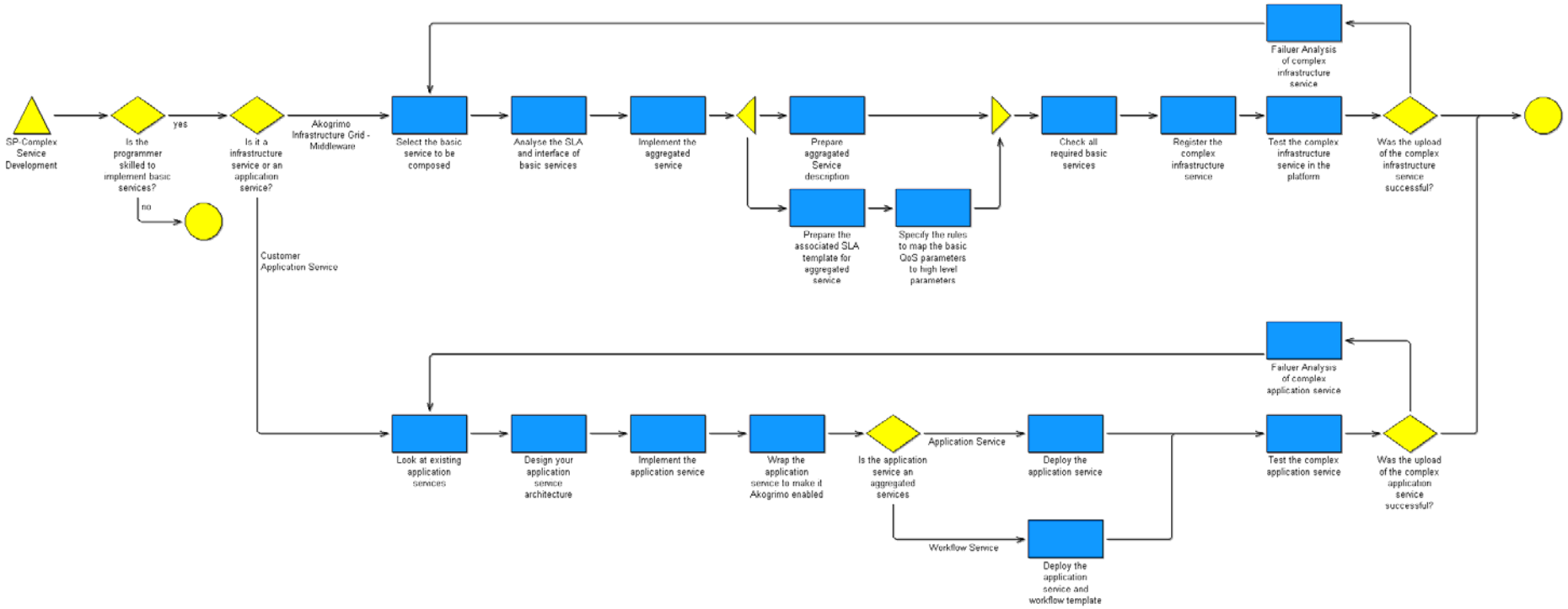


Figure 27 The Reference Complex Service Development Process
for a detailed view please refer to http://83.65.190.82:8080/

5.4.5 The Sub-Process Testing and Deployment

The final sub-process is the deployment and testing process. The deployment of the services is described before the four step testing procedure is recommended. It is expected that this part of the roadmap will be refined when first experience during the testing procedure of Akogirmo services will evaluate the roadmap. It is planned to install a Standardised Testing Procedure that is documented similar to the Service Fact Sheet.

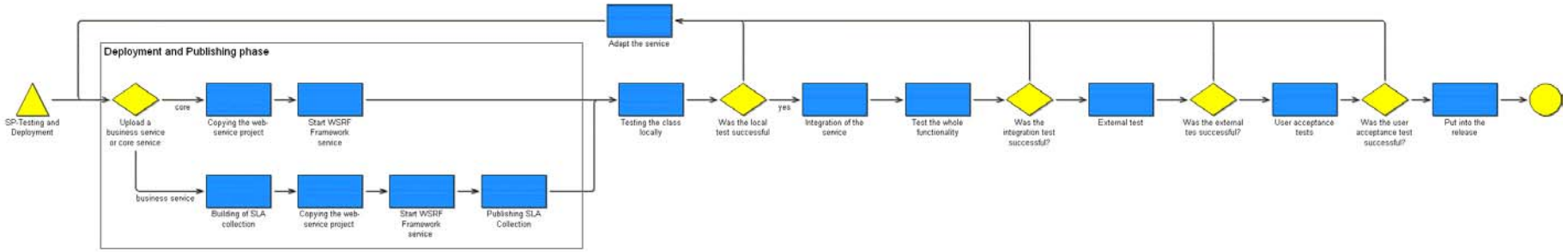


Figure 28 The Reference Testing and Deployment
for a detailed view please refer to http://83.65.190.82:8080/

5.5 VO –Processes as End Point

The previously described Akogrimo Roadmap provides support for the implementation and the testing of a service in Akogrimo. The VO-Processes of Akogrimo have been identified to describe how to proceed, if the service has successfully been completed.
The identification of VO processes means to describe the VO processes via a fact sheet describing the goal of the process as well as the input, output and trigger.
The goal is therefore to support the service developer in successfully registering the service by the Virtual Organisation (VO) ,which determines rights, restrictions and resources. In the Akogrimo vision the VO lifecycle is managed using two main concepts that are described in detail in Deliverable 4.4.1 (Architecture of the Complex Service Layer V1) in section 2.1.1.

- BVO (Base VO): this is a static organisation which has a broad task: let participants establish (search/negotiate) Operative VOs and provide basic identity management (profiles, participant registry…). A service may be published and associated with a Base VO but is not capable of execution within a VO until the creation of an Operative VO which includes that service.
- OpVO (Operative VO) is a dynamic organization with a specific task: let a user or a service achieve a specific business goal. In general behind an OpVO a workflow template and the overall users will be executed and services involved during the lifetime of a OpVO will represent a business process. An OpVO can only be created within the context of a BVO.

In particular the BVO is an enabler VO that offers services useful to create new VOs (OpVOs). Further ,the BVO is responsible for the formation of business agreements (among BVO participants) in order to drive the creation of the OpVO.

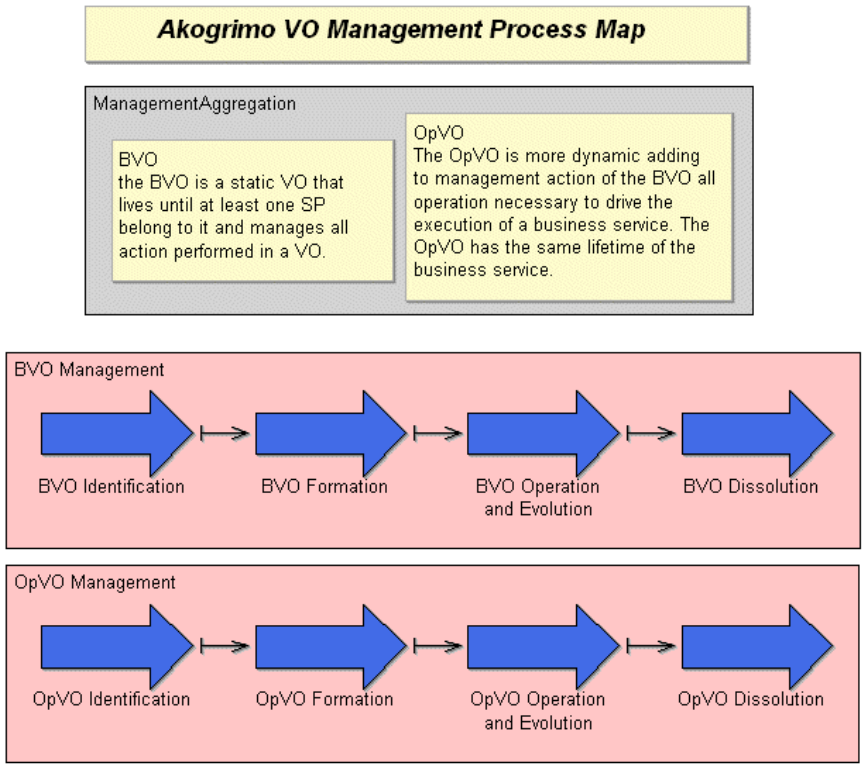


Figure 29 Akogrimo VO Management Process Map

The Akogrimo roadmap models the VO concept of Akogrimo using the highest level of detail ,the process map ,as shown in Figure 29 ,to enable an easy navigation through the VO concept of Akogrimo .Especially for developers who are not highly involved in the implementation of the Akogrimo platform.

5.5.1 VO – Process Fact Sheet

Generally the modelling of processes has four levels of details:

- the first and highest level of detail is the identification of the process in a process map,
- the second and more detailed level is the specification of the sequence,
- the third level of detail is the description of generic sequence's attributes and finally
- the fourth and most detailed level is the description of individualised attributes of the sequence.

The selection of the level of detail depends on the modelling objectives. For the VO-Management Processes the highest level of detail – the identification in a process map – with an additional process fact sheet has been selected, as this representation should support the service development by identifying the key processes and pointing to more detailed information resources.

In the following a standard process fact sheet is introduced that has been used to identify the Akogrimo VO-Processes.

Process Fact Sheet		
Process name	Process Name	
Purpose	General description of the process in 2-3 sentences.	
	Who are the main user of this process	What is the expectation of the process
Output	What is the output (the result) of the process?	
Input	What is the input of the process?	
Interfaces	Which processes provide input to the process?	Which process requires the result of this process?
	Who coordinates the process?	
Resources	Which human resources are required	
	What Information, Documents and Know How are required to perform the task?	
	Which infrastructure is required by the process?	
	What are critical success factors of this process?	
	Which elements are required? what are the elements that significantly bring the process into danger?	

Table 3 Explanation of Process Fact Sheet

5.5.2 Base VO (BVO)

BaseVO Identification		
Process name	BaseVO Identification	
Purpose	This process represents the first phase of a BaseVO life cycle. It involves: - Identification of participants - Identification of services being offered and required by participants - Roles definition	
	Conventional organizations intending to offer and use services.	To have a clear definition of the intended participants and roles expected to be required in the BaseVO.
Output	Identification of participants.	
Input	An understanding of the purpose of the BaseVO.	
Interfaces	It is the first process identified in the BaseVO management. The precondition is a set of services associated with each participant.	BaseVO Formation
Resources	D4.4., BP design tool, Workflow design tool	

Table 4 BaseVO Identification description

BaseVO Formation		
Process name	BaseVO Formation	
Purpose	This task is related to the setup phase of a BaseVO. The BaseVO is constituted from participants who may be providers or users of services. The role of each participant is specified	
	Conventional organizations intending to offer and use services.	Once the BaseVO is formed, OpVOs may be formed.
Output	The BaseVO is setup and ready to be used.	
Input	Action by initiating organisation	
Interfaces	BaseVO Identification BVO Management	BaseVO Operation and Evolution
Resources	D4.4, Grid and application middleware available	

Table 5 BaseVO Formation

BaseVO Operation and Evolution description		
Process name	BaseVO Operation and Evolution	

Purpose	This phase is related to usage of the BaseVO and in particular it foresees:
	- Usage
	  o User subscription Policy management
	  o Workflow execution support
	  o Creation of OpVOs
	- Administration
	  o Membership management (e.g. adding and removing users)
	  o Security checking and enforcement	

	Existing and new participants	To update the BaseVO
Output	OpVOs ready for use	
Input	Participant requests related to membership and application initiation.	
Interfaces	BaseVO Formation	BaseVO Dissolution
Resources	D4.4.1, Grid and application middleware available	

Table 6 BaseVO Operation and Evolution

BaseVO Dissolution description		
Process name	BaseVO Dissolution	
Purpose	This process has to release all resources involved in the BaseVO. This means at least: - Checking on OpVOs present - Depending on policy, releasing of participants and their associated services. All business relationships, created during the formation, have to be correctly cleared.	
	Participant with the appropriate role.	The BaseVO will be correctly closed.
Output	Final accounting report.	
Input	Dissolution request.	
Interfaces	BaseVO Operation and Evolution BVO Management	Last process in the BaseVO.
Resources	D4.4.1, Grid and application middleware available	

Table 7 Process tables for BaseVO Dissolution

5.5.3 Operative VO (OpVO)

OpVO Identification		
Process name	OpVO Identification	
Purpose	This process represents the first phase of an OpVO life cycle. It involves: - Business process design - Workflow template design - Roles definition - Identification of services (basic and aggregated) involved in the business process - Business process publishing	
	Business designer, workflow designer, application provider	To have a clear definition of what the application is and does.
Output	The business process description and the workflow template.	
Input	Application requirements.	
Interfaces	It is the first process identified in the OpVO management. As precondition, a Base VO must exist. BVO Management	OpVO Formation
Resources	D4.4.1, BP design tool, Workflow design tool	

Table 8 OpVO Identification

OpVO Formation		
Process name	OpVO Formation	
Purpose	This task is related to the setup phase of an OpVO. The OpVO is associated to a specific application and as subtask we can identify: - Business process discovery (mapped with a workflow) Negotiation of all services involved in the application - Security settings configuration for involved services - Roles assignment	
	Application Customer, Application Provider, Service Provider	To have an instance of OpVO, this means to have an application ready to be used.
Output	The OpVO is setup and ready to be used.	
Input	Application Customer request.	
Interfaces	OpVO Identification BVO Management	OpVO Operation and Evolution
Resources	D4.4.1, Grid and application middleware available	

Table 9 OpVO Formation

OpVO Operation and Evolution		
Process name	OpVO Operation and Evolution	

Purpose	This phase is related to the runtime execution of the OpVO and in particular it foresees:
	- Usage
	  o User access (e.g. login and logout)
	  o Policy management
	  o Workflow execution support
	  o Application QoS monitoring
	  o Accounting resources usage
	- Administration
	  o Membership management (e.g. adding and removing users)
	  o Security checking and enforcement	

	Application users, Application Customer, Service Provider	A controlled coordination and collaboration between all services involved into the execution of the application.
Output	OpVO monitoring execution and application final results.	
Input	User requests related to the specific application, data streams to be managed (related to the application or to context change) and exception events.	
Interfaces	OpVO Formation	OpVO Dissolution
Resources	D4.4.1, Grid and application middleware available	

Table 10 OpVO Operation and Evolution

OpVO Dissolution description		
Process name	OpVO Dissolution	
Purpose	This process mustrelease all resources involved in the OpVO. This means at minimum: - Negotiated Services releasing - Application workflow instance releasing All dynamic business relationships, created during the formation mustbe correctly cleared.	
	Application customer, application provider	The OpVO will be correctly closed.
Output	Final accounting report.	
Input	Dissolution request.	
Interfaces	OpVO Operation and Evolution BVO Management	It is the last process.
Resources	D4.4.1, Grid and application middleware available	

Table 11 Process tables for OpVO life cycle

6 Summary

The Akogrimo Roadmap is seen as a model-based process-oriented assistant system that guides the service developer through the Akogrimo content to support the development of services and applications on top of the innovative platform.

The development of such a system requires the analysis of three layers:

The Content:

The Akogrimo Roadmap supports the four phases of service development.

The Requirement Specification Phase is conceptually supported by providing a method and the according tools to describe the application scenario using business processes to derive service requirements. The service requirements can be specified using so-called Business Service Sheets to trigger the implementation of an Akogrimo service.

The Implementation Phase of an application or a service is supported by providing knowledge that is graphically represented on the development of Akogrimo services. The knowledge has been identified based on individual development processes, and deriving necessary key topics out of the process. The topics have been refined in so-called knowledge structures that are discussed in more detail in Chapter 3. The topics are identified and related to documents to provide more detailed information. Finally the individual development processes have been aggregated to a final Reference Roadmap to Akogrimo convergence.

The knowledge of the partners of the development of Akogrimo services have been acquired via questionnaires and acquisition tables, that have been discussed in telephone conferences and further refined in individual telephone modelling workshops. When finished with the individual telephone modelling workshops the cycle started again by generating new acquisition tables. The knowledge of the partners have been externalised in the form of graphical models and made usable by the reference of documents to the graphical models. The result of the roadmaps can be seen in Chapter 5.

The Testing Phase of an application or a service is supported by indicating the testing and deployment procedure in the roadmap.

The Roll Out Phase is supported by indicating the VO-Processes that describe the access handling of the Akogrimo VO.

The focus of the Akogrimo Roadmap is:

1. Support the requirement specification of applications by providing a method and a tool.
2. Support the implementation of an application that uses Akogrimo by providing knowledge.

The Method:

Three independent graphical modelling methods are necessary to provide the necessary concepts for the Akogrimo Roadmap. The initial prototype foresaw the integration of the necessary parts into one graphical modelling method. This was too complexand such a large modelling method would have been hardly usable. Therefore ,there was the separation according the different phases of service development and the different types of support.

This required the implementation of a new business process oriented modelling method that fulfils the requirements of Akogrimo (EPC, BPEL and the Business Service Sheet), the adaptation of the PROMOTE® modelling method to graphically represent the knowledge of the partners and to integrate the well-known UML™ modelling method using a conceptual synchronisation. This conceptual synchronisation is defined by the Business Service Sheets and the Service Fact Sheets as well as the concept of the Use Cases and enables a unique interaction of different modelling methods to support Service Oriented Development.

The *Akogrimo Business Process Management method (AKBPM)* has been developed based on the requirements of Akogrimo. This method can be used to specify the requirement by a top down approach that starts from the domain specific business model and enables the layered analysis of processes to a high technical level. This technical level supports the definition of Business Service Sheets and Use Cases Diagrams that will be used for the implementation of the services or application.

The *UML™* method has been used to design the services and the applications starting from requirements from the AKBPM method. UML™ models can be used for the design as required. A sufficient design of the service in the form of the Service Fact Sheet is seen as the trigger for the implementation that is supported by knowledge represented in PROMOTE®.

The *PROMOTE®* method is used to make the partners knowledge explicit by modelling the implementation process of an Akogrimo service. This process is seen as a guideline to navigate through the Akogrimo knowledge.

The goal of the Akogrimo Roadmap is to enable a homogeneous graphical modelling support that starts from the business view and continues to the design of the services and accompanies the developer during the implementation.

Chapter 2 discusses the usage concept of the Akogrimo Roadmap to identify usage scenarios of the Akogrimo Roadmap. A more detailed discussion on the graphical modelling methods is provided in Chapter 4 whereas a more detailed description is presented in the Annex.

The Tool:

A graphical evolution tool was developed to provide a model editor that implements the graphical modelling method mentioned above. This implies the configuration of a stand alone model editor and the adaptation of a web-based model editor that implements AKBPM, UML™ and PROMOTE®. Useful interfaces and additional Web-Components for the distribution, usage and documentation have been developed that provide different access to the graphical models.

The usage scenario of this tool is described in the Deliverable 5.4.2 Graphical Evolution Tool and was therefore not covered in this deliverable.

Concluding Remarks:

The result of this Workpackage is the technical infrastructure to enable the generation and usage of Akogrimo roadmaps, the conceptual background to use graphical modelling methods for service development in Akogrimo and the Akogrimo Roadmap that makes the existing know how in service development explicit using a graphical representation.

The modelling of the roadmap demonstrated that each partner had itsown interpretation of the service development process and the integration of the Akogrimo Reference Roadmap resulted in a common understanding and a common terminology.

It could be observed that modelling the knowledge structure resulted in a common understanding on the key topics, their relevance for service developer and the related documents.

The model based approach had been found useful to motivate and moderate discussions on the Akogrimo content and the procedure on service development. It is seen to accompany the development of a common understanding and the standardisation of development procedures. A clear advantage is the enhanced possibility to document existing knowledge using a graphical representation.

7 References

[1] Akogrimo Deliveralbe 5.4.2, Graphical Evolution Tool
[2] IST-2002-004293 Akogrimo – Annex I, pp 104
[3] Akogrimo Internal Deliverable 5.4.1, Initial Roadmap to Akogrimo Convergence
[4] Akogrimo Deliverable 3.2.2, The Akogrimo Business Modelling Framework
[5] Process-oriented Knowledge Management System based on KM-Services: The PROMOTE® approach, International Journal of Intelligent Systems in Accounting, Finance & Management Published onine in Wiley InterScience (www.interscience.com) Int. J. Intell. Sys. Acc. Fin. Mgmt. 11, 253-267, 2002
[6] Wikipedia Encyclopedia, http://en.wikipedia.org/wiki/Technology_roadmapping, access: 01.09.2005
[7] Kühn et al., Enterprise Model Integration, Proceedings of the 4th International Confernce ECWeb 2003, p3
[8] Kühn et al, Metamodellierug im Geschäftsprozessmanagement: Konzepte, Erfahrungen und Potentiale, Modellierung 99, March 1999, p 75-90
[9] Projektmanagement: http://www.kfunigraz.ac.at/inmww/Projektmanagement_SS04.pdf, access: 01.09.2005
[10] Die Welt des Projektmanagements: http://www.rkw-bw.de/Info/files/p05-0067.pdf, access: 01.09.2005
[11] 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
[12] Roadmap Repository http://83.65.190.82:8080, access: 02.12.2005
[13] Akogrimo Homepage, http://www.akogrimo.org, alternative:[http://www.mobilegrids.org], access: 02.12.2005
[14] Akogrimo BSCW Repository, http://bscw.hlrs.de/bscw/bscw.cgi/, access: 02.12.05
[15] Akogrimo Deliverable D 3.1.1, Overall Architecture Definition and Layer Integration, http://bscw.hlrs.de/bscw/bscw.cgi/d64103/
[16] Akogrimo Deliverable 6.3.1, Market Study,
[17] Zimmermann O., Krogdahl P., Gee C., Service-Oriented Architecture (SOA) http://www-128.ibm.com/developerworks/webservices/library/ws-soad1/, access: 28.11.05
[18] FODA, http://www.sei.cmu.edu/domain-engineering/FODA.html, access: 02.12.05
[19] Working group on EPC, http://www.epk-community.de/, access: 01.09.05
[20] Derntl M., Geschäftsprozessorientierte Anwendungsentwicklung – Ein Ansatz zur Kopplung von ADONIS® und Rational Rose, University of Vienna, Eigenverlag, 2001
[21] Riggio R., Managing Model Interoperability in Enterprise Modelling: an XMI-based Approach, Universita degli Studi Mediterranea di Reggio Calabria, Eigenverlag, 2004
[22] Kleiner N., Herbst J., A Model for Business Process Supporting Web Applications, SSGRR 2002S, L'Aquila, Italy, July 29 – August 4, 2002
[23] Kühn H., Modellierungssprachen für Web-Anwendungen: Klassifikation und Vergleich, Working Conference eBP 2001 – „Elektronische Geschäftsprozesse", University of Klagenfurt, 24 and 25 September 2001, P. Horster (Ed.), it Verlag pp. 379 – 390
[24] Kühn H., Murzek M., Bayer F., Business Model Interoperability using Enterprise Model Integration, Cunningham P., Bunningham M.,(Eds.): Proceedings of the eChallenges 2004 Conference, eAdoption and the Knowledge Economy: Issues, Applications, Case Studies – Part1, Vienna, Austria, October 2004, IOS Press, pp 233-240
[25] Larman Craig, UML 2 und Patterns angewendet – Objekorientierte Softwareentwicklung, mitp Bonn, 2005
[26] UML, http://www.omg.org, access: 01.12.05

[27] 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
[28] Jeckle M, UML 2.0 Die neue Version der Standardmodellierungssprache, http://www.jeckle.de/files/umltutorial.pdf, access: 23.0905
[29] PROMOTE, http://www.boc-eu.com/bochp.jsp?file=WP_582571cc1ed802de.b05236.f5980937ac.-7fdc, access: 01.09.05
[30] Woitsch R., Karagiannis D., Proceedings of the I-Know 2004 from the Special Track BPOKI'04 30th of June, Graz Austria 2004
[31] Woitsch R, Process-Oriented Knowledge Management: A Service Based Approach, PhD Thesis, University of Vienna, 2004

A. Annex: List of Tools and formats

A.1. Tools with XMI-Support:

Omondo

URL: http://www.omondo.com
Open Source: no
Costs: from €1 ,990
XMI Version: 1.2
Comment: XMI support will be available at the end of 2005 only.
Supported elements: N/A

myEclipse

URL: http://www.myeclipseide.com
Open Source: no
Costs: $29,95/year
XMI Version: 1.0
Comment: Supports roundtrip engineering but not XMI 1.2
Supported elements: N/A

Rational Rose

URL: http://www-306.ibm.com/software/rational/
Open Source: no
Costs: from $2,495
XMI Version: 1.0, 1.1, 1.2
Comment:
Supported elements: N/A

Fujaba

URL: http://wwwcs.uni-paderborn.de/cs/fujaba/
Open Source: yes
Costs: free
XMI Version: N/A
Comment: Supports reverse engineering but not XMI exchange.
Supported elements: N/A

JDeveloper

URL: http://www.oracle.com/technology/products/jdev/index.html
Open Source: no
Costs: free
XMI Version: 1.0
Comment: Supports reverse engineering but only XMI 1.0
Supported elements: N/A

Poseidon UML

URL: http://www.gentleware.com
Open Source: no
Costs: from €699
XMI Version: 1.2
Comment: Supports roundtrip engineering.
Supported elements:

Poseidon -> Akogrimo Modeller	Akogrimo Modeller -> Poseidon
Package diagram	Not supported.
Class model	
Attributes of modelling objects	
Methods declaration	
Relations between classes	

Enterprise Architect

URL: http://www.spaxsystems.com.au
Open Source: no
Costs: from $335
XMI Version: 1.0, 1.1, 1.2
Comment:
Supported elements: N/A

Telelogic TAU

URL: http://www.telelogic.com
Open Source: no
Costs: from N/A
XMI Version: 1.0, 1.1
Comment: Does not support XMI 1.2
Supported elements: N/A

Visual Paradigm for UML

URL: http://www.visual-paradigm.com

Open Source: no

Costs: from $699

XMI Version: 1.2

Comment: Supports roundtrip engineering.

Supported elements:

Visual Paradigm -> Akogrimo Modeller	Akogrimo Modeller -> Visual Paradigm
Package diagram	Package diagram
Class diagram	Class diagram
Attributes of modelling objects	Attributes of modelling objects
Method declaration	Method declaration
Interface specification	Interface specification
Relations between classes and interfaces	Relations between classes and interfaces
Comments	Constraints
Constraints	

This UML-XMI 1.2 interface for the Visual Paradigm Plug for Eclipse was implemented to demonstrate the export of models that are derived from Business layers to an operative development platform and to show a reverse engineering scenarios.

This configuration can be adapted to the required tools from the users, ifa different format mustbe supported.

A.2. BPEL Engines and Designers

ActiveWebflow Designer

URL:http:// www.active-endpoints.com,

Open Source: no

Costs: from $995

Supported elements:

Akogrimo Modeller -> ActiveWebflow Designer

BPEL processes can be exported and then imported into the ActiveWebflow Designer.

ActiveBPEL Engine

URL: http://www.activebpel.org

Open Source: yes

Costs: free

Supported elements:

Akogrimo Modeller -> ActiveBPEL Engine

BPEL processes can be exported and then imported into the ActiveBPEL Engine.

Oracle BPEL Process Manager

URL: http://www.oracle.com/technology/product/bpel

Open Source: no

Costs: free

Supported elements:

Akogrimo Modeller -> Oracle BPEL Process Manager

BPEL processes can be exported and then imported into the Oracle BPEL Process Manager.

A.3. Modelling Tools

ADONIS®

URL: http://www.boc-eu.com

Open Source: no

Costs: from € 3.600

Comment: Business Modelling Tool

ARIS

URL: http://www.ids-scheer.de

Open Source: no

Costs: N/A

Comment: Business Modelling Tool

Supported elements:

ARIS -> Akogrimo Modeller

An interface is provided which enables the import of ARIS EPC models.

Cinderella

URL: http://www.cinderella.dk

Open Source: no

Costs: from $2.500

Comment: Modelling and Simulation tool for UML and SDL

Microsoft Visio

URL: http://www.visio.com

Open Source: no

Costs: from $499

Comment: Modelling software for Microsoft Windows, supports UML Modelling

B. Annex: Excurse[4] on Process and WF-standards

B.1. Abstract

This excurse lists some basics on process and WF-standards that were used as basic consideration for developing the Akogrimo Business Process Modelling (AKBPM) method. The following discussion that is based on [1] has the aim to provide an overview on business modelling standards and their conceptual background that was used when developing the AKBPM.

B.2. Introduction

Various specifications for Web Service based Business Process Modelling and Web Service composition are currently available. These specifications contribute to the heterogeneity of XML interchange formats for Business Process Modelling. Different consortia including Object Management Group (OMG), Organization for the Advancement of Structured Information Standards (OASIS), Business Management Initiative (BPMI), United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT), World Wide Web Consortium (W3C), and Workflow Management Coalition (WfMC) and other individual software vendors as well as academic groups have proposed metamodels and related interchange formats for Business Process Modelling.

B.3. Process and WF-Standards Classification Criteria

Thirteen criteria are used for the classification of process and WF-Standards, as recommended in [1]:

- **Task I/O**. Refers to basic units or work whose temporal and logical relationships are modelled in a process. The input and output (I/O) of these tasks may be modelled using simple or XML complex types.
- **Task Address.** The address specifies how or where a service can be located to perform a task. The address can be modelled directly via a URI reference of a service or indirectly via a query that identifies a service address.
- **Quality Attributes.** When a set of potential services is generated via a query, quality attributes may be used to identify the best service.
- **Task Protocol.** The protocol defines a set of conventions to control interaction with a service-performing task. Web Services use SOAP as a protocol.
- **Control Flow.** The control flow defines the temporal and logical relationships between different tasks. Control flow can be specified via directed graphs, block-oriented nesting of control instructions, or process algebra.
- **Data Handling.** Data handling specifies which variables are used in a process instance and how the actual values of these variables are calculated.
- **Instance Identity.** This concept addresses how a process instance and related messages are identified. Correlation uses a set of message elements that are unique for a process

[4] The Excurse has not been developed during WP 5.4 but is mentioned to provide background information on the topic.

instance in order to route messages to process instances. The generation of a unique identifier, which is included in the message exchange, is an alternative approach.

- **Roles.** Roles allow for an abstraction of participants in a process. Roles are assigned to tasks and users are assigned to roles. A staff resolution mechanism can then allocate tasks of a process to users.
- **Events.** Events represent real-world changes. Respective event handlers provide the means to respond to them in a predefined way.
- **Exceptions.** Exceptions or faults describe errors during the execution of a process. In the case an exception occurs, dedicated exception handlers undo unsuccessful tasks or terminate the process instance.
- **Transactions.** ACID transactions define a short-run set of operations that have all-or-nothing semantics. They have to be rolled back when one partial operation fails. Business transactions represent long-running transactions. In case of failure a compensation process erases the effects of a business transaction.
- **Graphic Position.** The graphical presentation of a business process model contributes to comprehensibility. The attachment of graphical position information can be an explicit part of the meta model.
- **Statistical Data.** Performance analysis of a business process builds on statistical data such as costs or duration of tasks.

B.4. Comparison of XML-based specifications for Business Process Modelling

BPEL4WS

The Business Process Execution Language for Web Services (BPEL4WS or BPEL) was initiated by Microsoft and IBM. Since that time, this language has received the support of most market players. Since 2003, the standardisation organisation OASIS [2] has been in charge of the evolution of the BPEL language.
BPEL is specified as an interchange format via an XML schema. BPEL models call Web Services whose input and output are specified by messages and whose address is identified via Uniform Resource Identifiers (URI) of WSDL port types. SOAP is used as the communication protocol. Data handling is expressed via variables and related operations. Roles of process participants are defined via so called partner link types. Furthermore, BPEL supports handling of events and faults as well as compensation of transactions.

BPML

The Business Process Modelling Language proposed by BPMI [3] is very similar to BPEL. The main difference is that BPML allows the specifying of multiple processes in one XML document and also related communication between those processes. BPML is not tied to WSDL. Accordingly, the communication protocol is left to a BPML compliant implementation.

BPMN

The Business Process Modelling Notation was also developed by BPMI with the aim to unify the different graphical notations for business processes. The specification also provides a mapping to BPEL. Therefore, the BPMN metamodel reflects most of BPEL's concepts.

BPSS

The Business Process Specification Schema is part of OASIS and UN/CEFACT's work on ebXML. It includes a metamodel and XML Schema for Web Service choreography. It does not address implementation aspects like data handling or process instance identification. It supports the definition of roles, exceptions, and transactions in an inter-organisational message exchange.

EPML

The Event-Driven Process Chain (EPC) Markup Language (EPML) is an academic proposal. It captures the control flow elements of EPCs. Further aspects can be defined using extensions. It also includes graphical position information for each EPC object.

OWL-S

OWL-Services (OWL-S) is an academic proposal for a service metamodel represented in OWL. OWL-S builds on an input-output-preconditions-effects quadruple to describe services.

PNML

The Petri Net Markup Language (PNML) [4] is an academic proposal for an XML interchange format for Petri Net models. It supports the Petri Net syntax. The eXchangeable Routing Language (XRL) is based on PNML and can be executed on a dedicated infrastructure.

UML 2 Activity Diagram

Activity Diagrams of Unified Modeling Language (UML) can be exchanged using XMI. Their metamodel includes concepts to model input and output tasks, control flow, data handling, roles, exceptions and graphical information.

WSCI

The Web-Service Choreography Interface (WSCI) was developed by W3C [5] and provides a set of extensions to WSDL in order to describe process behaviour of message exchanges. Beyond input and output message types, WSDL bindings and correlation WSCI also supports roles, exception handling and transactions.

WSCL

Hewlett-Packard developed the Web-Service Choreography Language (WSCL). It includes message types, protocol and service location.

WSFL

IPM developed the Web Services Flow Language (WSFL). It is one of the predecessors of BPEL. It includes most of the concepts excluding transaction support, graphical position information and statistical data. Control flow in WSFL is modelled via directed graphs.

XLANG

Microsoft developed XLANG, which is the second predecessor of BPEL. It provides means for defining message correlation, roles, events and exception handling as well as transaction declaration.

XPDL

XML Process Definition Language (XPDL) is a standardised interchange format for business process models proposed by WfMC [6]. It includes various concepts like task input/output and address, control flow, data handling, roles, events, and exceptions. It is the only specification that addresses process statistics such as durations and costs.

Figure 30 Comparison of the XML-based specifications - shows the different specification in comparison. A plus sign indicates that the concept is included in the metamodel of the certain specification. A minus sign denotes that the concept is not included. The table shows that none of the specification already includes all of the 13 concepts. BPEL4WS, BPMN and WSFL yield the best results. BPEL is probably the most frequently used and widely accepted business process execution language. It has become the de facto standard for business process execution. The Akogrimo Business Modelling method supports the important elements of the BPEL specification and allows the modelling of BPEL processes. Furthermore the export of BPEL processes is supported, which can then be imported into, e.g. ActiveWebflow Designer.

	BPEL4WS	BPML	BPMN	BPSS	EPML	OWL-S	PNML	UML Act. D.	WSCI	WSCL	WSFL	XLANG	XPDL
Task I/O	+	+	+	+	-	+	-	+	+	+	+	+	+
Task Address	+	+	+	-	-	+	-	-	+	+	+	+	+
Quality Attributes	-	-	-	+	-	+	-	-	-	-	+	-	-
Protocol	+	-	+	-	-	+	-	-	+	+	+	+	-
Control Flow	+	+	+	+	+	+	+	+	+	+	+	+	+
Data Handling	+	+	+	-	-	-	-	+	-	-	+	-	+
Instance Identity	+	+	-	-	-	-	-	-	+	-	+	+	-
Roles	+	+	+	+	-	+	-	+	+	-	+	+	+
Events	+	+	+	-	+	-	-	-	-	-	+	+	+
Exceptions	+	+	+	+	-	-	-	+	+	-	+	+	+
Transactions	+	+	+	+	-	-	-	-	+	-	-	+	-
Graphic Position	-	-	+	-	+	-	+	+	-	-	-	-	-
Statistical Data	-	-	-	-	-	-	-	-	-	-	-	-	+

Figure 30 Comparison of the XML-based specifications

B.5. References of Annex B

[1] Mendling J., Neumann G., Nüttgens M., A Comparison of XML Interchange Formats for Business Process Modeling, Vienna University of Economics and BA and Hamburg University of Economics and Policitics, 2004

[2] Business Process Execution Language for Web Services Specification 1.1, 2003, http://ifr.sap.com/bpel4ws/ access: 21.11.05

[3] BPMI – http://www.bpmi.com, access: 21.11.05

[4] PNML – http://informatik.tu-berlin.de/top/pnml/about.html, access: 21.11.05

[5] W3C – http://w3.org, access: 21.11.05

[6] WfMC – www.wfmc.org/standards/XPDL.htm, access: 21.11.05

C. Annex: Excurse[5] An Overview on Web-Modelling Methods

C.1. Abstract

This excurse discusses some basics on web-modelling methods that support the requirement analysis, the software development and the hypertext engineering of web-applications that is used for the domain specific application adaptation roadmap. The aim is to discuss available web-modelling methods in more detail than in chapter 4.2, AKBPM. The chapters focus on roadmaps in Akogrimo and only briefly discussing the method analysis. This excurse provides some details on alternative methods and describes why the E-BPMS has been selected as the basis for the new AKBPM method. This analysis is based on [1]

C.2. Introduction

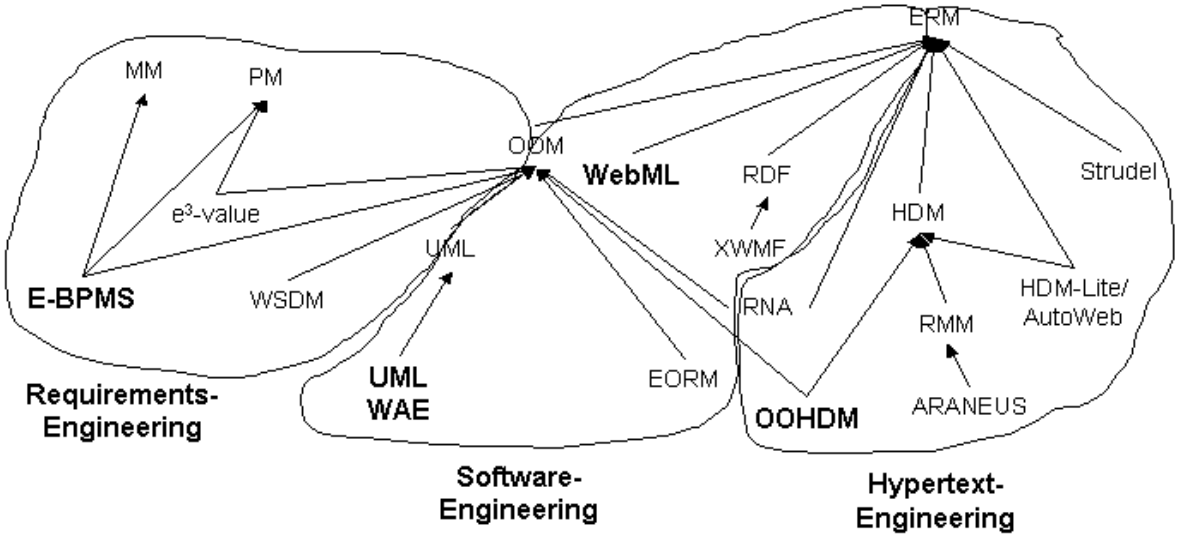
The interoperability between organisation is a major issue in current web-application development. The rapid changing web-infrastructure enables organisation, public administration and customers to interact, provide or use web-applications [2]. The complexity of such applications is a key topic beside the communication over standards between cross-organisational application. Heterogeneous technologies, different business processes, different roles and participants as well as fast development cycles require a structured and analytical support of web-application development. Model-driven approach helps to analyse the requirements, structure the key elements and support an analytical procedure. Such models are designed using a modelling method that consists of modelling elements and rules. Therefore a web-modelling method consists of modelling elements and rules that are used to describe web-applications. The model of the modelling method is commonly used as the "meta-model". In [3] describes requirements for web-modelling methods.

The next section introduces briefly some modelling approaches for web-applications. This is followed by an introduction of criteria for a classification of these approaches to finally classify the listed web-modelling methods. In the following the languages OOHMD (Object-Oriented Hypermedia Design Methodology), WebXML (Web Modelling Language), UML WAE (UML Web Application Extension) and E-BPMS (E-Business Process Management Systems) are discussed.

C.3. Overview on web-modelling methods

Currently there is a set of modelling methods available to design web-applications that partly evolved out of the scientific domain or from industrial developments. In the following an overview of web-modelling approaches is depicted distinguishing between "requirement engineering", "software engineering" and "hypertext engineering".

[5] The Excurse has not been developed during WP 5.4 but is mentioned to provide background information on the topic.



Legend:
MM: Meta modeling, PM: Process modeling, OOM: Object-oriented modeling, UML: Unified Modeling Language, ERM: Entity-Relationship-Modelling, RDF: Resource Description Framework

Figure 31 Overview of existing web-modelling approaches

Garzotto et. al. was the first who introduced the Hypertext Design Model (HDM) in 1991 and refined it later to HDM2 [4]. HDM is a two-step modelling approach, where the first step is a conceptual schema specifying entities and relations to problem areas. The second step is a navigation schema using views on the conceptual schema.

Lange introduced the Enhances Object Relationship Model (EORM) [5] that extended the Object Modelling Technique (OMT) with the concept of "link classes".

Isakowitz et. al. published the Relationship Management Methodology (RMM) [6] and introduced a structured and formal approach to design relations in HDM.

Fernandez introduced STRUDEL Web-Site Management Systems [7] as a development environment for complex web-sites.

Fraternali and Paolini published HDM-Lite [8] that is based on ERM and HDM introducing three modeling layers the structure schema, the navigation schema and the presentation schema.

De Troyer and Leune described the Web Site Design Method (WEDM) that is a user-orientated design method for web-applications [9].

Atzeni et al. developed the ARANEUS methodology [10] that distinguishes between six phases. The conceptual database design, the logic database design, the conceptual Hypertext design, the logic Hypertext design, the presentation design and the view design.

Klapsing and Neuman published the eXtensible Web Modelling Framework (XWMF) [11] that is a formal description of web-applications. XWMF is based on RDF.

Gordjin et al. introduced the e^3-VALUE [12]framework that focus on requirement definition for web-applications. There are three abstraction layers, the business model, the business process model and the underlying software architecture.

Yoo and Bieber published a method and domain and a method independent methodology called Relationship Navigation Analysis (RNA) [13] to define and analyse relations. There are five steps to analyse a Web-Application: Stackholder Analysis, Element Analysis, Relation Analysis, Navigation Analysis and Implementation Feasibility Analysis.

C.4. Web-modelling methods: classification criteria

Seven criteria are used for the classification of the web-modelling methods, each of them weighted in three steps. The symbol "+" means that the criteria is fully filled, the symbol "O" identifies a criterion partly filled and the symbol "-" points out that the criterion is not filled.

Criterion 1 - Roles and Permissions:

A web-modelling method should identify different user groups, profiles and different navigations. The features, the navigation and the content are based on the users profile. The web-modelling method should therefore enable the specification of profiles, roles and permissions.

Criterion 2 - Navigation:

A difference between web-applications and traditional applications are the navigation capabilities. A navigation concept should be part of the design from a web-modelling method.

Criterion 3 – Operation and Application Logic:

Web-application often integrates flexible and dynamic parts like services, scripts or remote program calls. A web-application modelling method has to provide constructs to describe these dynamic components.

Criterion 4 – Business Process and Workflow:

User requirements that are based on the business process or the executing workflow have to be linked to the actual web-application design. Therefore a linkage between the requirements of the business process and the IT-solution is required for web-application modelling methods.

Criterion 5 – The Size of the Application:

Web-applications are often divided based on their size. Web-modelling methods have to consider the integration of web-services, components and integrated organisations.

Criterion 6 – The Architecture of the Application:

Scalability and maintainability are strongly influenced by the size of the application. A web-modelling method has to consider not only the size but also the requirements on the architecture in terms of presentation, navigation, application logic, and data.

Criterion 7 – The Methodology:

The design of a complex web-application can be guided and supported by a methodology that provides modelling constructs for different phases of the methodology.

C.5. Comparison of web-modelling methods

OOHDM

The focus of OOHDM [14] is the navigation structure. There are four phases "Conceptual Design", "Navigational Design", "Abstract Interface Design" and "Implementation". These phases represent the logical modelling layers of OOHDM.

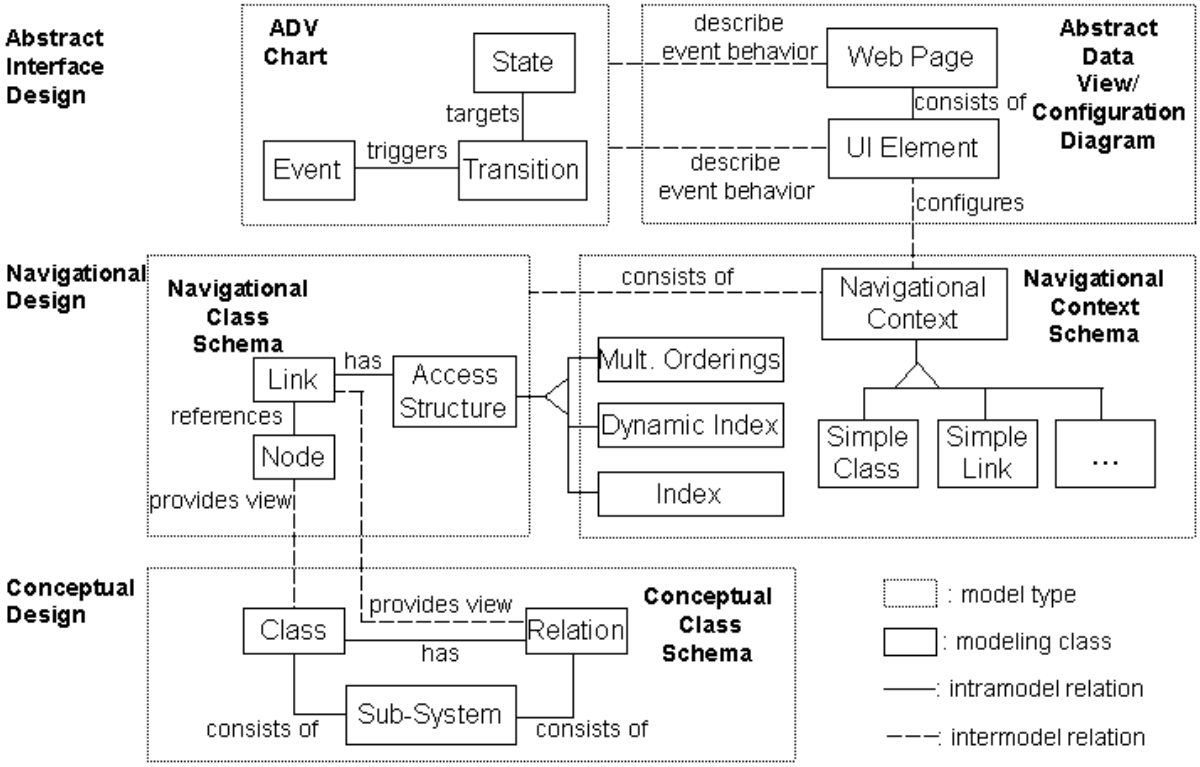


Figure 32 Meta model of OOHDM

The conceptual design describes the problem area and is seen as a conceptual data design consisting of sub-systems, relations and classes.

The navigational design defines views on the conceptual design that are interpreted as navigation possibilities. There are constructs such as links, nodes and access structures available.

The navigational context schema describes the accessible and visible navigation objects for the user. The abstract interface design describes, how the navigation objects are represented for the user.

WebML

WebML [15], [16] consists of four perspectives that are "structural model", "hypertext model", "presentation model" and "personalised model". Besides the different perspectives there is also a suggested methodology that covers the phases "Requirements Collection", "Data Design", "Hypertext Design In-The-Large", "Hypertext Design In-The-Small", "Presentation Design", "User and Group Design" and "Customisation Design".

The structural model describes the logic data model for web-application consisting of entities and relations. The hypertext model is described using two views. The first view is the composition model where individual pages (units) are defined. The navigational model describes the relation between the different sites (units). The presentation model is a target platform for independent layout for sites, whereas the presentation model describes users and user groups of a web-application with their views on the structural model.

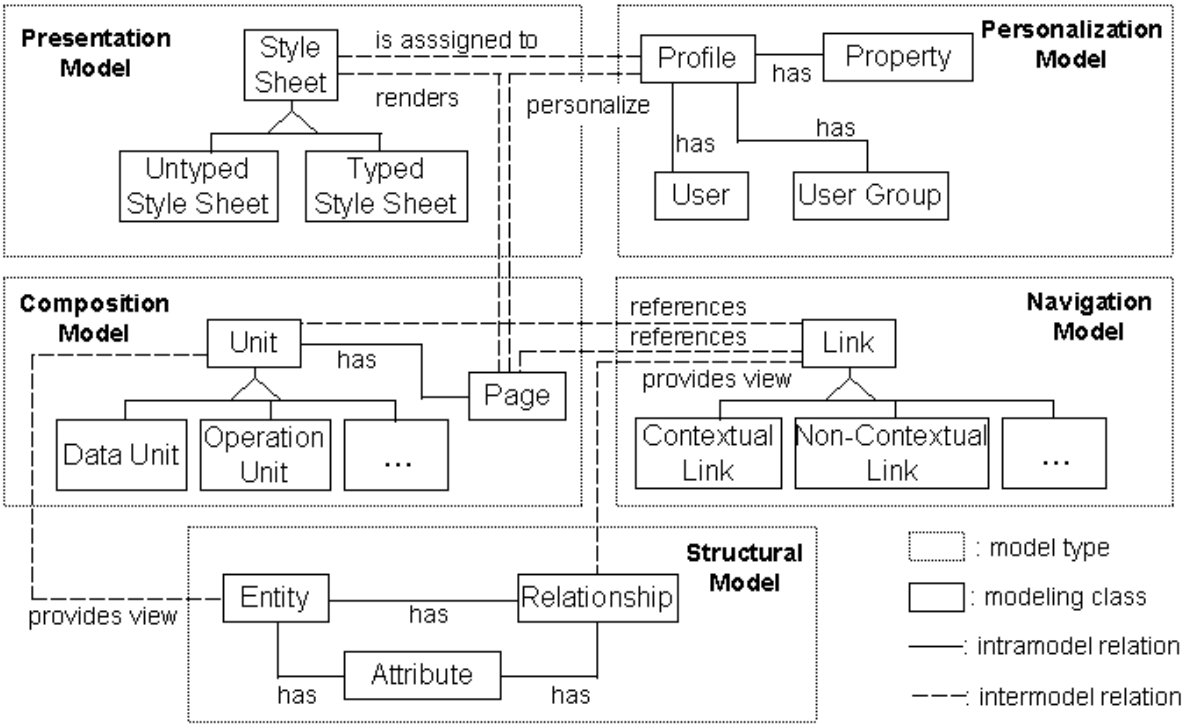


Figure 33 WebML Metamodel

UML Web Application Extention

UML was first presented as a standardised modelling method for object oriented specifications in 1997, since then it has been continuously updated with extensions and is now widely accepted and used for software design and requirement analysis. This paragraph explains the special extension that has been added to UML in 1999 for web-application [17], [18] and [19]. The use of UML in Akogrimo is mentioned in chapter 4 Methods for the Akogrimo Roadmap.

Because of the Web Application extension there are new constructs like "stereotypes", "tagged values" and "object constraints languages" that are specially designed to model web-applications.

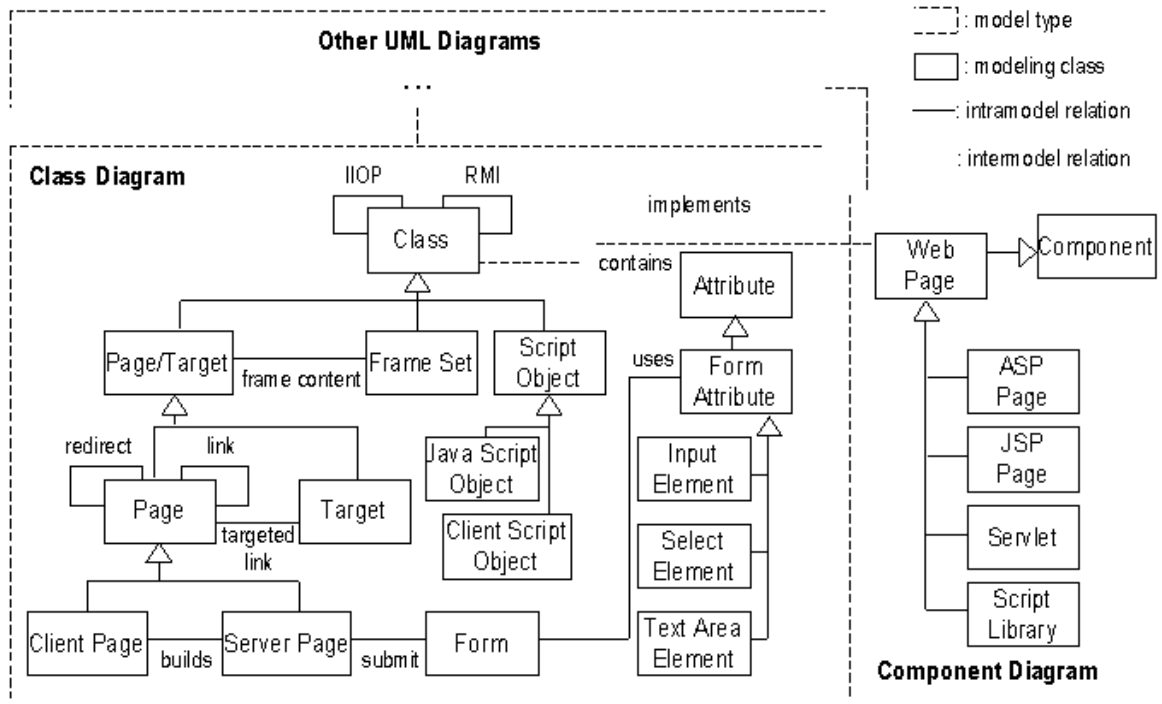


Figure 34 UML WAE Metamodel

The model types "Class Diagram" and "Component Diagram" have been adapted for the requirements of web applications. The basic elements of class, association, attribute and component have been changed to include the definition of server pages, client pages, forms, frames, target, script objects, targeted links, frame content, submits, redirect, Internet Inter-ORB (IIOP), Remote Method Invocation (RMI), input elements, web pages, ASP pages, JSP, Servlets or script libraries.

E-BPMS

The E-BPMS [2] framework is used for supporting business processes mentioned in chapter 2 and is the basis for the adapted Akogrimo Business Process Management framework (AKBPM) that will be used for the domain specific application adaptation roadmap. This section discusses the E-BPMS framework in comparison to the previously introduced frameworks.

The main focus of the E-BPMS is to design the business model, and the related web-applications. The methodology of E-BPMS is based on the "Strategic Decision Process", "Re-Engineering Process", "Resource-Allocation Process", "Workflow Process" and "Performance Evaluation Process". Based on these processes the model types "business model", "product model", "process map", "business process model", "organisational model", "interaction process model", "information system model" and "it-infrastructure model" have been derived.

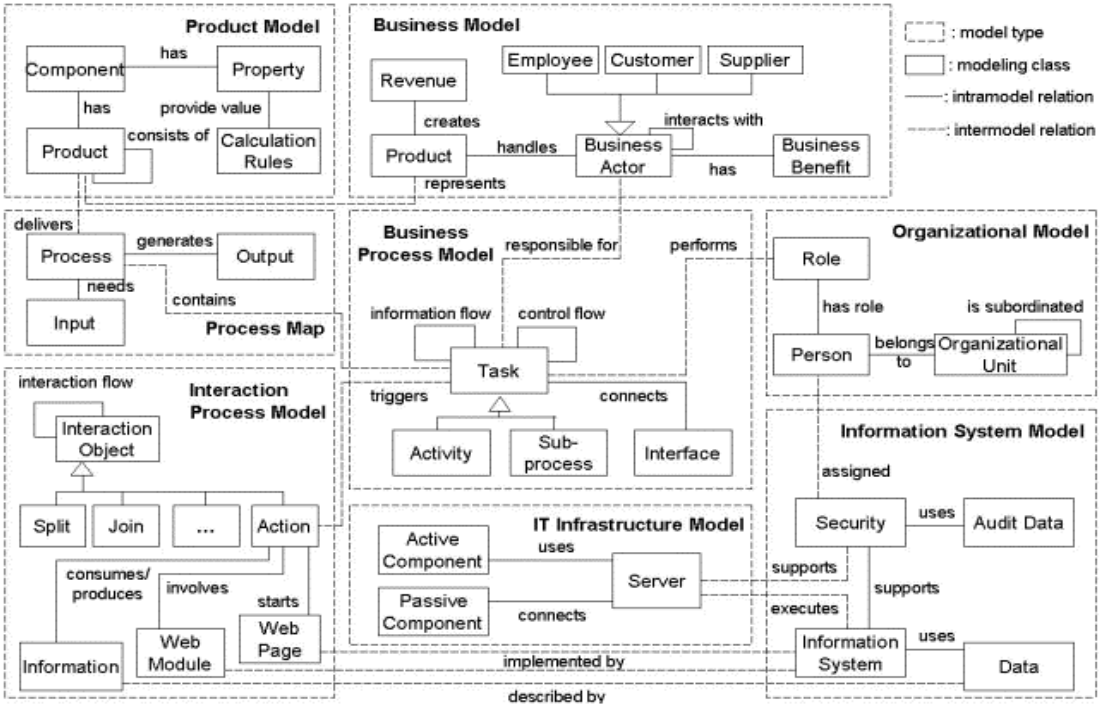


Figure 35 E-BPMS Metamodel

C.6. Classification of web-modelling methods

Here the main criteria that have been introduced to classify modelling methods for web-applications are applied to compare the four web-modelling methods OOHDM, WebML, UML WAE and E-BPMS.

	OOHDM	WebML	UMLWAE	E-BPMS
C1 - Roles and Permissions	O	+	O	+
C2 - Navigation	+	+	+	+
C3 – Operation and Application Logic	O	+	+	+
C4 – Business Process and Workflow	-	-	O	+
C5 – The Size of the Application	O	+	+	+
C6 – The Architecture of the Application	O	+	+	+
C7 – The Methodology	O	O	+	O

Table 12 A comparison between the Web-modelling methods with the previously described rating

C.7. References of Annex C

[1] Kühn H., Modellierungssprachen für Web-Anwendungen: Klassifikation und Vergleich, Working Conference eBP 2001 – „Elektronische Geschäftsprozesse“, University of Klagenfurt, 24 and 25 September 2001, P. Horster (Ed.), it Verlag pp. 379 - 390

[2] Bayer F., Junginger S., Kühn H., A Business Process-oriented Methodology for Developing E-Business Applications. In: Proceedings of the 7th European Concurrent Engineering Converence (ECEC 2000), Society for Computer Simulation (SCS), April 2000, p 32-40

[3] Retschitzegger W., Schwinger W., Towards Modeling of DataWeb applications – A Requirments' Perspective. In: Proceedings of the Americas Conference on Information Systems, AMCIS 2000, Long Beach California, August 2000, Vol. I., p149 – 155

[4] Garzotto F., Paolini P., Schwabe D., HDM – A Model for the Design of Hypertext Applications. In: Proceedings of the ACM Hypertext '91 Conference, ACM Press, 1991, p 313-328

[5] Lange D., An Object-Oriented Design Method for Hypermedia Information Systems. In: Proceedings of the 27th Hawaii International Conference on System Science, 1994. p.366-375

[6] Isakowitz T., Stohr E. A., Balasubramanian P.: RMM: A Methodology for Structured Hypermedia Design. In: Communications of the ACM, 38 (8), 1995, p 26-29

[7] Fernandez M., Florescu D., Kang J., Levy A., Suciu D., STRUDEL : A Web-site Management System. In : Proceedings of the ACM SIGMOD Conference, 1997, p 549-552.

[8] Fraternali P., Paolini P,: A Conceptual Model and a Tool Environment for Developing More Scalable, Dynamic and Customizable Web Applications. In: Proceedings of International Conference on Extending Database Technology (EDBT98), 1998, p 421-438

[9] De Troyer O., Leune C. J., WSDM : A User Centered Design Method for Web Sites. In: Proceedings of the WWW7 Conference, 1998, p85-94

[10] Atzeni P., Mecca G., Merialdo P., Design and Maintenance of Data-Intesive Web-Sites. In: Proceedings of the International Conference on Extending Database Technology (EDBT'98), 1998, p436-450

[11] Klapsing R., Neumann G., An eXtensible Web Modeling Framework. In: Proceedings of the 8th World Wide Web Conference. Poster Session 1999

[12] Gordijn J., Akkermans, H., van Vliet H., Requirements for e-commerce applications are created rather than elicited. In: Proceedings of the Second Nordic Workshop on Software Architecture (NOSA'99), 1999.

[13] Yoo S., Bieber M., Finding Linking Opportunities through Relationship-based Analysis. In: Proceedings of the International ACM Hypertext 2000 Conference, 2000, p 181-190

[14] Schwabe D., Rossi D., Barbosa S.D.J., Systematic Hypermedia Application Design with OOHDM. In: Proceedings of the Conference Hypertext '96, 1996, p116-128

[15] Ceri S., Fraternali P., Maurino A., Paraboschi S., One-to-One Personalization of Data-Intensive Web-Sites. In: Proceedings of the WEBDB workshop 1999 p.1-6

[16] Ceri S., Fraternali P., Bangio A., Web Modeling Language (WebML): a modeling language for designino Web sites. In Proceedingsof the WWW9 conference, 2000, p137-157

[17] OMG – Object Management Group: OMG Unified Modeling Language Specification.[6]

[18] Hitz M., Kappel G., UML@Work. Dpunkt-Verlag, Heidelbert 1999

[19] Conallen J., Building Web Applications with UML. Addison Wesley, Longman Inc. 2000

[6] The Web Applicatin Extention was initially defined on top of UML 1.3

D. Akogrimo Business Modelling Method

Corresponding to the requirements defined in D3.2.2 a Business Modelling Method was developed for Akogrimo. In the following all model types of this method are introduced. For each necessary model type an example of the use within the Akogrimo project is depicted.

D.1. The Business Model

This strategic model defines the business participants and the information as well as product and cash flows between them. The relevant products and processes are identified and described on a more detailed level in other model types.

The Business Model consists of the following elements:

Element	Icon	Description
Business actor		The "Business actor" describes one specific actor of your Business Model. The actor can be classified, e.g. as a Company, Department, Customer, Service Provider
BMProduct		The "BMProduct" is a Product, which is generated through your Business Model. The product corresponds to a good or a service.
Organisation		The "Organisation" is an aggregation. It supports the logical structure of the model content. Objects placed in the same Organisation belong to the same Organisation.
BMProcess		The "BMProcess" is a process, which was identified in your Business Model. The corresponding process can be modelled in detail and referenced.
Slot		The "Slot" is a communication point between the objects. A connector number can be added to the Slot.

Table 13 Notation of the Business Model in AKBPM

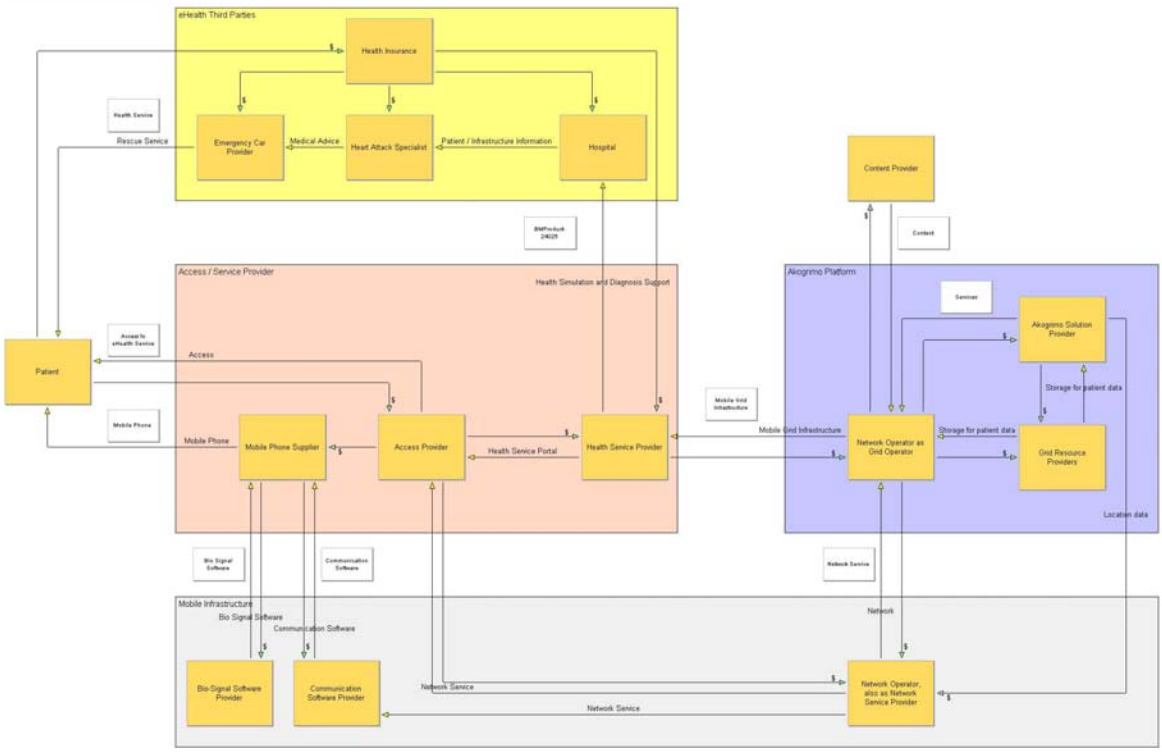


Figure 36 Business Model for the Akogrimo eHealth Scenario

D.2. The Company Map

The Company Map is an overview model that identifies all processes within a business scenario. This includes Business Processes and EPC Processes.

The Company Map consists of the following elements:

Element	Icon	Description
Process		A „Process“ is a reference on a Company Map to a business process model or an EPC Process model.
Group		The “Group” supports the logical structure of model contents on the drawing area.
Note		The „Note“ allows free text to be placed within a model.

Table 14 Notation of the Company Map in AKBPM

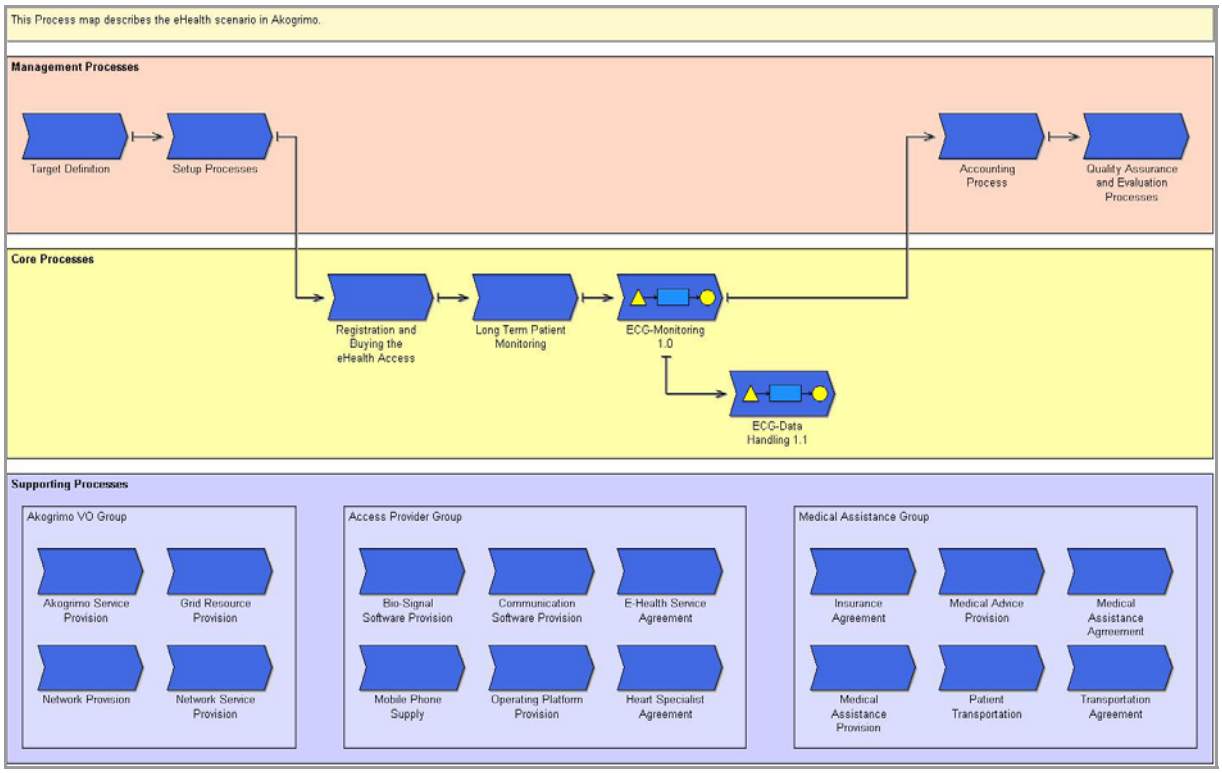


Figure 37 Sample Company Map

D.3. The Business Process Model

For every relevant Business Process for each business, an actor is identified and specified in the Business Process Model. The Business Process Model includes all relevant activities, as well as performers and resources.

The Business Process Model consists of the following elements:

Element	Icon	Description
Process start		The "Process start" symbolises the beginning of a business process. Each business process model has to contain exactly one Process start object. The object „Process start" has no predecessor and only one successor and can be set with variables.
Subprocess		The "Subprocess" is used to call other processes. This makes sense especially when a particular process is carried out in several areas of the business. Rather than repeating the same process within a larger model a number of times, it makes sense to create a subprocess, which can be called wherever necessary. The Sub-process object can also be useful when you want to structure your business processes to maximise clarity. It must have at least one predecessor and at most one Successor.
Activity		"Activities" describe which tasks in a business process model have to be executed. The object "Activity" has at least one predecessor and at most one Successor. The level of granularity at which activities are described can vary according to the

		requirements of the modeller. Workflow models, which are triggered, can be referenced. Furthermore references to supporting documents or services can be added.
Decision		The "Decision" within a business process model allows pre-defined variables to be queried in order to determine the path through the process. A decision has at least one predecessor and at least one Successor. During a simulation only one of the Successors must evaluate to "true".
Parallelism		The "Parallelism" makes it possible for several paths in a business process to be executed at the same time. Every parallelism ends with a union (Merging object). The object parallelism has at least one predecessor and one successor. A transition condition can be set on a successor relation following a parallelism object. This means that one (or more) of the "parallel" paths may be followed only in certain conditions. The object "Parallelism" should have the same number of Successors as the corresponding "Merging" object has predecessors.
Merging		With the "Merging" object, parallel paths are re-joined. The object "Merging" has as many predecessors as the corresponding "Parallelism" object has successors and it has at most one successor.
End		The object „End" marks the end of a path of a business process. Several objects of the class "End" are allowed in one model. The object "End" has at least one predecessor and no successor.
Group		The "Group" supports the logical structure of model contents on the drawing area.
Note		The "Note" allows free text to be placed within a model.
Variable		Variables are used to enable transition conditions to be defined in the model. These transition conditions are set in the "Successor" relation (after a Decision or Parallelism). The object "Variable" has no predecessor or successor. The object „Variable" can be related to the object „Random generator" with the relation "Sets variable". One variable can be set with different values at different times through the use of several "Random generators".
Random generator		The Random generator sets a variable with a certain value. The value can be set through a steady or a discrete distribution function. The object „Random generator" has no predecessor or Successor. The object „Random generator" can be related to the following objects:

		"Variable" through the relation „Sets variable" "Process start", "Activity" through the relation "Sets". It can also be related to other objects such as "Decision", "Parallelism" etc. through the relation "Sets". Several Random generators can set the same "Variable" object with values.
Resource		The "Resource" represents an EDP system, a communication system or any other resource, which is necessary for executing the business process.

Table 15 Notation of the Business Process Model in AKBPM

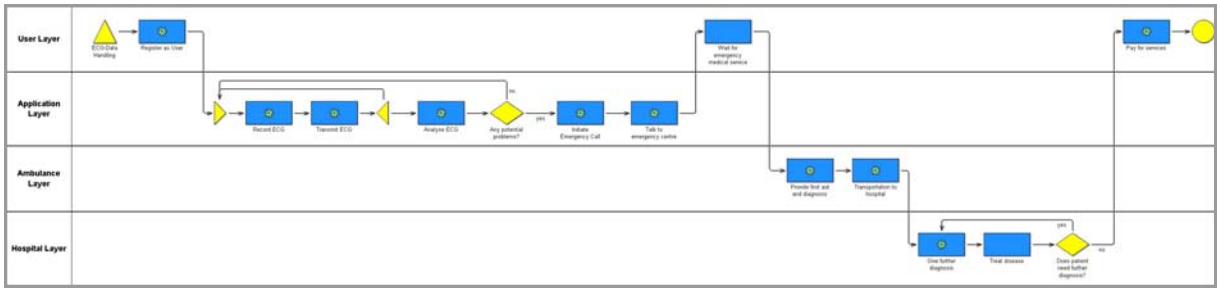


Figure 38 Sample Business Process Model

D.4. The EPC Process Model

With the use of an EPC Process Model a more technical view on the process can be specified. This model type allows the import of existing models from different modelling tools.

The EPC Process Model consists of the following elements:

Element	Icon	Description
Process start - EPC		The "Process start – EPC" symbolises the start of an EPC process. The object „Process start - EPC" has no predecessor and only one successor.
Event		The "Event" is used to trigger a function. In addition each function creates an Event
Function		The "Function" is the lowest task, which the system is able to execute. The "Function" transforms Input- to Output-Data
Process path indicator		The "Process path indicator" represents a link with another model of the same type.
End - EPC		The object „End - EPC" marks the end of a path of an EPC process. Several objects of the class „End - EPC" are allowed in one model. The object „End - EPC" has at least one predecessor and no successor.

XOR operator		The "XOR operator" represents a one or the other connection
OR		The "OR" represents an or-connection
AND		The "AND" represents an and-connection
Merging - EPC		With the „Merging - EPC" object, parallel paths are re-joined. The object „Merging - EPC" has as many predecessors as the corresponding object successors, and it has at most one successor.
Organisational unit		The "Organisational unit" allows a reference to an organisational unit in the corresponding EPC-Pool.
Location		The "Location" allows a reference to a location in the corresponding EPC-Pool.
Position		The "Position" allows a reference to a position in the corresponding EPC-Pool.
EPC-Resource		The "EPC - Resource" represents an EDP system, a communication system or any other resource, which is necessary for executing the business process.
ASAP documentation		The "ASAP documentation" allows the linking of documents.
EPC Subprocess		The "EPC Subprocess" is used to call other processes. This makes sense especially when a particular process is carried out in several areas of the business. Rather than repeating the same process within a larger model a number of times, it makes sense to create a subprocess, which can be called wherever necessary. The EPC Subprocess object can also be useful when you want to structure your processes to maximize clarity. It must have at least one predecessor and at most one Successor.
Organisational unit type		The "Organisational unit type" is part of the eEPC method. It is used to describe the type of an organisational unit. Allows a reference to an organisational unit type in the EPC-Pool Model.
Information carrier		The "Information carrier" is part of the eEPC method and it is used to model e.g. documents or files, which are necessary for execution of a function. It allows a reference to an information carrier in the EPC-Pool Model.
Person		The Person describes an individual person. Allows a reference to person in the EPC-Pool Model.
Person type		The "Person type" describes the kind of person, which is involved in the execution of the function. Allows a reference to person type in the EPC-Pool Model.

General resource		The "General resource" describes a resource, which is necessary for the execution of the function. Allows a reference to general resource in the EPC-Pool Model.
Operational resource		The "Operational resource" describes a resource, which is necessary for the execution of the function. Allows a reference to an operational resource in the EPC-Pool Model.
Hardware component		The "Hardware component" describes a resource, which is necessary for the execution of the function. Allows a reference to a hardware component in the EPC-Pool Model.
Application system		The "Application system" describes a system, which is necessary for the execution of the function. Allows a reference to an application system in the EPC-Pool Model.
Application system type		The "Application system type" describes the type of system, which is necessary for the execution of the function. Allows a reference to an application system type in the EPC-Pool Model.
Aggregation		The "Aggregation" supports the logical structure of model contents on the drawing area.
Note		The "Note" allows free text to be placed within a model.

Table 16 Notation of EPC in AKBPM

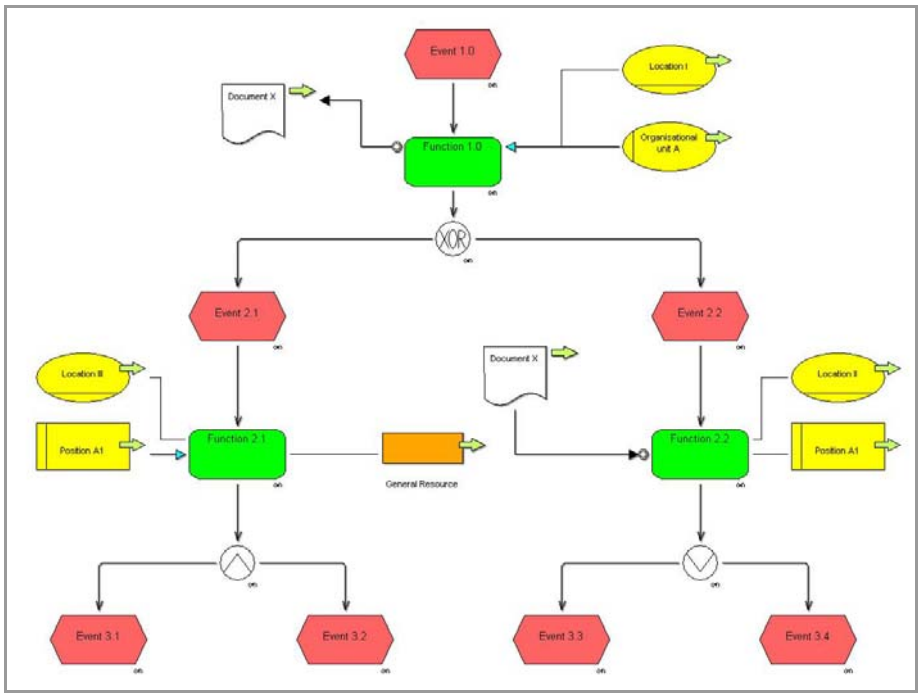


Figure 39 Example EPC Process Model

D.5. The Working Environment Model

The Working Environment Model describes the organisational environment such as organisational units, performers and roles.

The Working Environment Model consists of the following elements:

Element	Icon	Description
Organisational unit (WE)		Definition of organisational units within a working environment.
Performer (WE)		Definition of a performer who belongs to an organisational unit.
Role (WE)		The Role describes the task-range or job description of a performer. Every performer can have one or more roles. Several performers can have the same role.
Resource		The "Resource" represents an EDP system, a communication system or any other resource, which is necessary for executing the business process.
Group		The "Group" supports the logical structure of model contents on the drawing area.
Note		The "Note" allows free text to be placed within a model.

Table 17 Notation of the Working Environment Model in AKBPM

D.6. The Organigram Model

Like the Working Environment Model the Organigram Model describes the organisational environment such as organisational units, performers and roles. Like the Working Environment Model belongs to the Business Process Model, the Organigram Model belongs to the EPC Process Model

The Organigram Model consists of the following elements:

Element	Icon	Description
Organisational unit		Definition of organisational units.
Performer		Definition of a performer who belongs to an organisational unit.
Role		The Role describes the task-range or job description of a performer. Every performer can have one or more roles. Several performers can have the same role.
Resource		The "Resource" represents an EDP system, a communication system or any other resource, which is necessary for executing the processes.
Group		The "Group" supports the logical structure of model contents on the drawing area.
Note		The "Note" allows free text to be placed within a model.

Table 18 Notation of the Organigram Model

D.7. The Use Case Diagram

The Use Case Diagram is a specification of requirements of a so-called business case. This model type became popular through the Unified Model Language Specification UML and is mainly used to define business requirements for a software system.

The Use Case Diagram consists of the following elements:

Element	Icon	Description
actor		An "actor" describes a role, which participants take concerning to the system. In modelling the system interface it is not important, which concrete persons make demands. All participants are divided into groups with the help of their demands. A role is assigned to every group. Different participants within a group who make the same demands, have the same role and are modelled with the help of a single symbol.
use case		A "use case" is the specification of a set of actions performed by a system, which yields an observable result, which is typically of value for one or more actors or other stakeholders of the

		system.
system boundary		With the help of the “system boundary” use cases that logically belong together can be grouped.
note		The „note“ allows free text to be placed within a model.

Table 19 Notation of the Use Case Model in AKBPM

D.8. The Workflow Model

The Workflow Model is used to model BPEL based business processes. This model also allows the export of the modelled processes. This allows the exported models to be imported into a BPEL engine.

The Workflow Model consists of the following elements:

Element	Icon	Description
Process start (BPEL)		The “Process start (BPEL)” symbolises the beginning of the workflow process. Each workflow model has to contain exactly one Process start object. The object „Process start (BPEL)“ has no predecessor and only one successor and it can be set with variables.
Subprocess (BPEL)		The “Subprocess (BPEL)” is used to call other processes. This makes sense especially when a particular process is carried out in several areas of the business. Rather than repeating the same process within a larger model a number of times, it makes sense to create a subprocess, which can be called wherever necessary. The Subprocess object can also be useful when you want to structure your workflow processes to maximise clarity. It must have at least one predecessor and at most one successor.
Activity (BPEL)		The “Activity (BPEL)” describes what tasks in a workflow model have to be executed. The object „Activity (BPEL)“ has at least one predecessor and at most one successor. The type of activity (receive, reply, invoke, assign, wait, empty, compensate) has to be specified. Furthermore partner links and variables can be referenced.
Decision (BPEL)		The “Decision (BPEL)” within a workflow model allows pre-defined variables to be queried in order to determine the path through the process. Here it is possible to choose between the <switch> or <pick> construct.
Decision (Merging)		The “Decision (Merging)” defines the end of <switch> or <pick> construct.
Variable (BPEL)		The “Variable (BPEL)” is used to define the data variables used by the process.

Partner link		The "Partner link" defines the different parties that interact with the business process. A partner link type and a role name characterise the partner link.
Parallelism		The "Parallelism" makes it possible for several paths in a Workflow model to be executed at the same time. Every parallelism ends with a union (Merging object). The object parallelism has at least one predecessor and one successor..
Merging		With the „Merging" object, parallel paths are re-joined.
End		The object „End" marks the end of a path of the Workflow model. The object „End" has at least one predecessor and no successor.
Group		The "Group" supports the logical structure of model contents on the drawing area.
Note		The "Note" allows free text to be placed within a model.

Table 20 Notation of the Workflow Model in AKBPM

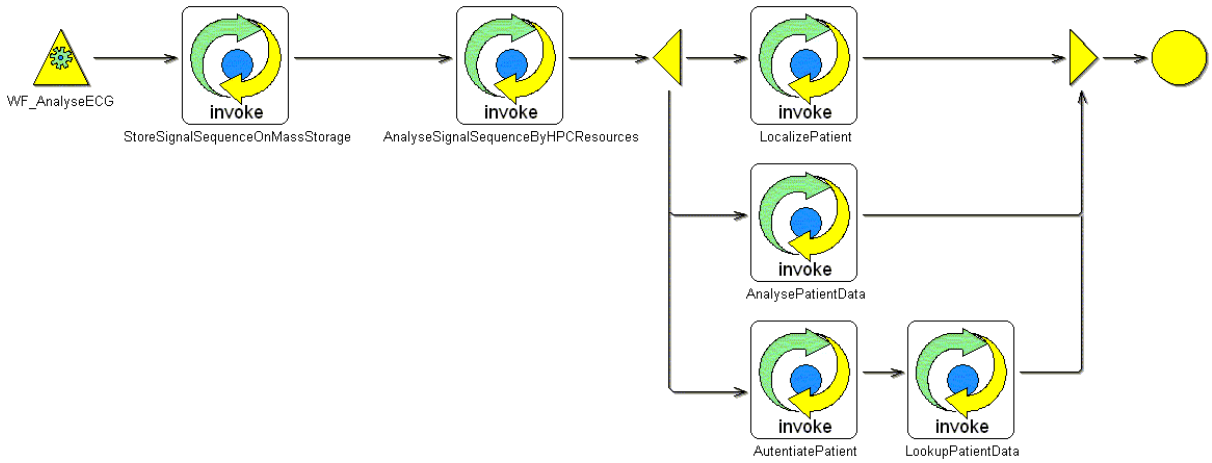


Figure 40 Example Workflow Process for the Akogrimo eHealth scenario

D.9. The Service-Pool

The Service-Pool is a pool model that includes all services, which are necessary for the execution of the processes (EPC processes, Business processes).

The Service-Pool consists of the following elements:

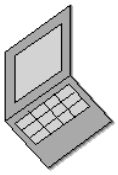
Element	Icon	Description
Service		The "Service" describes a service in detail. Among other things the goals, the characteristics, the service receiver, the service provider, the required resources, the service level and also dependencies, interfaces and success factors can be defined.
Group		The "Group" supports the logical structure of model contents on the drawing area.
Note		The "Note" allows free text to be placed within a model.

Table 21 Notation of the Service Pool

D.10. The Information-Pool

The Information-Pool is a pool model that includes all documents, which support the execution of the processes (EPC processes, Business processes).

The Information-Pool consists of the following elements:

Element	Icon	Description
Document		The "Document" corresponds to a document, which supports the processes. A reference to the document can be entered. The document can be linked with an activity or function
Group		The "Group" supports the logical structure of model contents on the drawing area.
Note		The "Note" allows free text to be placed within a model.

Table 22 Notation of the Information Pool

D.11. The EPC-Pool

The EPC-Pool is a pool model, which contains several elements of the EPC process model. It is an overview of the referenced elements and is necessary for EPC import.

The EPC-Pool consists of the following elements:

Element	Icon	Description
Organisational unit (pool)		The "Organisational unit" describes an organisational unit
Location (pool)		The "Location" describes a location.
Position (pool)		The "Position" describes a position.
Organisational unit type (pool)		The "Organisational unit type" is part of the eEPC method. It is used to describe the type of an organisational unit.
Information carrier (pool)		The "Information carrier" is part of the eEPC method and it is used to model e.g. documents or files, which are necessary for execution of a function.
Person (pool)		The Person describes an individual person.
Person type (pool)		The "Person type" describes the kind of person, which is involved in the execution of the function.
General resource (pool)		The "General resource" describes a resource, which is necessary for the execution of the function.
Operational resource (pool)		The "Operational resource" describes a resource, which is necessary for the execution of the function.
Hardware component (pool)		The "Hardware component" describes a resource, which is necessary for the execution of the function.
Application system (pool)		The "Application system" describes a system, which is necessary for the execution of the function.
Application system type (pool)		The "Application system type" describes the type of system, which is necessary for the execution of the function.

Table 23 Notation of the EPC Pool in AKBPM

E. Annex: Excurse[7] on Process-Oriented Knowledge Modelling Methods

E.1. Abstract

This excurse discusses some basics of process-oriented knowledge modelling methods that is used for the service roadmap and is seen as a separate chapter of this deliverable. The aim is to discuss currently available process-oriented knowledge modelling methods in more detail than in chapter 4.4. The chapter focuses on roadmaps in Akogrimo and only briefly presents the method analysis. This excurse provides some details on alternative methods and describes why the method PROMOTE® has been selected and how it is designed. The original text has been published in [1].

E.2. Introduction

Business Process Oriented KM (BPOKM) is a possible entry point into knowledge management (KM). The enhancement of business processes (BPs) is a core goal in KM (compare [2], [3], or [4]).

Practical KM however is still not manifested as a consequent follow-up of Business Process Re-engineering projects. This conflict is also evident vice versa, as KM-projects are seldom based on BPs.

Business Processes and KM still suffer from heterogeneous integration. Such integration is established by the BPOKM approach. There are three different meanings mentioned for BPOKM (compare [5]).

Within Akogrimo the development of a service how it is mentioned in chapter 5 is seen as a so-called core business process. A business process consisting of a set of development activities with the final goal being to implement a service into the Akogrimo middleware. In the field of knowledge management there are three different types of process-oriented knowledge management.

Business Process as knowledge. The first meaning of BPOKM is to define the Business Process as knowledge items. This means that a BP-Model is seen as a document that provides knowledge. The management and the distribution of BP-Models are defined as KM. Therefore the acquisition, the analysis, the simulation and distribution of BPs are related to core KM activities (compare [3]). Today's research aims to distribute BP-Models in a more flexible way using Internet access for modelling tools and dynamic model allocation depending on the knowledge role of the user. The storage of large BP-Models is executed using knowledge bases and the distribution is realised by knowledge dissemination strategies.

Business Process as an entry point and integration platform. The second meaning of BPOKM defines the BP as the starting point. The business process is used to define a process-oriented functional specification. This is realised by analysing each activity and identifying the so-called "Knowledge Intensive Tasks" (KIT). The BP is therefore the entry point for a more detailed specification of the knowledge platform. All requirements of an E-KMS are therefore directly related to the needs of the BP. This leads to a BP-centred architecture that interprets the process as an integration platform.

[7] The Excurse has not been developed during WP 5.4 but is mentioned to provide background information on the topic.

KM activities as processes. The third interpretation of BPOKM is to define KM activities as processes. In today's literature these processes are named differently as either "Knowledge Processes" or "Knowledge Management Processes" and define a sequence of KM activities. In this text the term "Knowledge Management Process" (KMP) is used, as the term "Knowledge Process" has a different meaning. This interpretation enables us to use management concepts like steering, controlling, and evaluation of KM. A research effort in this topic is e.g. to automate knowledge management processes like the distribution of a best practice article.

These three interpretations are seen as an evolutionary development of BPOKM; first, starting with defining, managing, and distributing the core processes; second, enriching these core processes with KM to make them more efficient; and finally third, manage the KM to make KM more efficient. This evolutionary approach implies that before process-oriented KM can be realised on the third level, level one and two need to be implemented sufficiently (compare [6]).

Based on the evolutionary view the first step is seen as important for the Akogrimo roadmaps. This process describes how the partners implemented their services. The experience of the partners is then made explicit by modelling the different activities and the core decisions that have been made.

This section introduces a layer concept that defines the BP as the top layer and the actual Knowledge Resource (KR), like documents, links or the BSCW access as the bottom layer.

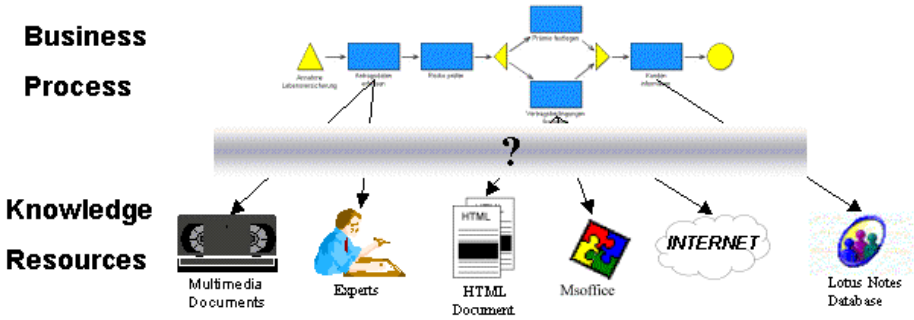


Figure 21 BP related to KR

Figure 21 depicts a BP as a starting point that enables the definition of knowledge resources (KR) references.

During the EU-project PROMOTE a concept has been developed to link KR to BPs using models. The Organisational Memory in this context is seen as a container of all KR that can be accessed by input, output, and maintenance KMPs.

In the following an introduction in knowledge models is discussed.

E.3. Modelling the Organisational Memory

The term "Organisational Memory" (compare [7]) is defined by (Hedberg) as "[…] the cognitive structure of information proceeding processes of the whole organisation, the Theory of Action" (translated from German) [8]. This section depicts model languages that describe the Organisation Memory (OM) and gives an overview of modelling methods.

The focus of this section is the description of three approaches that are either well-known in BPM (ARIS and ADONIS) or have been run under the EU to research in the field of BPOKM (DÉCOR and PROMOTE). The other approaches are mentioned to provide a better overview.

PROMOTE

After a requirement analysis of the PROMOTE® application scenarios, a model language was defined that is sufficient to describe knowledge management approaches in a method and tool independent way [9], [10], [11], [12] or [4].

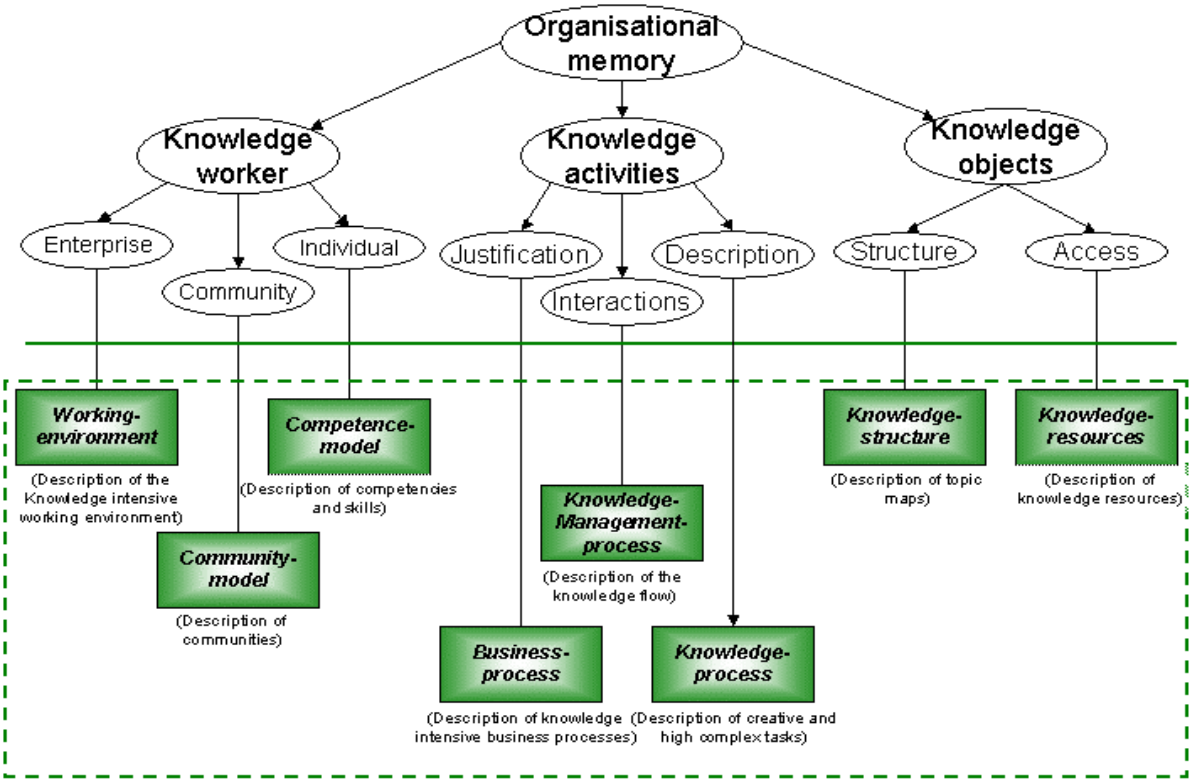


Figure 22 Overview of the PROMOTE model language

Figure 22 depicts the PROMOTE® model language that is used to describe the OM and the knowledge interactions between knowledge workers. This model language is based on the grammar of the natural language using "subject", "predicate", and "object" whereas the subjects are the knowledge workers, the predicates are the knowledge activities and the objects are the knowledge objects.

The "knowledge workers" are described on an individual level using skill profiles (skill model), on a community level to describe communities of practice (community model), and on an enterprise level to describe the competence profile of departments (working environment).

The already mentioned "knowledge activities" are justified by business processes, the interaction with the OM is defined by using the previously mentioned "Knowledge Management Processes" and knowledge intensive tasks (that have been defined as critical due to the analysis of the business process) are described in detail using a special type of process a so-called "Knowledge Process".

The "knowledge objects" are categorised by using "Knowledge Structure Models" and accessed by using a modelled index, the "Knowledge Resource Model".

DÉCOR

DÉCOR is an EC-funded project in the 5th framework, concerned with business process-oriented KM [13], [14]. The main goal is to develop methods and tools to build an intelligent information assistant helping users to perform knowledge intensive business tasks.

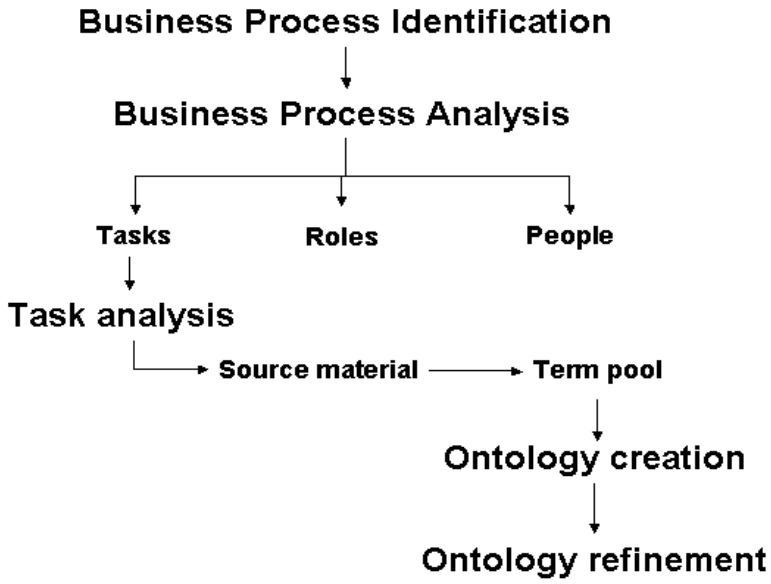


Figure 23: Overview of Ontology Development Method

Figure 23 depicts the five steps of the DÉCOR Ontology development method. This "DÉCOR Business Knowledge Method" is the first out of six modules from the DÉCOR toolbox. The second module of DÉCOR is the "Business Knowledge Modelling Tool" MS-Visio®. The modelling method uses two model types. The first is the "Knowledge enhanced business process modelling" for modelling elements like "tasks", "KM-tasks", "events", "connectors", "objects", "roles", or "persons. The second model type "Ontology modelling" is used for modelling elements like "kinds" (mapped to "structure units"), "Characteristics" (mapped to "definable attributes") and "Relations" (mapped to links). The tool CognoVision® interprets this Ontology.

Module 5 of DÉCOR is concerned with "weakly structured workflow models". The idea is to model weakly structured workflows and to export these models into a Workflow Management System. These weakly structured workflows support knowledge intensive Business tasks.

ARIS for Hyperwave

Another modelling approach is the ARIS method where three additional model types are introduced to the existing business process models [8], [15].

ARIS exchanges their KM-models with the KM-Portal "Hyperwave®" [16], [17]. The starting point is defined by Business Process Models enhanced with elements like "Knowledge category" and "Explicit Knowledge". In addition to the Business Process Model there are three additional model types to describe the organisational memory.

The first model type is the "Knowledge landscape" that is used to describe skill profiles of experts. The second model type is the "Knowledge structure" that is used to describe the context of knowledge categories. The third model type is the "Knowledge generation and Knowledge usage" model that is used to describe the development of knowledge and the involved knowledge workers.

Income

Based on [18] the Income Process Designer represents knowledge resources as objects and generates a knowledge landscape for documentation. Knowledge resources are similar to PROMOTE resources and can also be distinguished between explicit or implicit resources by referencing to a KR or to persons in the organisational diagram. Processes defining the knowledge transfer are not covered in this modelling approach. An interesting aspect is the definition of the use of KRs.

Workware

Based on [18] the core idea of Workware is that incomplete processes are supported, since the implicit knowledge of the knowledge worker enable the completeness of the process during execution. Implicit knowledge is included via notes into the process model.

Process oriented KM in the context of Workware means, that BP-Model improvement considers the experience of the users that posted their comments into the model. Process-oriented knowledge is only concerned with the explicit knowledge within the BP-model.

EULE2

Based on [18] this knowledge-based system supports agents of the Swiss Life Insurance Company at complex cases. Similar to Workflows the problem solving steps are modelled, these cluster a complex problem into smaller parts. This approach interprets process oriented KM as depicted in step 3. As shown previously, step 3 is until now the last step in the process oriented KM and therefore only weakly supported. That is the reason why there is no distinction between explicit and implicit knowledge and the knowledge models are seen as tasks descriptions.

K-Modeler

The K-Modeler (Knowledge-Modeller) is a modelling method of knowledge intensive BPs that identifie systematic knowledge and information within a process [18].

The basic concept is a communication structure analysis that has been developed by [19] at the TU Berlin to analyse knowledge interaction. There are organisational and technical dimensions that are used on top of BPs Models to exemplify the communication. The organisation is visualised via tasks, units, information and information flow.

These basic objects of [19] have been slightly improved to enable a better distinction between explicit knowledge and human oriented knowledge.

E.4. Comparison of PROMOTE, DÉCOR, and ARIS

The following table depicts of overview on the three major modelling methods PROMOTE, DÉCOR, and ARIS.

		Static models		Dynamic models	
Business models	PROMOTE	Working Environment (WE)	Describes the organisational structure	Business Process (BP)	Describes the organisational processes.
	DÉCOR	Included in BP		Business Process (BP)	
	ARIS	Included in BP		Business Process (BP)	
Knowledge models	PROMOTE	Skill Documentation (SD)	Describes Skill-profiles and Competences.	Knowledge Management Processes (KMP)	Describes knowledge interaction.
	DÉCOR	---		---	
	ARIS	Knowledge landscape		Knowledge usage / development	
	PROMOTE	Knowledge Structure (KS)	Description of topics and themes.	Knowledge Process (KP)	Describes knowledge intensive tasks.
	DÉCOR	Ontology	Description of units, attributes, and links.	Weakly structure Workflows	Description of weakly structured ad-hoc workflows.
	ARIS	Knowledge structure		---	
Overview model	PROMOTE	Community Model (CM)	Overview of interest groups and communities of practice.	Process Pool (PP)	Overview of processes.
	DÉCOR	---		---	
	ARIS	---		---	
	PROMOTE	Knowledge landscape (KL) Overview of all models			
	DÉCOR	---			
	ARIS	---			
Modelling Tool	PROMOTE	PROMOTE® / ADONIS®			
	DÉCOR	MS-Visio®			
	ARIS	ARIS®			
Platform	PROMOTE (Prototype)	PROMOTE® Portal with an underlying Meta-Model Database			
	DÉCOR (Prototype)	CognoVision® Portal with an underlying Document Management System			
	ARIS (Product)	Hyperwave® Portal with an underlying Document Management System			

Table 24 Model language and Platform comparison

The above table shows differences between the modelling approaches of PROMOTE, DÉCOR and ARIS and indicate the holistic framework of PROMOTE that covers all main aspects. To establish an overview of knowledge models, the PROMOTE modelling method is depicted in more detail.

E.5. References of Annex E

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

[2] Wiig K.: An Emerging Discipline Rooted in a Long History, Knowledge Research Institute Inc., Draft of Chapter 1. In: Knowledge Management, Hrsg. v. D. Chauvel; C. Despres. scheduled for publication Fall 1999.

[3] Probst G., Raub S., Romhard K., Wissen managen. Wie Unternehmen ihre wertvollste Ressource optimal nutzen. Gabler, 1997, Wiesbaden

[4] Woitsch R., Karagiannis D., „Process-Oriented Knowledge Management Systems Based on KM-Services: The PROMOTE® Approach", In Proceeding of the fourth International Conference on Practical Aspects of Knowledge Management (PAKM 02), 2-3 December 2002, Vienna, Austria

[5] Abecker A., Hinkelmann K., Maus H., Müller H. J., Geschäftsprozess-orientiertes Wissensmangement, Springer Verlag, 2002, Berlin

[6] KPMG, The knowledge journey: A Business Guide to Knowledge Management, http://www.atoskpmgconsulting.co.uk/research/othermedia/wf2_kbase.pdf, access: 31.08.03

[7] Dengel A., Abecker A., Bernardi A., van Elst L., Maus H., Schwarz S., Sintek M., Konzepte zur Gestaltung von Unternehmensgedächnissen, in Künstliche Intelligenz 1/02, Gesellschaft für Informatik, arendtap, Bremen 2002, pp5-11

[8] Lehner, F.: Organisational Memory. Konzepte und Systeme für das organisatorische Lernen und das Wissensmanagement. München u.a.: Hanser 2000.

[9] Karagiannis D., Telesko R., (2000) "The EU-Project PROMOTE: A Process-oriented Approach for Knowledge Management", In Proceedings of the Third International Conference on Practical Aspects of Knowledge Management (PAKM 2000), October 30-31 2000,Basel, Switzerland

[10] Karagiannis D., Woitsch R.,(2002a) "Modelling Knowledge Management Processes to describe organisational knowledge systems", In Proceedings of 15th European conference on Artificial Intelligence, WS Knowledge Management and Organizational Memories, 21-26 July 2002, Lyon, France

[11] Karagiannis D., Woitsch R., (2002b) "The PROMOTE Prototype: A meta2Model Based Process Oriented KMS", In Proceedings of the Theory and Application of Knowledge Management (TAKMA 2002) in conjunction with Database and Expert Systems Application (DEXA 2002), September 2-6 2002, Aix-en-Provence, France

[12] Stefanidis G., Karaginnis D., Woitsch R. (2002), "The PROMOTE approach: Modelling Knowledge Management Processes to describe knowledge management systems", In Proceedings of the third European Conference on Organizational Knowledge, Learning, and Capabilities (OKLC 02), 5-6 April 2002, Athens, Greek

[13] Abecker A., Mentzas G., Legal M., Ntioudis S., Papavassiliou G., Business Process Oriented Delivery of Knowledge through Domain Ontologies, Proceedings of DEXA conference, Second International Workshop on Theory and Applications of Knowledge Management, 3-7 September, 2001, Munich

[14] Papavassiliou G., Ntioudis S., Mentzas G., Abecker A., Business Process Knowledge Modelling: Method and Tool, In Proceedings of the Theory and Application of Knowledge Management (TAKMA 2002) in conjunction with Database and Expert Systems Application (DEXA 2002), September 2-6 2002, Aix-en-Provence, France

[15] ARIS for Hyerwave, www.ids-scheer.de/sixcms/detail.php/4977, access 15.08.02

[16] Hyperwave Homepage, www.hyperwave.com, access: 15.08.02

[17] Hyperwave, White Paper, 2002

[18] Gronau N., Modellieren von wissensintensiven Geschäftsprozesses mit der Beschreibungssprache K-Modeler, in Gronau N. Wissensmanagement: Potentiale – Konzepte - Werkzeuge, GITO Verlag, Berlin 2003, p3-30
[19] Hoyer R., Organisatorische Voraussetzungen der Büroautomation. Rechnergestützte, prozeßorientierte Planung von Büroinformations – und Kommunikationssystemen. Reihe Betriebliche Informations- und Kommunikationssysteme – Band 11. Berlin 1988 referred in [18]

F. The PROMOTE® Modelling Method

F.1. The Knowledge Map

The knowledge map describes an overview of the whole roadmap. This model will be used to identify the process, structures and additional relevant information.

The Knowledge Map consists of the following elements:

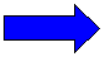
Element	Icon	Description
Subject		The "Subject" is a reference to an object of either the working environment or the skill documentation.
K_Category		The "K_Category" is a reference to a Category from the Knowledge structure model or to a Product from the Knowledge product model.
K_Topic		The "K_Topic" is a reference to the Topic from the Knowledge structure model.
K_Resource		The "K_Resource" is a reference to the Resources from the Knowledge resource model.
K_Process		The "K_Process" is a reference to a Process from the Company map.
K_Tool		The "K_Tool" is a reference to a Tool from a Knowledge Tool Model.
Aggregation		The "Aggregation" supports the logical structure of model contents on the drawing area.
Note		The "Note" allows free text to be placed within a model.

Table 25 Notation of the Knowledge Map in PROMOTE®

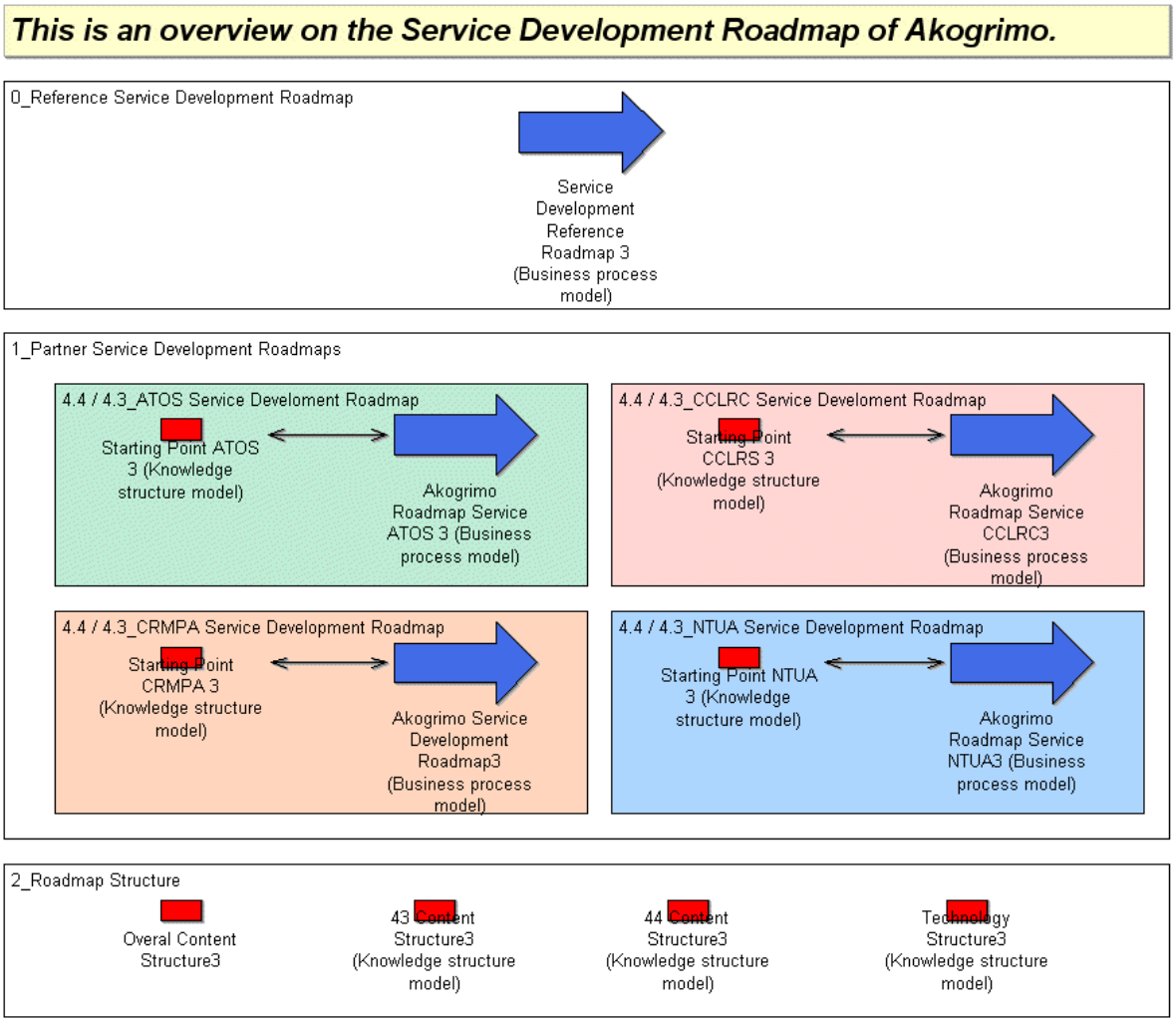


Figure 24 Knowledge Map showing an Overview of the Roadmap

F.2. The Knowledge Product Model

The knowledge product model describes how the information will be provided to the user. This model specifies the type of the product, responsibilities and provides a description of how the product will be used.

The Knowledge Product Model consists of the following elements:

Element	Icon	Description
Knowledge product		The "Knowledge Product" specifies a product. Responsibilities are assigned. Furthermore a description about the usage and the access to the product is provided.
Aggregation		The "Aggregation" supports the logical structure of model contents on the drawing area.
Note		The "Note" allows free text to be placed within a model.

Table 26 Notation of the Knowledge Product Model in PROMOTE®

F.3. The Process Map

The process map is an overview model that identifies all processes within the roadmap. This includes business processes that in this scenario define the roadmap as well as knowledge management processes that define the knowledge transfer.

The Process Map consists of the following elements:

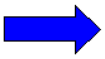
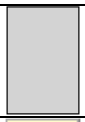
Element	Icon	Description
Process		A „Process“ is a reference on a Company Map, to a business process model, a knowledge management model or to another company map.
Aggregation		The “Aggregation” supports the logical structure of model contents on the drawing area.
Note		The “Note” allows free text to be placed within a model.

Table 27 Notation of the Process Map in PROMOTE®

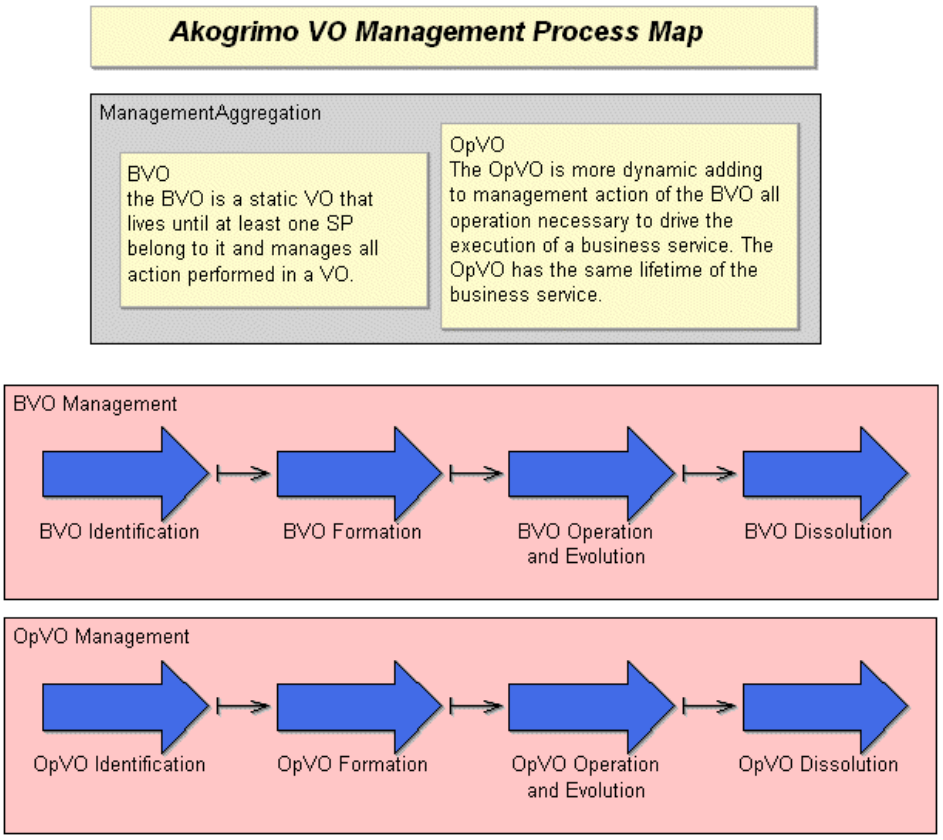
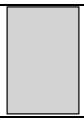


Figure 25 Sample Process Map of possible platform processes

F.4. The Business Process Model

The business process model describes a sequence of logical related activities. In this scenario this model type is used to define the roadmap.

The Business Process Model consists of the following elements:

Element	Icon	Description
Process start		The "Process start" symbolises the beginning of a business process. Each business process model has to contain exactly one Process start object. The object "Process start" has no predecessor and only one successor and it can be set with variables.
Sub process		The "Sub process" is used to call other processes. This makes sense especially when a particular process is carried out in several areas of the business. Rather than repeating the same process within a larger model a number of times, it makes sense to create a sub process, which can be called wherever necessary. The Sub-process object can also be useful when you want to structure your business processes to maximise clarity. It must have at least one predecessor and at most one Successor.
Activity		"Activities" describe which tasks in a business process model have to be executed. The object "Activity" has at least one predecessor and at most one Successor. The level of granularity at which activities are described can vary according to the requirements of the modeller.
Decision		The "Decision" within a business process model allows pre-defined variables to be queried in order to determine the path through the process. A decision has at least one predecessor and at least one Successor. During a simulation only one of the Successors must evaluate to "true".
Parallelity		The "Parallelity" makes it possible for several paths in a business process to be executed at the same time. Every parallelism ends with a union (Merging object). The object Parallelity has at least one predecessor and one successor. A transition condition can be set on a successor relation following a Parallelity object. This means that one (or more) of the "parallel" paths may be followed only in certain conditions. The object "Parallelity" should have the same number of Successors as the corresponding "Merging" object has predecessors.
Merging		With the "Merging" object, parallel paths are re-joined. The object "Merging" has as many predecessors as the corresponding object "Parallelity" successors, and it has at most one successor.
End		The object „End" marks the end of a path of a business process. Several objects of the class "End" are allowed in one model. The object "End" has at least one predecessor and no successor.
Group		The "Group" supports the logical structure of model contents on the drawing area.

Note		The "Note" allows free text to be placed within a model.
Variable		Variables are used to enable transition conditions to be defined in the model. These transition conditions are set in the "Successor" relation (after a Decision or Parallelism). The object "Variable" has no predecessor or successor. The object "Variable" can be related to the object "Random generator" with the relation "Sets variable". One variable can be set with different values at different times through the use of several "Random generators".
Random generator		The Random generator sets a variable with a certain value. The value can be set through a steady or a discrete distribution function. The object „Random generator" has no predecessor or Successor. The object "Random generator" can be related to the following objects: "Variable" through the relation "Sets variable" "Process start", "Activity" through the relation "Sets". It can also be related to other objects such as "Decision", "Parallelism" etc. through the relation "Sets". Several Random generators can set the same "Variable" object with values.

Table 28 Notation of the Business Process in PROMOTE®

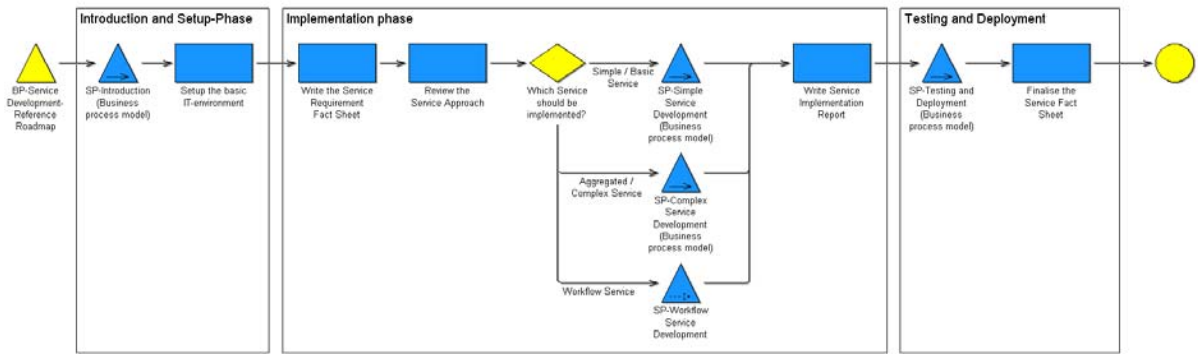


Figure 26 Example Business Process Model "How to implement an Akogrimo service?"

F.5. The Knowledge Management Process

The knowledge management process defines the knowledge flow within an organisation. In this scenario this model type is not used.

The Knowledge Management Process Model consists of the following elements:

Element	Icon	Description
K-Start		Every Knowledge Management Process model must contain exactly one Process start object, which symbolises the beginning of the Knowledge Management Process model. The object "K-Start" has no predecessor and only one successor and it can be set with variables.
KMP-Sub-process		It is possible to call other KM-Processes with the object "KMP-Subprocess". This makes sense especially when a particular KM-Process is carried out in several areas of the business. Rather than to repeat the same KM-Process within a larger model a number of times, it makes sense to create a KMP-Subprocess which can be called whenever necessary. The KMP-Subprocess object can also be useful when you want to structure your KM-Processes to maximise clarity. It must have at least one predecessor and at most one successor.
K-Activity		K-Activities describe which tasks in an information flow have to be executed. The object "K-Activity" has at least one predecessor and at most one successor. The level of granularity at which activities are described can vary according to the requirements of the modeller.
Decision		The "Decision" within a Knowledge management process model allows pre-defined variables to be queried in order to determine the path through the process. A decision has at least one predecessor and at least one Successor. During a simulation only one of the Successors must evaluate to „true".
Parallelity		The "Parallelity" makes it possible for several paths in a Knowledge managment process to be executed at the same time. Every parallelism ends with a union (Merging object). The object Parallelity has at least one predecessor and one successor. A transition condition can be set on a successor relation following a Parallelity object. This means that one (or more) of the „parallel" paths may be followed only in certain conditions. The object "Parallelity" should have the same number of Successors as the corresponding "Parallelity" object has predecessors.
Merging		With the "Merging" object, parallel paths are re-joined. The object „Merging" has as many predecessors as the corresponding object „Parallelism" successors, and it has at most one successor.
K-End		The object "K-End" marks the end of a path of a Knowledge management process. Several objects of the class "K-End" are allowed in one model. The object "K-End" has at least one predecessor and no successor.

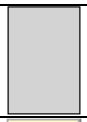
Aggregation		The "Aggregation" supports the logical structure of model contents on the drawing area.
Note		The "Note" allows free text to be placed within a model.
Variable		Variables are used to enable transition conditions to be defined in the model. These transition conditions are set in the "Successor" relation (after a Decision or Parallelism). The object „Variable" has no predecessor or successor. The object "Variable" can be related to the object "Random generator" with the relation "Sets variable". One variable can be set with different values at different times through the use of several "Random generators".
Random generator		The Random generator sets a variable with a certain value. The value can be set through a steady or a discrete distribution function. The object „Random generator" has no predecessor or Successor. The object „Random generator" can be related to the following objects: "Variable" through the relation „Sets variable" "Process start", "Activity" through the relation "Sets". It can also be related to other objects such as "Decision", "Parallelity" etc. through the relation "Sets". Several Random generators can set the same "Variable" object with values.

Table 29 Notation of the Knowledge Management Process in PROMOTE®

F.6. Team Map

The team map is an overview model that identifies all human and organisational relevant aspects. This includes organisational units, departments, roles, employees or skills. This model type will not be used in this scenario.

The Team Map consists of the following elements:

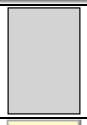
Element	Icon	Description
Subject		The "Subject" is a reference to an object of either the working environment or the skill documentation.
Aggregation		The "Aggregation" supports the logical structure of model contents on the drawing area.
Note		The "Note" allows free text to be placed within a model.

Table 30 Notation of the Team Map

F.7. The Working Environment Model

The Working Environment Model describes the organisational environment such as organisational units, performers and roles.

The Working Environment Model consists of the following elements:

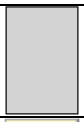
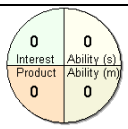
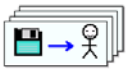
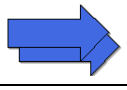
Element	Icon	Description
Organisational unit		Definition of organisational units within a working environment.
Performer		Definition of a performer who belongs to an organisational unit.
Role		The Role describes the task-range or job description of a performer. Every performer can have one or more roles. Several performers can have the same role.
Resource		The "Resource" represents an EDP system, a communication system or any other resource, which is necessary for executing the business process.
Aggregation		The "Aggregation" supports the logical structure of model contents on the drawing area.
Note		The "Note" allows free text to be placed within a model.

Table 31 Notation of the Working Environment Model

F.8. The Skill Environment Model

The skill environment model describes the skill-profiles of persons and optionally enables the definition of knowledge access rights. This model type will only partly be used in the roadmap scenario.

The Skill Environment Model consists of the following elements:

Element	Icon	Description
Person		A "Person" is a specific individual within an organisation. The working environment model can refer to this element. A person has a specific role, skills and specific permissions.
Knowledge role		The "Knowledge role" is a role to group persons with specific knowledge.
Skill profile		Specific "Skill profile" of a person. The skill profile belongs to one of the four skill types (Interest, Ability (self), Ability (manager) and Product skills).
Aggregated skills		The "Aggregated skills" define an aggregated view of different skill profiles of a person. The skills are subdivided into Interest, Ability (self), Ability (manager) and Product skills.
Service group		The "Service group" defines several services, which logically belong together. Permission can be assigned to the "Service group", to give access to the group.
Process group		The "Process group" is a reference to one or several processes. Permission can be assigned to the "Process group", to give

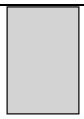
		access to the group.
Permission		The "Permission" describes permission in detail. A person can refer to this element. It is then the person has this Permission (e.g. read-only)
Aggregation		The "Aggregation" supports the logical structure of model contents on the drawing area.
Note		The "Note" allows free text to be placed within a model.

Table 32 Notation of the Skill Environment Model

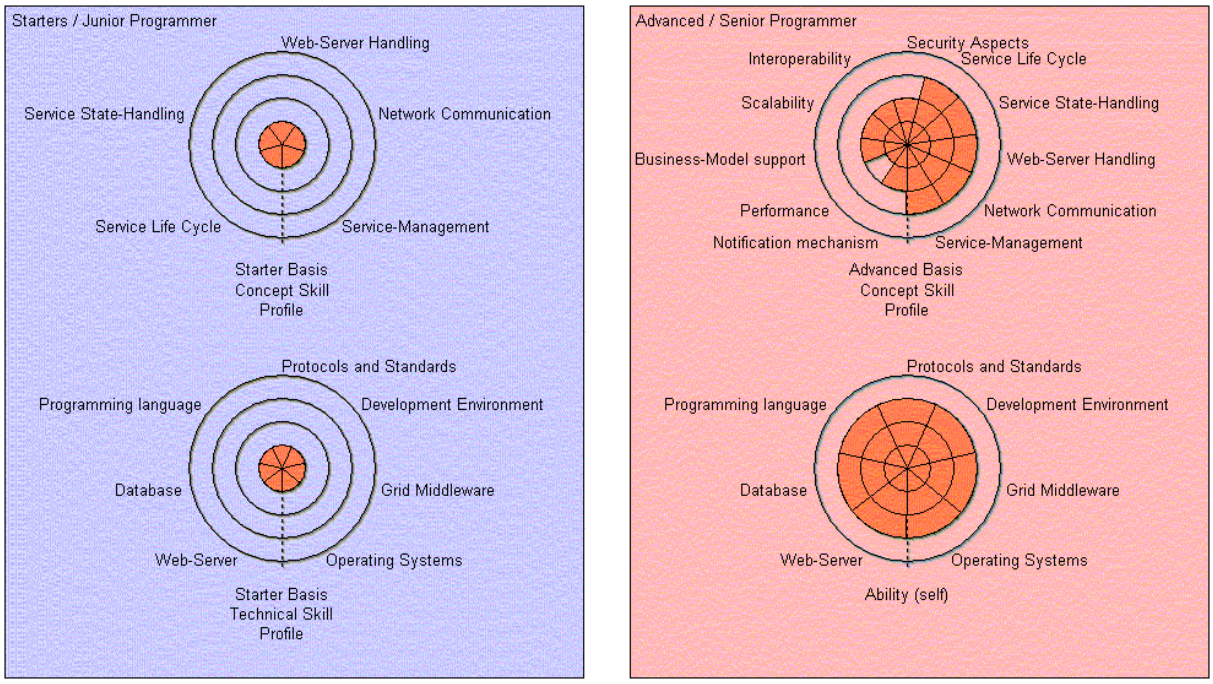


Figure 27 Akogrimo Example Skill Profiles

F.9. Knowledge Structure Model

The knowledge structure model enables the classification of the content. This model type will be used to define the starting point and the end point of the roadmap as well as to define the organisational, the technical, the human and the content structure.

The Knowledge Structure Model consists of the following elements:

Element	Icon	Description
Category		The category defines the structure of a knowledge source. This structure can be independent of the content that is defined via the model object "topic". Categories are seen as explicitly defined.
Topic		The topic defines the content of a knowledge source. This content can be independent of the structure that is defined via the model object "category". Topics depend on the interpretation of communities and are therefore individualised.
Category aggregation		The "Category aggregation" is an element for grouping categories.
Topic aggregation		The "Topic aggregation" is an element for grouping topics.
Note		The "Note" allows free text to be placed within a model.

Table 33 Notation of the Knowledge Structure Model

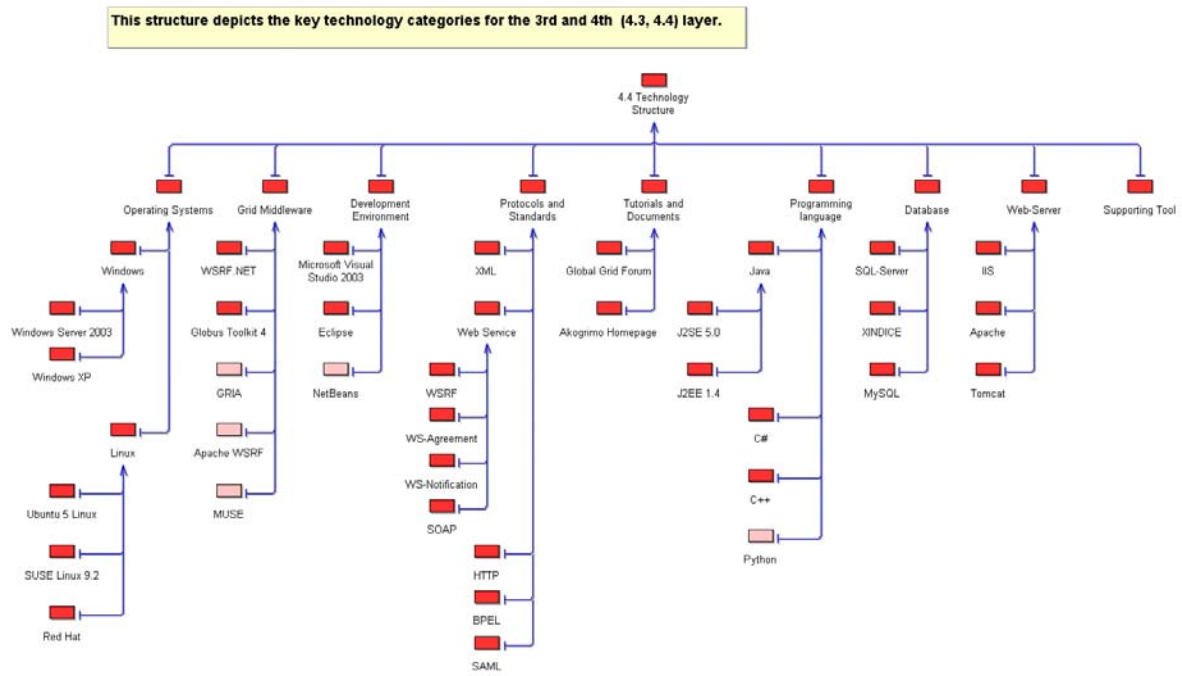


Figure 28 Example Knowledge Structure Model of the Technological Infrastructure

F.10. Knowledge Tool Model

The knowledge tool model is a pool model that collects and describes all relevant tool links.

The Knowledge Tool Model consists of the following elements:

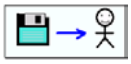
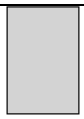
Element	Icon	Description
Tool		With the "Tool" object all necessary tools for the execution of the processes are described. Furthermore the configuration of the tools is described.
Service		Services and the configuration of them are described, using the "Service" object.
Aggregation		The "Aggregation" supports the logical structure of model contents on the drawing area.
Note		The "Note" allows free text to be placed within a model.

Table 34 Notation of the Knowledge Tool Model

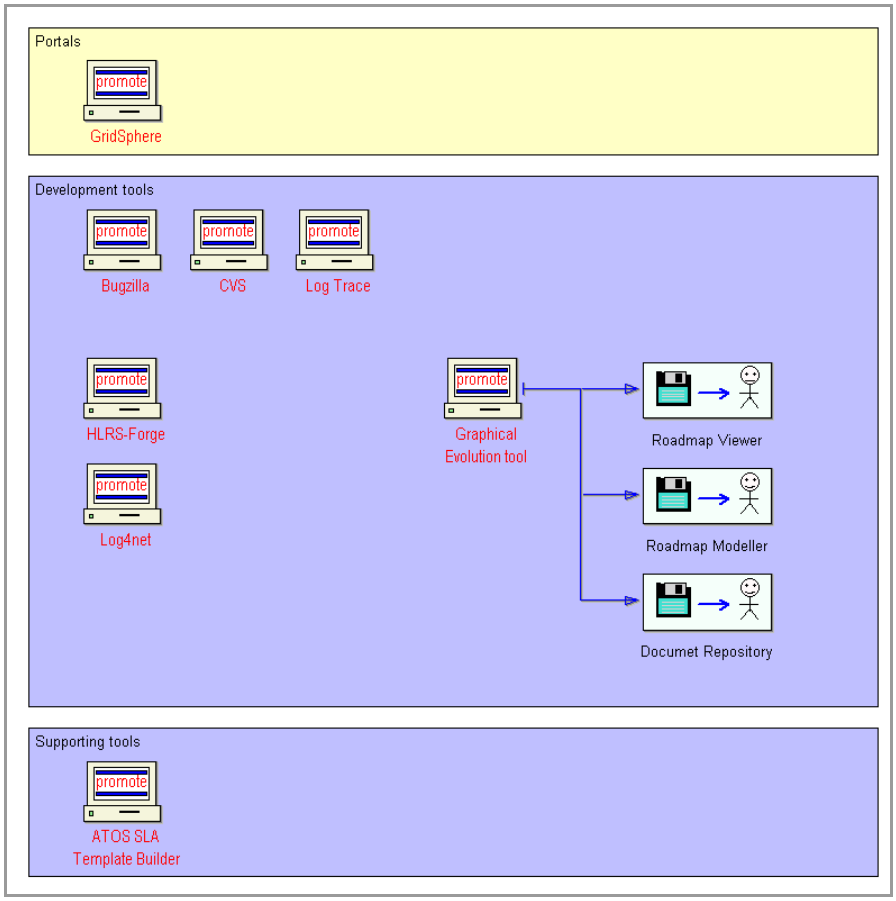


Figure 29 Example Knowledge Tool Model showing Supporting Tools

F.11. Knowledge Resource Model

The knowledge resource model is a pool model that collects and describes all relevant resource links.

The Knowledge Resource Model consists of the following elements:

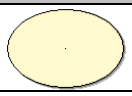
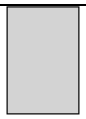
Element	Icon	Description
Knowledge source		A "Knowledge source" describes the source of the knowledge and how to access it.
Knowledge resource		With the help of the "Knowledge resource" a resource is described in detail. There are different types of "Knowledge resources", e.g. a Document, a Database, the Internet or a Performer.
Resource aggregation		The "Resource aggregation" groups related resources.
Note		The "Note" allows free text to be placed within a model.

Table 35 Notation of the Knowledge Resouce Model

G. Annex: The AKBPM Procedure

This Annex indicates the relationship of AKBPM models, and how to navigate through the models starting on a high business level. To demonstrate the AKBPM method the eHealth Scenario has been selected to model the processes

Starting on a high business level the Business Model enables the definition of application scenarios. Figure 30 shows the Business Model of the eHealth Scenario, as it is used within the Application Development Scenario.

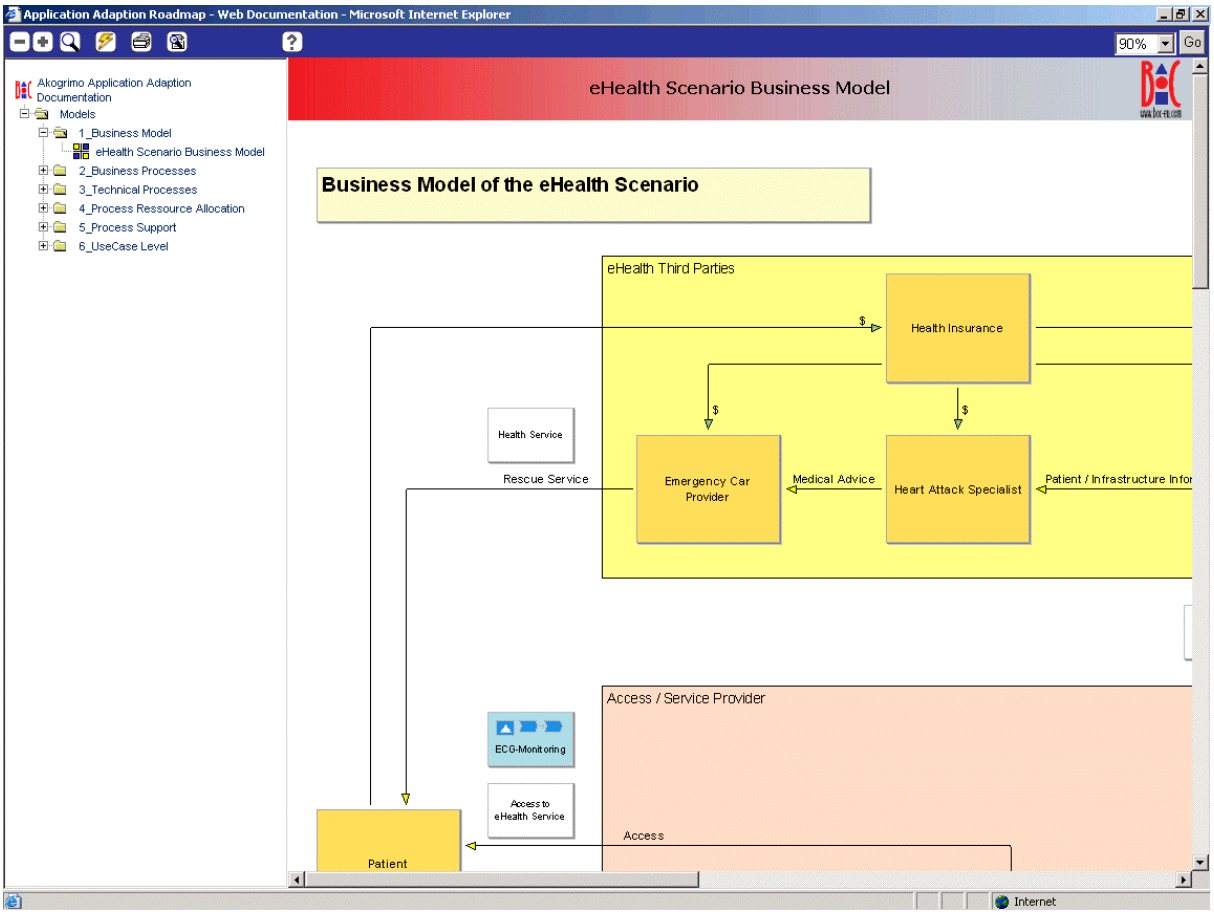


Figure 30 The eHealth Scenario Business Model

With the help of the Business Model all relevant actors, flows and business processes can be identified. Processes can either be high-level organisational processes or low-level technical processes. An identified process for the eHealth scenario is the Akogrimo Emergency Scenario process.

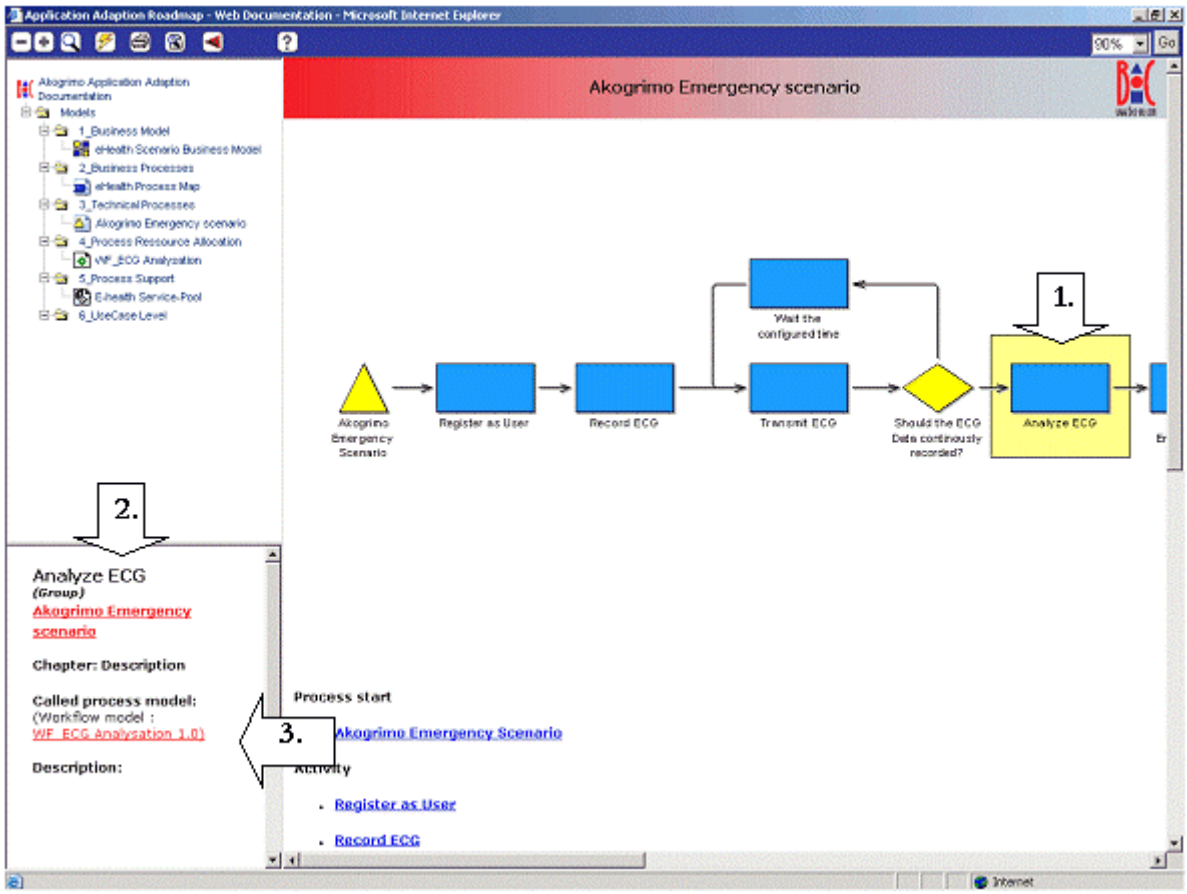


Figure 31 The Akogrimo Emergency Scenario Process

Figure 31 shows the Akogrimo Emergency Scenario Process as it is used in the Application Development Scenario. To navigate through the models there are three steps necessary:

1. A mouse click on the yellow Group opens the details window on the left hand side of the window.
2. Here the user can look for references to other models. In case there are references, the model is named and a link is provided. In the case of this group a workflow model called ECG Analysation is referenced.
3. A click on the red underlined link opens the model in a new window, as the following figure illustrates.

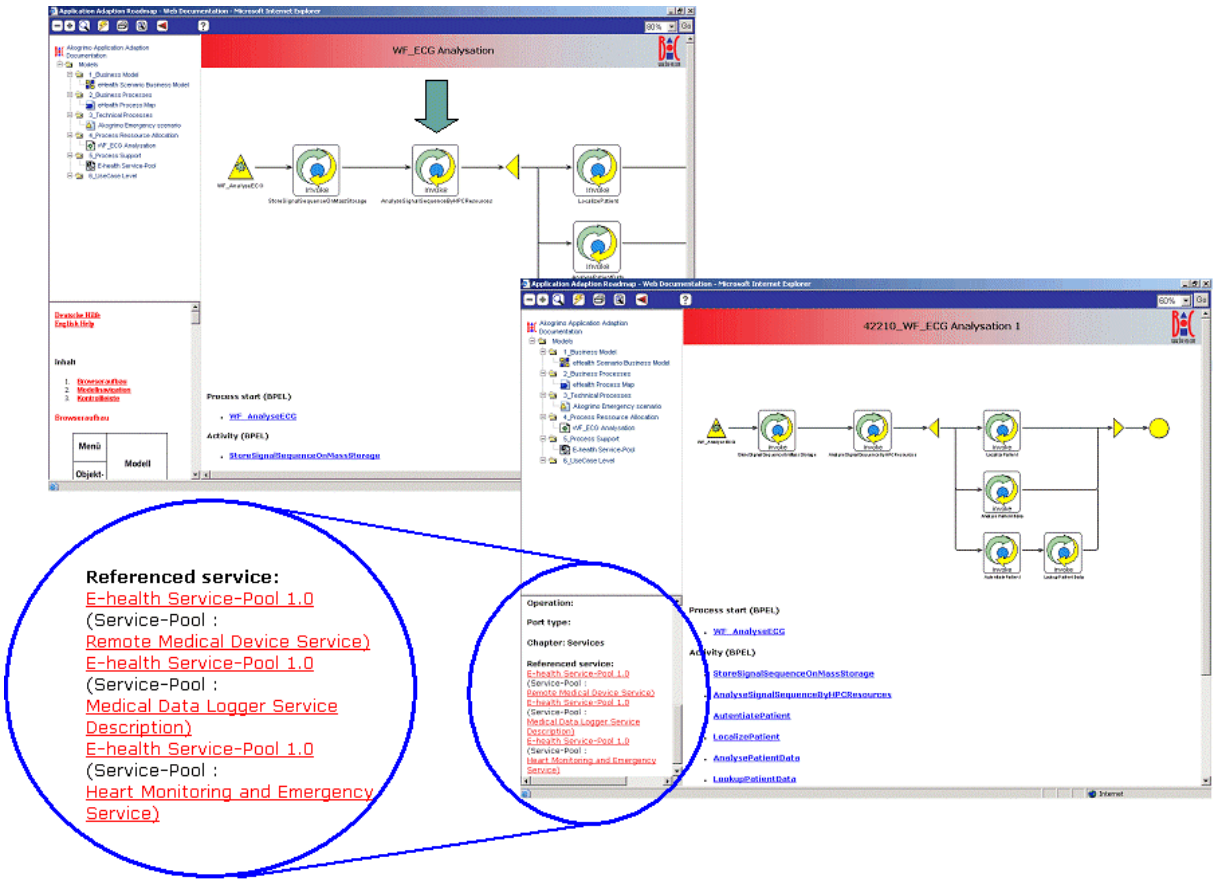


Figure 32 Workflow Model ECG Analysation

As in the previous described example, a click on any symbol opens the details window. In Figure 32 the activity "Analyse Signal Sequence" is chosen, leading to the corresponding details window opening on the left side of the window. The details window shows links to four services, which are necessary for the execution. Clicking on the Service the user wants to look at again opens the corresponding model. In the case of this example the eHealth Service Pool opens as shown in the following picture.

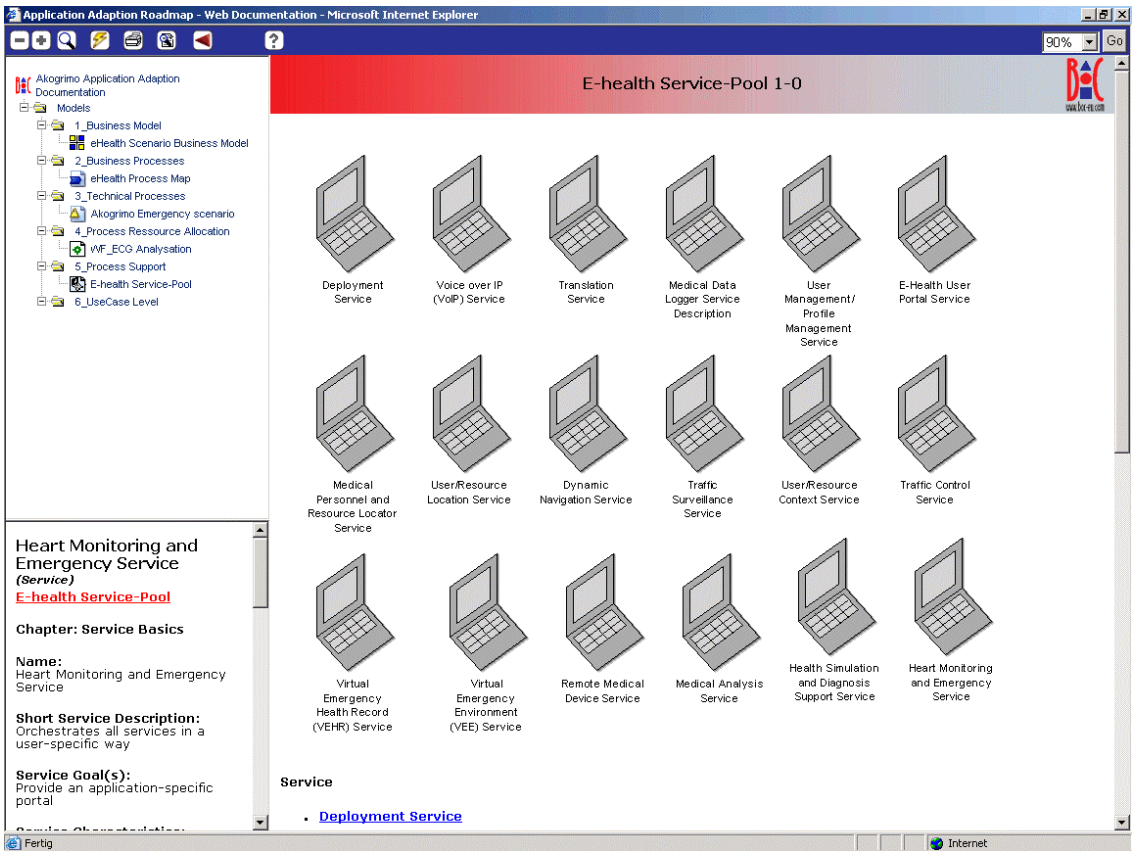


Figure 33 The eHealth Service Pool

Figure 33 shows the eHealth Service Pool. Clicking on a Service the user wants to look at opens the corresponding details window. There the user can find information about the service. The details describe the behaviour of the service and are called "Business Service Sheets".