Grant Agreement N° 215483

Title:	*First Version of Integration Framework*
Author:	*UniDue, Tilburg, CITY, CNR, FBK, INRIA, Lero-UL, POLIMI, SZTAKI, TUW, UCBL, UOC, UPM, USTUTT, UniHH, VUA*
Editor:	*Andreas Gehlert (UniDue), Marco Pistore (FBK), Pierluigi Plebani (POLIMI), Loredana Versienti (CNR)*
Reviewers:	*Andreas Metzger (UNIDUE)*
	Michael Parkin (Tilburg)
	Raman Kazhamiakin (FBK)
Identifier:	*Deliverable # CD-IA-3.1.3*
Type:	*Deliverable*
Version:	*1.0*
Date:	*15 December 2009*
Status:	*Final*
Class:	*External*

Management Summary

This document describes the first version of the S-Cube Integrated Research Framework (IRF). The main goal of the IRF is to define a coherent holistic framework that integrates the principles, techniques, methods and mechanisms provided by the joint research activities JRA-1 and JRA-2, and the results of the validation and empirical evaluation obtained from WP-IA-3.2.
Aim of this deliverable is twofold. On the one hand, we introduce the main elements composing the IRF: views, research challenges, research questions, and research results. These elements are classified according to the perspectives introduced in the IRF baseline (i.e., conceptual framework, reference life-cycle, logical run-time architecture, logical design environment) and the terms included in the Knowledge Model. On the other hand, in order to support the management of the IRF, in this paper we also introduce the web application developed on purpose that forms integral and substantial part of this document.

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under grant agreement n° 215483 (S-Cube).

File name: CD-IA-3.1.3.pdf

Members of the S-CUBE consortium:

University of Duisburg-Essen	Germany
Tilburg University	Netherlands
City University London	U.K.
Consiglio Nazionale delle Ricerche	Italy
Center for Scientific and Technological Research	Italy
The French National Institute for Research in Computer Science and Control	France
Lero - The Irish Software Engineering Research Centre	Ireland
Politecnico di Milano	Italy
MTA SZTAKI – Computer and Automation Research Institute	Hungary
Vienna University of Technology	Austria
Université Claude Bernard Lyon	France
University of Crete	Greece
Universidad Politécnica de Madrid	Spain
University of Stuttgart	Germany
University of Hamburg	Germany
VU Amsterdam	Netherlands

Published S-CUBE documents

These documents are all available from the project website located at http://www.s-cube-network.eu/

Contents

Chapter 1

Introduction

This deliverable describes the first version of the S-Cube Integrated Research Framework (IRF), which is one of the pivotal elements for ensuring overall integration, consistency and harmonisation of the research efforts undertaken by the network. This represents the first outcome of the of the activities undertaken in Task T-IA-3.1.2 of the IA-3 activity.

In this introductory chapter, we define the context of the IRF. We start by describing the objectives of IA-3, where the definition of the IRF belongs (Section 1.1). We then discuss the approach that we adopted for defining the IRF (Section 1.2). We finally discuss the structure of the rest of the deliverable (Section 1.3).

1.1 Activity IA-3: Integration Framework

The overall goal of Integration Activity 3 (Integration Framework for Service-based Applications) is to guarantee the coherency and integration of the research efforts undertaken by S-Cube [1]. This is achieved through the definition of a holistic framework that aligns and coordinates the results of the joint research activities. This overall goal will be achieved through the realization of the following detailed objectives:

- To define, and progressively refine, the Integrated Research Framework (IRF), i.e., a coherent, holistic framework for S-Cube research, which allows for integrating the principles, techniques, methods and mechanisms studied in S-Cube. In particular, the IRF should encompass those aspects of the research that are cross-cutting.
- To guarantee the overall coherence and alignment of all the research work-packages by defining their contributions to the IRF and by identifying boundaries and interfaces among the investigations undertaken by the different work-packages.
- To validate the IRF through suitable industrial case studies. The ultimate goal of the validation is to revise and improve the IRF. For this reason, it will be conducted iteratively for the whole duration of the activity.
- To identify the different classes of users that are involved with different roles in a Service Based Application (SBA), and to define customisations and refinements of the IRF that are tailored and personalised to these various classes of users.
- To help comparing the investigation undertaken by S-Cube with other proposals and approaches, both from industry and from academics; to help strengthening the relations of S-Cube with the research community; and to help identifying missing competences within the S-Cube consortium.

1.2 The approach

The First version of the IRF enriches the previously delivered baseline [5], also considering the relevant research challenges, questions, and results, their relationships with the IRF elements and the Industrial Case Studies (see, IA-2.2) [4]. In addition, a strict connection to the Validation Framework defined in the IA-3.2 activities is defined.

As discussed in Chapter 2, since the IRF collects contributions from all the partner involved in the project, we involve all the partners to (i) define the elements composing the Framework, (ii) make these elements consistent to the Validation Framework and the Industrial Case Studies, and (iii) make possible a continuous update of the elements in the framework according to the research planned to be done in the project in the future. As a result of this effort we obtained:

- A *Conceptual Model* to collect all the relevant elements composing the IRF and the relationships among these elements. In this document, the conceptual model is represented by describing the set of tables that will be used as a basis for the web application.
- A *Web application* to make possible to both S-Cube and non S-Cube researchers to easily understand and possibly contribute to the research framework. At this stage, the Web application is available under the Web portal and it is not yet visible to non-authenticated users. We plan to make it available to external user whenever a consolidated version of the IRF will be defined.

It is worth noting that the content of the IRF available through the Web application forms integral and substantial part of this deliverable. Thus, the goal of this deliverable is to discuss the model of the IRF and the way in which this model is implemented. The specific research results achieved in the Joint Research Activities are visible by accessing to the Web application.

1.3 Structure of the deliverable

The structure for the remaining chapters in the deliverable is the following. In Chapter 2, we introduce the approach followed to define the elements of the IRF. As a result, in Chapter 3, we introduce the conceptual model of the IRF. Such a model puts the basis for the design and the development the web application supporting the management of the IRF as discussed in Chapter 4. Since this application had been used by the partners to insert all the information about the research conducted so far according to the IRF, in Chapter 5, we give an overview about the information currently inserted. Finally, Chapter 6 provides some concluding remarks.

Chapter 2

Framework structure

The main goal of the IRF is to define a coherent holistic framework that integrates the principles, techniques, methods and mechanisms provided by the joint research activities JRA-1 and JRA-2, and the results of the validation and empirical evaluation obtained from WP-IA-3.2. Since the sources of this information are the S-Cube partners, we involved them in the definition of this first version of the IRF. The resulting framework takes into account all the elements able to describe the on-going research in the project. Moreover, the same framework can be adopted to also include the research issues that will be considered in the future.

The following elements put the basis for the IRF definition:

- *Research activities*: they represent the scientific elements of the S-Cube project. In these activities, organized in 6 workpackages, we aim at covering most of the topics about the design and the execution of service based application. As one of the results of the first 18 months of the project, it is now clear which are the main research challenges and questions.
- *IRF Baseline*: it represents the first attempt to give a coherent view of the research efforts that should be done in the project. As a consequence, the IRF baseline is the natural starting point for defining the first version of the IRF.
- *Validation Scenario*: The IA-3.2 workpackage's first result, presented in the deliverable CD-IA-3.2.1, validate the conformance of the IRF baseline with respect to the aim of the project and the work done in the research activities.
- *Industrial Case Study*: published in the deliverable IA-2.2.2, the selected five main case studies represent possible interesting situations in which the service based applications studied in S-Cube coulf be useful.

To have a complete overview of the on-going research, the S-Cube partners identify the main *research challenges* with respect to the main relevant topics of the related workpackage. More precisely:

- WP-1.1: Engineering Principles, Techniques and Methodologies for Hybrid, Service-based Applications.
- WP-1.2: Adaptation and Monitoring Principles, Techniques and Methodologies for Service-based Applications.
- WP-1.3: End-to-End Quality Provision and SLA Conformance.
- WP-2.1: Business Process Management (BPM).
- WP-2.2: Adaptable Coordinated Service Compositions.

- WP-2.3: Self-* Service Infrastructure and Service Discovery Support.

As discussed in Section 2.1, the list of the research challenges is used by all the S-Cube partners as a reference point for classifying the identified *research questions* and, if any, the *research results* achieved so far. In addition, to obtain a coherent view of the IRF, the workpackage leader needs to ensure the quality of the different contributions coming from the partners.

2.1 IRF definition and quality Assurance

The definition of the IRF is a collective effort undertaken by all the partners of the S-Cube project. In particular, all the problems and questions that are investigated, and all the results that are achieved by the partners within the scope of the project, will be submitted as elements of the IRF. This way, the IRF will give a comprehensive picture of the scientific activities of the project.

Clearly, not all the part of the IRF are supposed to evolve at the same rate. In particular, the more general parts, including the different views to the project introduced in CD-IA-3.1.1, the research challenges, and the industrial case studies, will be more stable since they capture long term organization and objectives of the project. The more specific parts of the IRF will include instead the specific research and validation achievements of the project, and will be expanded, refined and revised frequently during the life of the project.

In order to guarantee the quality and coherence of the contributions that appear in the IRF, a lightweight quality assurance process has been defined. The aim is to identify duplicated or misplaced contributions, and to check the coherence of the partner contributions with the overall S-Cube research plan and strategy. More precisely, each partner contribution has to be validated by a responsible person before it is accepted as part of the IRF. The JRA work-package leaders have the responsibility of validating the specific research and validation achievements; this guarantees a fast, flexible validation. For what concerns the more general components of the framework, instead, the responsibility of validating and accepting the changes is assigned to the Integration Committee, which has to evaluate the overall effect of these changes on the project as a whole.

2.2 IRF quality Validation

Generally, workpackage WP-IA-3.2 will validate the research framework through a series of deliverables [6]. The idea is establish a cycle of building and validating the framework throughout the entire duration of the project. The continuous validation of the framework will be achieved by internal verification and external validation (see Figure 2.1):

- Internal Verification: Internal verification checks the consistency of the integration framework contents and, therefore, ensures the overall quality and integrity of the integration framework. E.g., this verification will reveal research questions without research results or research results without proper validation (gap analysis). These problems will be mirrored back to the joint research activities and to the mobility program. While joint research activities may decide to close those gaps, e.g. by producing research results or by validating them, the mobility program will act as driver and will trigger mobility exchanged, which are likely to close the identified gaps.

- External Validation: External validation ensures that the integration framework is useful outside the S-Cube project. Here we distinguish between external validation in research and industry:

  - External Validation in Research: The focus of this validation activity is on estimating how the integration framework covers existing research frameworks in the Service Oriented Computing research community.

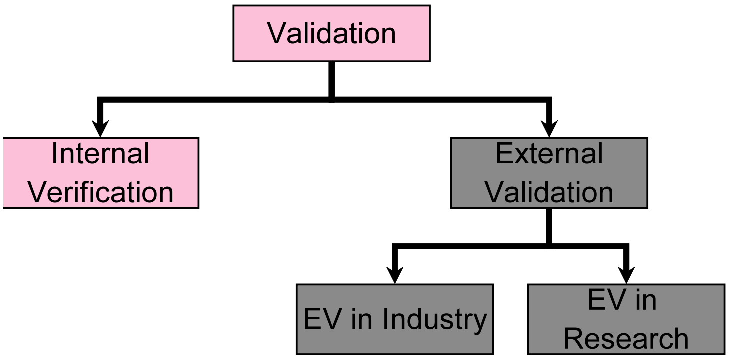


Figure 2.1: Validation schema.

- External Validation in Industry: The focus here is to estimate how much the S-Cube project and particularly its integration framework influences industrial research agendas. This external validation will be carried out in close collaboration with the workpackage IA-2.2.

Given the fact that the definition level of the integration framework will be complete at the 3rd quarter of year 2 [7], the validation activities in IA-3.2 in year 2 will focus on internal verification only.

Chapter 3

Conceptual model

In this chapter, we present the conceptual model for the Integrated Research Framework. This model defines all the components that define the IRF and the relations and links among them. We will also illustrate all these components with one example.

3.1 Overall Structure of the IRF Conceptual Model

From a high-level point of view, the IRF consists of eight components, which are clustered in four blocks — see also Figure 3.1, which also defines the main relations existing between the eight components.

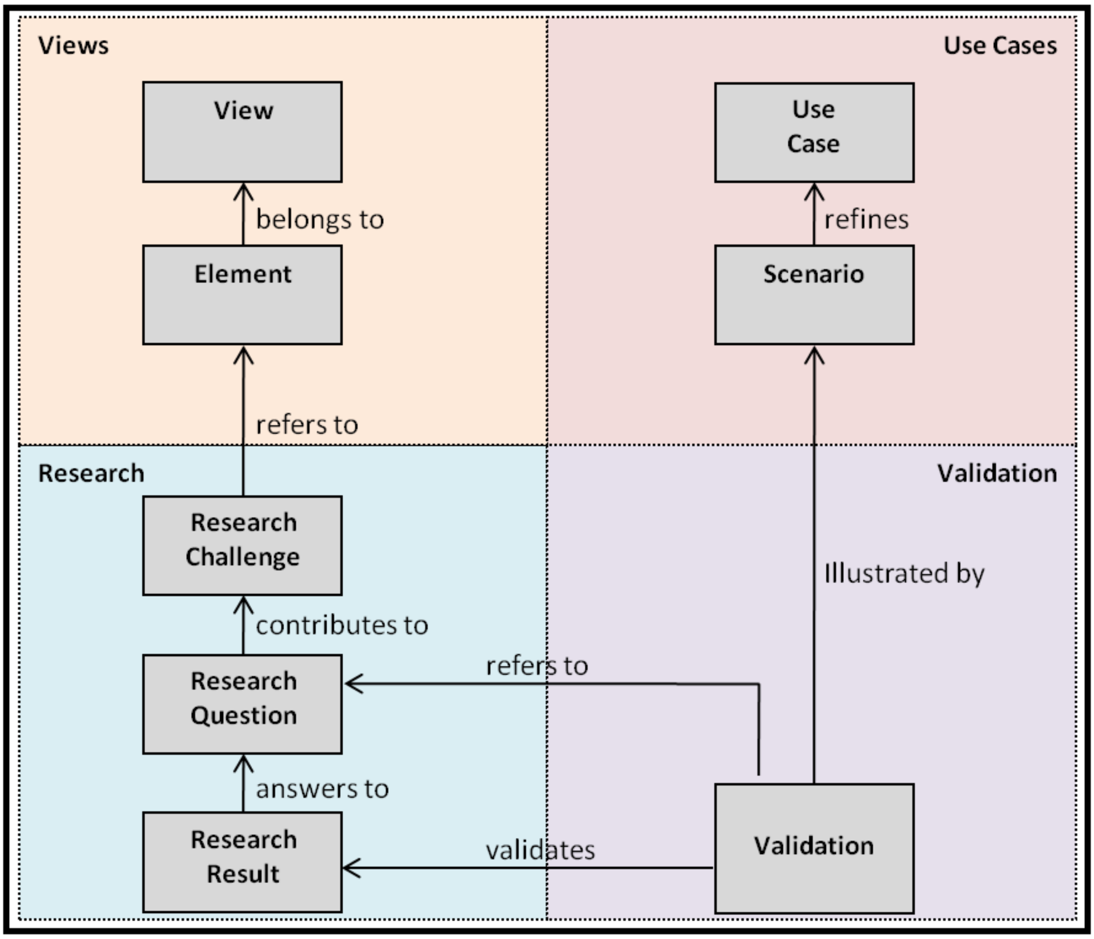


Figure 3.1: High-level structure of the IRF.

- The first block contains the description of the four *Views* corresponding to the IRF baseline defined in deliverable CD-IA-3.1.1, namely the “Conceptual Research Framework”, the “Reference Life-Cycle”, the ”Logical Run-Time Architecture”, and the ”Logical Design Environment” [5]. It also contains the description of the *Elements* of these views, e.g., the blocks that define the Conceptual Research Framework and the arrows defining the relations among them, or the phases in the Reference Life-Cycle, or also the modules of the Logical Run-Time Architecture. This is the most stable part of the IRF, and serves as a reference for the other components of the IRF.

- The second block corresponds to the *Research* undertaken or promoted by S-Cube. It is structured in three components, namely the *Research challenges*, which capture the long term, high-level scientific problems investigated by S-Cube, the *Research questions*, which define the specific, shorter term problems addressed by S-Cube to answer to the challenges, and the *Research results*, which are the answers produced or proposed by S-Cube for the research questions. While challenges will be relatively stable, new research questions and results will emerge during the whole life of the project, according to the progress of the research. New questions and results will be reported in a collective way by the the research work-packages and by the consortium partners, as part of the research activities.

- The third block describes the *Case Studies* which have been identified in activity IA-2.2 for validating Cube results [4]. This block has two components, namely the case studies themselves and the *Scenarios* associated to these case studies. Case studies describe relevant industrial or application domains for the investigations undertaken in S-Cube, and are used for instance to illustrate the research challenges. Scenarios are instead more detailed and correspond to specific situations in the case studies that are used to illustrate specific aspects (problems, questions, results, and so on) of the research undertaken in S-Cube. While case studies are relatively stable, new scenarios are contributed by partners whenever there is a need of illustrating a specific situation, as well as whenever they are needed for validation purposes.

- The final block describes the *Validation* of research results. It captures all the aspects related to the validation, including: the specific goal and scope of the validation (e.g., scalability, or usability of one specific research result); the scenario that is used in the validation; the set-up and execution of the validation; and the results and outcomes of the validation. Also this part of the IRF will be contributed by all S-Cube partners as part of the research activities undertaken within the research work-packages, and under the supervision of IA-3.2.

In the following sections, we describe in more detail these four blocks, defining the structure of the elements belonging to them. In particular, for each element define a table which describes the list of attributes defining the structure of the element. Some of these attributes are common to all elements — and will appear in italic in the tables — while others are specific to one element — and will appear in bold face in the table. The common attributes are:

- *Name*: the name of the specific entry in the IRF.

- *Synopsis*: a brief summary explaining scope and role of the entry.

- *Authors*: authors of the entry in the IRF.

- *Description*: free text describing the entry; it may coincide with the synopsis or contain a more detailed description.

- *References*: links to papers, web pages, or other resources associated to the entry.

- *Glossary*: list of references to terms in the Knowledge Model defined in activity IA-1.1.

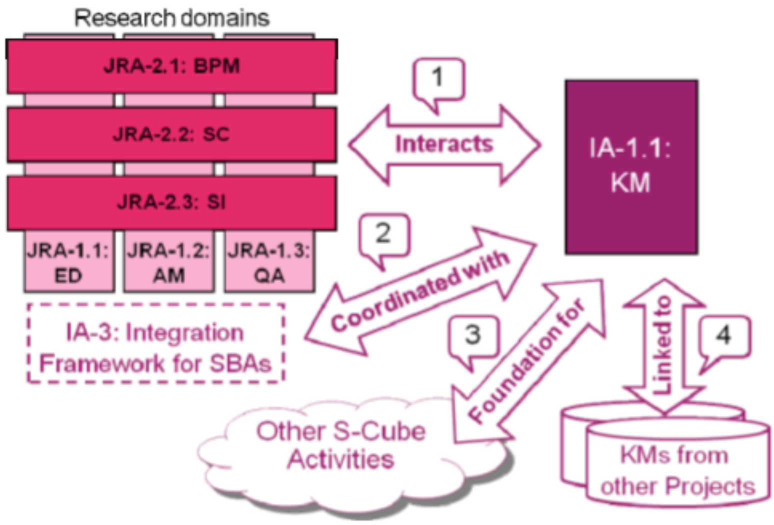


Figure 3.2: Integration of KM and other activities [3]

- *Keywords*: list of keywords to facilitate cataloging and searching the entry.

We remark the importance of field "Glossary", which links the IRF with the Knowledge Model defined by IA-1.1. Indeed, as stated also in the CD-IA-1.1.2 "Separate knowledge models for functional layers" (see Figure 3.2), the definition of the IRF needs to proceed in strict cooperation with the definition of the Knowledge Model, aiming at a continuous alignment between these two key aspects of the integration activities in S-Cube. In the IRF, terms in the glossary are completed with other free "Keywords", i.e., any other terms that do not appear in the Knowledge Model but that are useful to identify and search the entries in the IRF. This field is also a practical way for identifying terms to be evaluated as new entries for the Knowledge Model.

3.2 Views

3.2.1 The "View" component

This component consist of the 4 views that appear in CD-IA-3.1.1, namely: "Conceptual Framework", "Reference Life=Cycle", "Logical Run-Time Architecture", "Logical Design Environment". These part of the IRF is not expected to change during the life of S-Cube.

View	
Name	Name of the view.
Synopsis	Brief summary of the view (1 - 2 sentences).
Authors	List of authors of the view.
Description	Short description of the view.
References	List of link to paper and web page, in which this view was used.
Glossary	List of references to the Knowledge Model.
Keywords	List of keywords to facilitate search.

3.2.2 The "Element" component

This component of the IRF collects the elements of the 4 views: blocks and arrows of the framework, steps in the life-cycle, components of the logical run-time architecture and of the logical design environment, and so on. An initial set of these elements has already been defined in deliverable CD-IA-3.1.1.

Other elements may be added during the progress of the project as a refinement of the IRF.

<table>
<tr><th colspan="3">Element</th></tr>
<tr><td>Name</td><td colspan="2">Name of the element.</td></tr>
<tr><td>Synopsis</td><td colspan="2">Brief summary of the element (1 - 2 sentences).</td></tr>
<tr><td>View</td><td colspan="2">View this element belongs to.</td></tr>
<tr><td>Authors</td><td colspan="2">List of authors of the element.</td></tr>
<tr><td>Description</td><td colspan="2">Short description of the element.</td></tr>
<tr><td rowspan="3">Related elements</td><td colspan="2">List of relations with other elements in the same view.</td></tr>
<tr><td>Element</td><td>Other element in the view that is related to this one.</td></tr>
<tr><td>Relation</td><td>Kind of relation (connected to, belongs to, refines).</td></tr>
<tr><td>References</td><td colspan="2">List of link to paper and web page, in which this element was used.</td></tr>
<tr><td>Glossary</td><td colspan="2">List of references to the Knowledge Model.</td></tr>
<tr><td>Keywords</td><td colspan="2">List of keywords to facilitate search.</td></tr>
</table>

3.3 Research

3.3.1 The ”Research Challenge” component

This component collects the long-term research challenges identifies by the different research work-packages in S-Cube, analyzed and revised as part of the integration activities in order to identify cross-work-package aspects, synergies, and gaps. While new challenges may appear during the project, they are a relatively stable part of the IRF.

Research Challenge	
Name	Name of the challenge.
Synopsis	Brief summary of the challenge (1 - 2 sentences).
Authors	List of authors of the challenge.
Description	Short description of the challenge.
IRF elements	List of elements of the IRF that are relevant for this challenge (NOTE: at least one element of the conceptual framework need to be references).
Related challenges	List of other challenges that are related to this one.
References	List of link to paper and web page, in which this challenge was used.
Glossary	List of references to the Knowledge Model.
Keywords	List of keywords to facilitate search.

3.3.2 The ”Research Question” component

This component collects the open questions and/or needs that need to covered in order to achieve the S-Cube research challenges. Research questions are expected to be extended and refined during the life of the project.

Research Questions	
Name	Name of the research question.
Synopsis	Brief summary of the question (1 - 2 sentences).
Authors	List of authors of the question.
Type	One of: principle, methodology, method, technique, language, notation, mechanism, technology, scientific event, ...
Description	Short description of the question.
Challenges	List of challenges addressed by this question.
IRF elements	List of elements of the IRF that are relevant for this question.
Related questions	List of other questions that are related to this one.
References	List of link to paper and web page, in which this question was used.
Glossary	List of references to the Knowledge Model.
Keywords	List of keywords to facilitate search.

3.3.3 The ”Research Result” component

This component collects the contributions towards the answer to the research questions. There may be more results for the same question, as well as questions that are still open ad have no associated result.

Research Result	
Name	Name of the research result.
Synopsis	Brief summary of the result (1 - 2 sentences).
Authors	List of authors of the result.
Type	One of: principle, methodology, method, technique, language, notation, mechanism, technology.
Research questions	List of research questions that are addressed by this result.
Related research results	Other results that are related to this one.
References	List of link to paper and web page, in which this result was used.
Glossary	References to the Knowledge Model.
Keywords	List of keywords to facilitate search.

3.4 Case studies

3.4.1 The ”Case Study” component

This component will initially contain the 5 case studies already identified in IA-2.2. Further case studies may be added during the life of the project if they contribute new aspects or features that are both very significant for S-Cube and not already covered by the existing case studies. The description of the case studies according to the next table is aligned with the one adopted in CD-IA-2.2.2 ”Collection of industrial best practices, scenarios and business cases”.

Case Study	
Name	Name of the case study.
Synopsis	Brief summary of the case study (1 - 2 sentences).
Authors	List of authors of the case study.
Description	Short description of the case study.
Context	Description of the context of the case study.
Goals	List of goals that are relevant in the case study.
Assumptions	List of assumptions that are relevant in the case study.
Domain analysis	Description of the domain analysis for the case study.
Actors	List of actors that are relevant in the case study.
Abstract scenarios	List of abstract scenarios associated to the case study.
References	List of link to paper and web page, in which this case study was used.
Glossary	References to the Knowledge Model.
Keywords	List of keywords to facilitate search.

3.4.2 The ”Scenario” component

This component collects the scenarios defined to illustrate specific aspects (problems, questions, results, and so on) of the research undertaken in S-Cube. In particular, scenarios are used in the validation: indeed, each validation which is part of the IRF and, in general, of the S-Cube project, will refer to a specific scenario. In S-Cube, each scenario in the IRF need to belong to and refine one of the case studies, and describe typical sequences of activities within this case study.

Scenario		
Name	Name of the scenario.	
Synopsis	Brief summary of the scenario (1 - 2 sentences).	
Authors	List of authors of the scenario.	
Scenario type	One of: context, interaction, internal scenario.	
Abstraction level	One of: instance, type or mixed scenario.	
Scenario usage	One of: positive, negative scenario.	
Description	Short description of the scenario.	
Scenario steps	Description of the interaction (course) of the scenario including messages and events. This interaction can be described verbally (structured English) or in form of UML Sequence or Activity Diagrams.	
Case study	**Id**	Associated case study.
	Abstract scenario	Associated abstract scenario within the case study.
	Actors	List of actors in the case study relevant for this scenario.
	Goals	List of goals of the case study related to this scenario.
IRF elements	List of IRF elements this scenario refers to.	
Challenges	List of challenges for which this scenario is relevant.	
References	List of link to paper and web page, in which this scenario was used.	
Glossary	References to the Knowledge Model.	
Keywords	List of keywords to facilitate search.	

3.5 Validation

3.5.1 The ”Validation” component

This component collects all the validations of the research results undertaken in S-Cube. As already represented graphically in Figure 3.1, this component also bridges between the ”research” part and the ”case studies” part of the IRF: indeed, each validation refer to specific research question and result, but also to a specific scenario in one of the industrial case studies.

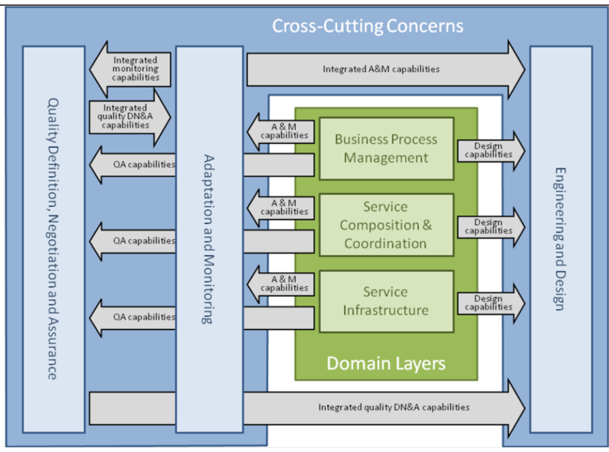
Validation Set-up & Result	
Name	Name of the validation result.
Synopsis	Brief summary of the validation result (1 - 2 sentences).
Authors	List of authors of the validation result.
Research questions	List of research questions addressed by the validation question.
Validation type	One of: usability, correctness, performance, effectiveness...
Scenario	Scenario used in the validation.
Research result	Research result validated.
Method	One of: experiment, case study, field study, prototype, proof.
Description	Short description of the validation as a whole.
Goal	Description of the goal of the validation.
Set-up	Description of the set-up of the validation.
Inputs	Description of the materials (e.g., data) used in the validation.
Outputs	Description of the outputs (e.g., results) of the validation.
Outcome	One of: positive, negative.
Experiences	Comments and experiences on the validation.
References	List of link to paper and web page, in which this result was used.
Glossary	References to the Knowledge Model.
Keywords	List of keywords to facilitate search.

3.6 Example

This section contains an example for each of the tables we just described. A much wider illustration of the content of the IRF can be in the live repository available through the S-Cube portal.

We start by providing an example of the representation of one of the four views, namely the ”Conceptual Research Framework”, and of one of the elements of this view, namely the ”Service Adaptation & Monitoring” block.

View	
Name	Conceptual Research Framework.
Synopsis	The conceptual research framework provides a high-level conceptual view of the S-Cube research activities.
Authors	The S-Cube consortium.

<table>
<tr><td>Description</td><td>


The conceptual research framework is the core element in the definition of the IRF. Its aim is to organise the joint research activities within S-Cube by providing a high-level conceptual architecture for the principles and methods for engineering service-based applications, as well as for the technologies and mechanisms which are used to realize those applications. The framework consists of six components, which are in 1-to-1 relation with the six research work-packages of the network. Moreover, the framework distinguishes between the horizontal components corresponding to the ”traditional” domain layers of a SBA, i.e., ”Service Infrastructure”, ”Service Composition and Coordination”, and ”Business Process Management”, and the vertical components, which correspond to the cross-cutting issues addressed by the project, namely ”Engineering and Design”, ”Adaptation and Monitoring”, and ”Quality Definition, Negotiation and Assurance”. We note that the distinction between the two kinds of components is one of the core elements of the S-Cube approach. Indeed, an element that makes the S-Cube framework unique when compared to the traditional ”layered” approach is that the framework systematically addresses cross-cutting issues. The framework sets out to make explicit the knowledge of the horizontal layers that is relevant for these cross-cutting issues, and that currently is mostly hidden in languages, standards, mechanisms, and so on that are defined and investigated in isolation at the different layers. More precisely, the approach underlying the framework is that the domain layers offer (design, monitoring, adaptation, verification) capabilities that are relevant for the cross-cutting issues. The research efforts in the vertical components are responsible of defining over-arching principles and methodologies for addressing cross-cutting issues by exploiting in suitable ways the capabilities exposed by the horizontal components.
</td></tr>
<tr><td>References</td><td>M. Pistore, R. Kazhamiakin, A. Bucchiarone (Eds.). Integration Framework Baseline, S-Cube Deliverable CD-IA-3.1.1</td></tr>
<tr><td>Glossary</td><td>Adaptable Service Based Application, Service Composition, Service Coordination, Adaptation, Monitoring, Quality of Service Based Applications.</td></tr>
<tr><td>Keywords</td><td>Service Infrastructure, Business Process Management, Design, Engineering.</td></tr>
</table>

Element of a View		
Name	Service Adaptation & Monitoring	
Synopsis	This element of the Conceptual Research Framework comprises research on languages and methods for monitoring and managing the adaptation of a SBA.	
View	Conceptual Research Framework.	
Authors	The S-Cube consortium.	
Description	This element of the Conceptual Research Framework covers the issues related to the adaptation of a SBA. Specifically, this comprises languages and methods for defining adaptation goals and different adaptation strategies, which are triggered by monitoring events. An example for an adaptation technique that falls into the responsibility of this aspect is a strategy that correlates the monitoring events across the functional layers, thereby avoiding conflicting adaptations, or the one that aims to predict the potential SBA problems and perform adaptation activities pro-actively.	
Related elements	**Element**	Integrated A&M capabilities.
	Relation	Provides.
	Element	A&M capabilities.
	Relation	Uses.
	Element	Integrated quality DN&A capabilities.
	Relation	Uses.
	Element	Integrated Monitoring capabilities.
	Relation	Provides.
References	M. Pistore, R. Kazhamiakin, A. Bucchiarone (Eds.). Integration Framework Baseline, S-Cube Deliverable CD-IA-3.1.1	
Glossary	Adaptation, Monitoring	
Keywords		

The next tables contain the description of one of the S-Cube research challenges (”HCI and context aspects in the development of service based applications”), of a related research question (”Exploiting user model knowledge in SBA engineering”), and finally of a research result for this question (”Codified user model knowledge for SBA engineering”).

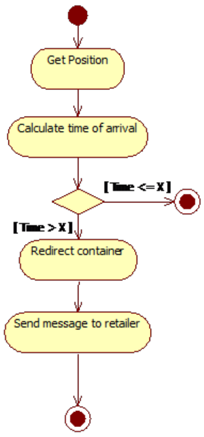
Research Challenge	
Name	HCI and context aspects in the development of service based applications
Synopsis	Understanding how to characterize and codify relevant context and human-computer interaction knowledge for SBA engineering and execution may permit the emergence of new adaptation requirements.
Authors	Neil Maiden, Angela Kounkou (CITY)
Description	Humans are involved in service-oriented computing as end users and consumers, but also as service designers and providers (e.g. Human-Provided Services). A foreseen change in the use and distribution of services, as exemplified in the vision of an upcoming Internet of Services, is expected to further draw humans within the service loop and to promote human-to-application interaction in addition to application-to-application interaction. To this day however, there has been little intersection between research in service-centric systems and Human-Computer Interaction. Human specificities, diversity and tasks characteristics are currently not taken into account for SBA design and delivery. These human properties however could be powerful drivers for SBAs configuration and personalization. An integration of HCI knowledge in SBA engineering is hence necessary for their design and delivery in ways fitting to human use wherever appropriate, for possible enhancements of SBAs existing capabilities, and for the delivery of new capabilities. To this end, the identification of relevant HCI knowledge and its codification for application at design and run time are required. Another important issue touches upon the characterization of the SBAs context in order to enable the identification of adaptation requirements; the observation of the context could guide the adaptation process.
IRF elements	**Conceptual Research Framework**: Service Engineering & Design, Service Adaptation & Monitoring. **Reference Life-Cycle**: Early Requirement Engineering, Requirement Engineering and Design, Construction and Quality Assurance, Deployment and Provisioning, Identify Adaptation Requirements, Identify Adaptation Needs.
Related challenges	Context and HCI aware SBA monitoring and adaptation.
References	PO-JRA-1.1.3 Codified Human-Computer Interaction (HCI) Knowledge and Context Factors.
Glossary	Human Computer Interaction, User Modeling, Task Modeling.
Keywords	-

Research Question	
Name	Exploiting user model knowledge in SBA engineering.
Synopsis	SBA engineering does not currently take into account end users properties such as abilities, needs and preferences. User models, used in HCI to encapsulate this type of information, are investigated for use in SBA engineering.
Authors	Neil Maiden, Angela Kounkou, Kos Zachos (CITY)
Type	Method.
Description	SBA engineering does not currently take into account end users properties such as abilities, needs and preferences. User models, used in HCI to encapsulate this type of information, are investigated for use in SBA engineering.
Challenges	- HCI and context aspects in the development of service based applications - Measuring, controlling, evaluating and improving the life cycle and the related processes.
IRF elements	**Conceptual Research Framework**: Service Engineering & Design. **Reference Life-Cycle**: Early Requirement Engineering, Requirement Engineering and Design, Construction and Quality Assurance.
Related questions	- Identifying relevant HCI knowledge for SBA engineering - Exploiting task model knowledge in SBA engineering
References	PO-JRA-1.1.3 Codified Human-Computer Interaction (HCI) Knowledge and Context Factors
Glossary	Human Computer Interaction, User Modeling
Keywords	-

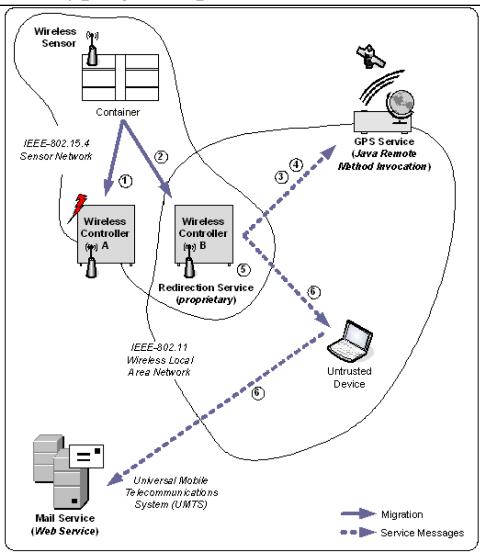
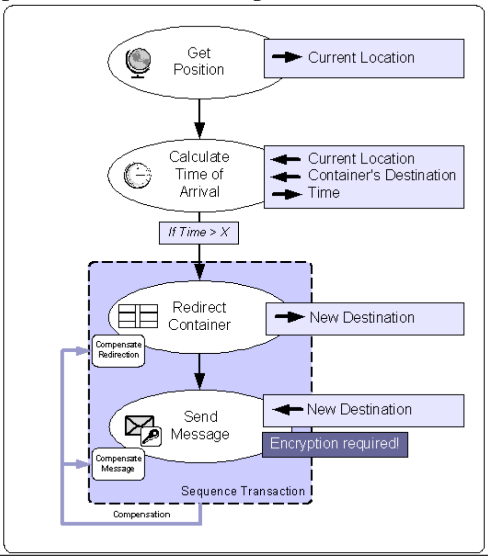
Research Result	
Name	Codified user model knowledge for SBA engineering.
Synopsis	User model data extracted and presented in usable form to inform SBA engineering.
Authors	Angela Kounkou, Neil Maiden, Kos Zachos (CITY)
Type	Technique.
Description	User model data extracted and presented in usable form to inform SBA engineering (more specifically, discovery and selection).
Research questions	Exploiting user model knowledge in SBA engineering.
Related research results	Codified task model knowledge for SBA engineering.
References	PO-JRA-1.1.3 Codified Human-Computer Interaction (HCI) Knowledge and Context Factors CD JRA 1.1.5 Analysis on how to exploit codified HCI and codified context knowledge for SBA engineering.
Glossary	Human Computer Interaction, User Modeling
Keywords	-

The following three tables illustrate a case study, namely ”Wine Production”, a scenario within this case study related to wine transportation, and a validation that uses this scenario against a specific research result related to ”Process Migration”.

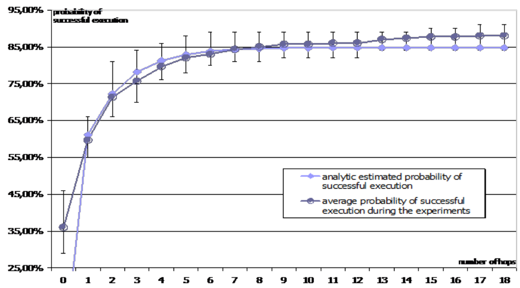
Case Study	
Name	Wine Production.
Synopsis	The case study shows processes involving the growing, harvesting of the grapes and the logistics to deliver the product to retailers.
Authors	Elisabetta Di Nitto (POLIMI)
Description	It involves a Wine Producer who wants to maximize his production in order to adapt it according to the monitored market needs. Other actors of the scenario are the Quality Manager, the Agronomist (i.e., an expert of a branch of agriculture which deals with held-crop production and soil management) and the Oenologist (i.e., an expert of wine and wine production). They have to observe the vineyard parameters and to react to critical conditions that may happen during the cultivation phase. Critical conditions may be represented by overcoming the threshold for some particular environmental parameter.
Context	Wine Producer depends on the Vineyard Operator to maximize sales volume and the wine quality. Vineyard Operators may be agents such as Agronomist, Oenologist, Quality Manager and Wine Grower. Wine Producer depends on the Retailer to stipulate contract and on the Market to Manage Market Needs.
Goals	- BG1: Observe market needs - BG2: Observe vineyard cultivation - BG3: Observe maturation, fermentation and harvesting
Assumptions	- DA1: The system to be should be driven by a self-managing business process - DA2: Vineyard is equipped with a wireless sensor and actuator network - DA3: Time between harvesting and processing should be limited - DA4: Logistic is supported through a RFID system
Domain analysis	The Wine Producer is the owner of the Vineyard, cultivated by the Wine Grower, he stipulates contracts with the Retailer and cares for the Wine Production. The Quality Manager represents an high level Vineyard Operator, while the Agronomist or the Oenologist are considered as specialized operators. The Quality Manager checks the Critical Condition may happen in the Vineyard or related to the Wine Production, and schedules, together with the Agronomist and the Oenologist, the Management Actions needed to maximise the Business Goals. The Delivery Company dispatches the stipulated Orders.
Actors	- Wine Producer - Retailer - Delivery Company - Agronomist - Quality Manager - Oenologist - Market
Abstract scenarios	- Wine Transportation.
References	Di Nitto E., Mazza V., Mocci A. (eds.), Collection of industrial best practices, scenarios and business cases, S-Cube Deliverable CD-IA-2.2.2
Glossary	-
Keywords	Wine Production, Wireless Sensor Networks.

Scenario		
Name	Collaborative Transport Chain Control.	
Synopsis	Reaction to situations in which monitored values differ from the estimated range while transporting the wine.	
Authors	Kunze, C. P.; Zaplata, S.; Turjalei, M.; Lamersdorf, W. (UniHH)	
Scenario type	Internal scenario.	
Abstraction level	Type.	
Scenario usage	Positive.	
Description	Reaction to situations in which monitored values differ from the estimated range while transporting the wine.	
Scenario steps		
Case study	**Id**	Wine Production.
	Abstract scenario	Wine Transportation
	Actors	Quality Manager, Delivery Company, Retailer.
	Goals	- BG3: Observe maturation, fermentation and harvesting - DA1: The system to be should be driven by a self-managing business process - DA4: Logistic is supported through a RFID system
IRF elements	**Conceptual Research Framework**: Service Composition & Coordination, Service Infrastructure, Service Adaptation & Monitoring **Reference Life-Cycle**: Operation & Management **Logical Run-Time Architecture**: Adaptation Engine	
Challenges	Context-aware SBA Monitoring and Adaptation.	
References	Kunze, C.P.; Zaplata, S.; Turjalei, M.; Lamersdorf, W.: Enabling Context-based Cooperation: A Generic Context Model and Management System. In: Abramowicz W, Fensel D (Eds.) 11th International Conference on Business Information Systems, Springer, Berlin: 459-470, 2008	
Glossary	Adaptation, Context Awareness.	
Keywords	Mobile Computing, Transport Chain Control.	

Validation	
Name	Validation of Process Migration.
Synopsis	The validation shows that the probability of the successful executing of service composition can be enhanced using process migration techniques. The result is derived by experimentation with a prototype on the DEMAC platform using a process with one activity and six devices.

Authors	Kunze, C. P.; Zaplata, S.; Turjalei, M.; Lamersdorf, W. (UniHH)
Research question	Recognising changes of context factors.
Validation type	Effectiveness.
Scenario	Collaborative Transport Chain Control.
Research result	Process migration runtime architecture.
Method	Prototyping + Experiments with the Prototype.
Description	 The previous figure shows the exemplary network infrastructure of a transshipment centre for container traffic, where a freezer container is monitored by a wireless sensor. In case of a malfunction of the cooling system, the wireless sensor instantiates a predefined mobile process template which specifies reactions to the detected situation. The resulting process instance is depicted in the following figure: 

Description	The figure shows a selected set of abstract activities and their input/output data: First, the current position of the container has to be acquired (Get Position). Second, the estimated time of arrival has to be calculated in order to decide whether the cargo will thaw until the container arrives (Calculate Time of Arrival). If the time until arrival is considered too long (Time > X), the container must be redirected, e.g. to a cold storage (Redirect Container). Furthermore, a message has to be generated to inform maintenance support where to find the defect container (Send Message). The last two activities of the process are realized as a transaction because the engineer will probably not be able to find the container without knowledge about its new destination. Therefore, the redirection has to be undone if the message cannot be sent within a specified deadline. Furthermore, the message's information must be transferred encrypted — which is attached as a non-functional requirement. Due to performance restrictions, the wireless sensor is not able to execute the process itself. As the process will therefore leave the sensor's sphere of control, it attaches a management descriptor which holds rules about its recovery, monitoring and logging requirements. In this use case, the management descriptor specifies that process execution should be monitored by a backup-device and that, in any case of irregularity, the process should be restarted by this device. Furthermore, process participants, failing devices and recovery actions should be logged and failing devices should be excluded from further process execution. If applicable, a context-based look-ahead procedure should be used to find the most appropriate migration path in order to avoid unnecessary migrations. A possible execution path of the mobile process is shown by the numbered arrows in the figure at the beginning of the description. The wireless sensor is not able to calculate a temporarily optimal execution strategy for the process. Therefore, it migrates the process to an arbitrary other device in its communication range, in this case to wireless Controller A (step 1). But Controller A has a malfunction and is not able to call any other service to execute the process. A timeout indicates its failure and the process is restarted. As this incident is also logged, the failing device is avoided during upcoming migrations and, consequently, in the second attempt Controller B is selected (step 2). As this controller is a stationary and quite powerful device, it is able to call a nearby GPS service to collect data about its current position as well as to calculate the estimated time to arrive at the container's destination (steps 3 and 4). Furthermore, it can decide about the necessity of redirection and uses its own local service to unload the container. However, as it is not connected to the Internet, it has to use an intermediary device to call an appropriate e-mail service. The message is therefore encrypted as described above. Furthermore, as the use of the network (e.g. UMTS) causes telephone charges, participant and payment details can be logged to the mobile process and can be refunded later.

Description	We evaluated the applicability of the generic context model and process management system with a prototype implementation realized in the DEMAC (Distributed Environment for Mobility-Aware Computing) project (cp. Kunze et al. 2008, pp. 467-469). The evaluation includes an experiment to determine the probability of successful execution. The environment for the experimental series consists of a simple process with one single activity, six heterogeneous devices with two devices having the capability to execute the processes' activity, and four devices unable to do so. Because sender and receiver of the mobile process cannot be the same, there are 5 possibilities for each process to migrate from one device to another. This leads to an execution probability of p=40% within the entire system. To test the behavior of the prototype under load, several test runs have to be carried out, each including 100 processes.
Goal	Evaluate the effectiveness of process migration in the mobile computing domain.
Set-up	The prototypical evaluation was executed with a prototype of DEMAC (Distributed Environment for Mobility-Aware Computing) mobile process engine and context management system, a middleware which realizes the concept of process migration.
Inputs	- DEMAC (Distributed Environment for Mobility-Aware Computing) Platform - Simple process with one single activity, six heterogeneous devices with two devices having the capability to execute the processes' activity, and four devices unable to do so.
Outputs	 The previous figure shows the average number of hops resulting from migrations necessary to execute the process successfully compared to the expected analytical value. The analysis of the experiments further shows that only a few hops suffice to increase the probability of successful execution to levels more than twice as high. The estimated probability and the applicability of the presented concept can therefore also be confirmed by practical experimentation.
Outcome	Positive.
Experiences	Not reported.
References	Kunze, C.P.; Zaplata, S.; Turjalei, M.; Lamersdorf, W.: *Enabling Context-based Cooperation: A Generic Context Model and Management System.* In: Abramowicz W, Fensel D (Eds.) 11th International Conference on Business Information Systems, Springer, Berlin: 459-470, 2008.
Glossary	-
Keywords	Mobile Computing, Transport Logistics, DEMAC.

Chapter 4

Implementation

As a second main achievement of the activities done in the IA-3.1 so far, we have developed a Web application based on the conceptual model discussed above. This application, accessible through the Web portal, had been used by the partners to make public the research done so far in the Joint Research Activities (JRA-1 and JRA-2)[1].

As explicitly stated in the S-Cube Deescription of Work (DOW) [1], this application is included in the project Web portal and it will be made available in the future, to external audience. At this stage, we decide to restrict the access to the S-Cube partner to make possible a proper validation of both the conceptual model and the software. It is worth noting that this application will be available during the whole duration of the project, so that the IRF can be continuously updated.

Focusing on the technical details, the application relies on the following technologies and tools:

- Apache HTTP Server.
- MySQL as the Relational Database Management System (RDBMS).
- PHP as the scripting language to develop the interaction layer with the DB.
- HTML and Javascript for developing the application layout.

At the moment (month 21), the application covers the blocks of the IRF corresponding to "Views" and "Research". The blocks for "Case Studies" and "Validation" are now under implementation and will be made available by M24. Also, at the moment, access to the functions of the system is only allowed to registered users authenticated by using login and password. In a near future, to make the application more usable, we plan to integrate this authentication system to the authentication system already adopted on the S-Cube portal.

In the rest of this chapter, we give an overview of the main elements composing the application design. Section 4.1 introduces the application data model, i.e., the database that implements the conceptual model descrived in the previous chapter. Then, Section 4.2 introduces the user interface design.

4.1 Application Data Model

As a first step for developing the IRF web application, we need to map the IRF conceptual model into a relational data model. In this way, the application will rely on a relational data base, to make possible the storage and, a successive retrieval, of data related to the IRF.

The IRF information have been modeled as a Data Model, where the information is represented in a form of tables, and use a DBMS (Database Management System) in order to :

[1]Chapter 5 gives a summary of the research-related information inserted at the time this document is delivered.

- perform ad hoc queries.
- have an independence of data from logical application.
- have a variety of front-end graphical user interface (GUI) tool.
- have easily many view of the same data.

To formalize the conceptual schema into a database schema we adopted the typical rules:

- each view of the conceptual model has been transformed into a table, rows in the tables are uniquely identified by an “Id”, managed by the DBMS (Data Base Management System).
- Many-to-many relations have been mapped into an associative table, which includes the combination of the “primary keys” indentifying the associated tables.

Figures 4.1- 4.4 show the class diagrams corresponding to the main elements of the conceptual model as they are implemented in the database [2]. More precisely:

- Figure 4.1 represents the ‘Skeleton’ part.
- Figure 4.2 represents the ‘Research’ part.
- Figure 4.3 represents the ‘Case Study’ part.
- Figure 4.4 represents the ‘Validation’ part.

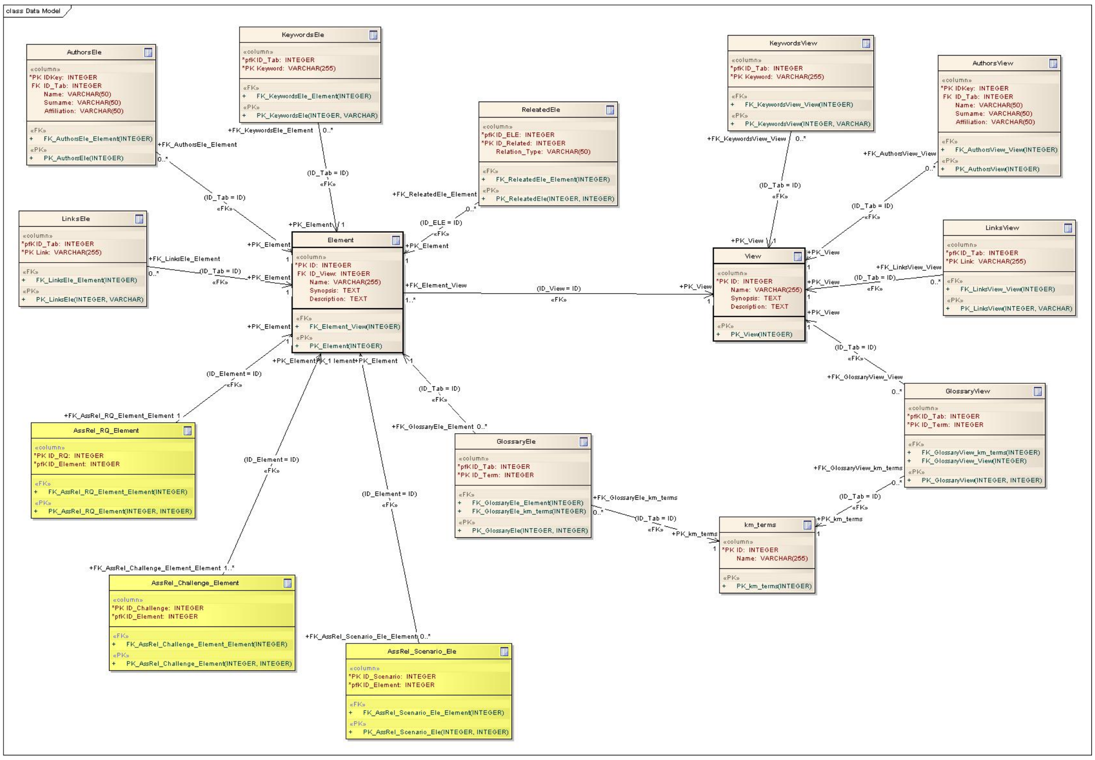


Figure 4.1: IRF skeleton data model.

[2]These class diagrams adopt the semantic defined in [2]

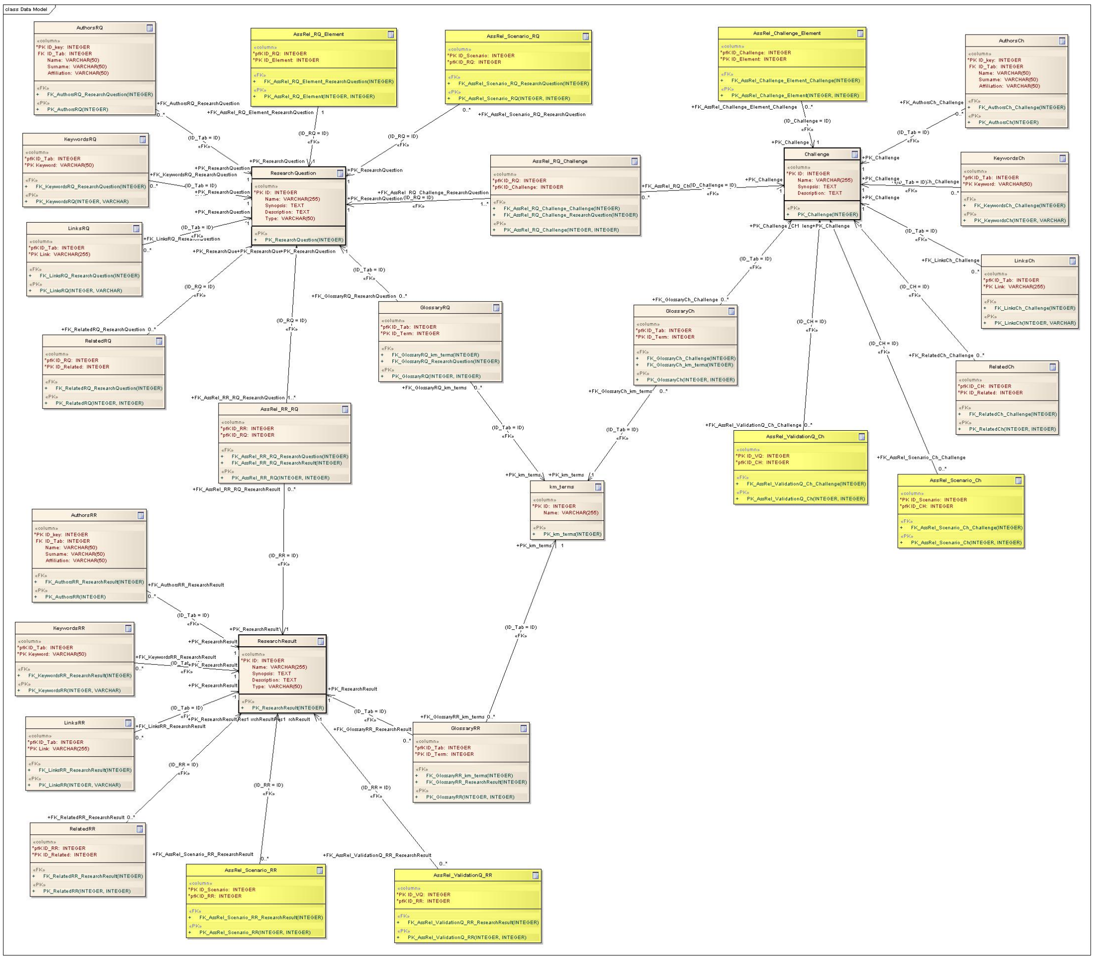


Figure 4.2: IRF research data model.

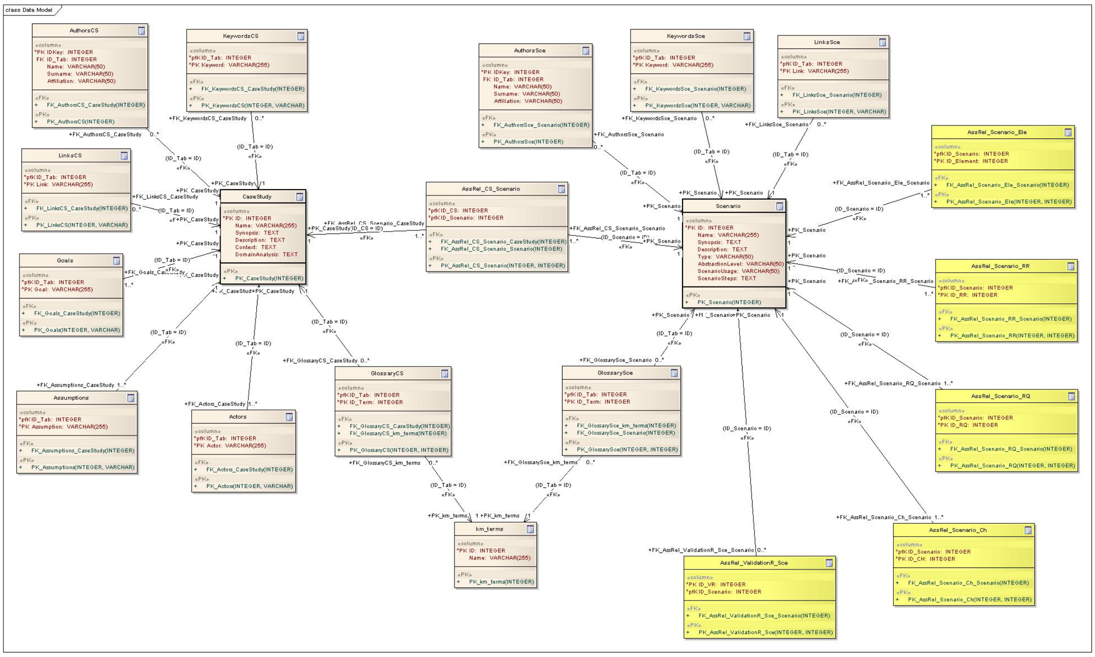


Figure 4.3: IRF case study data model.

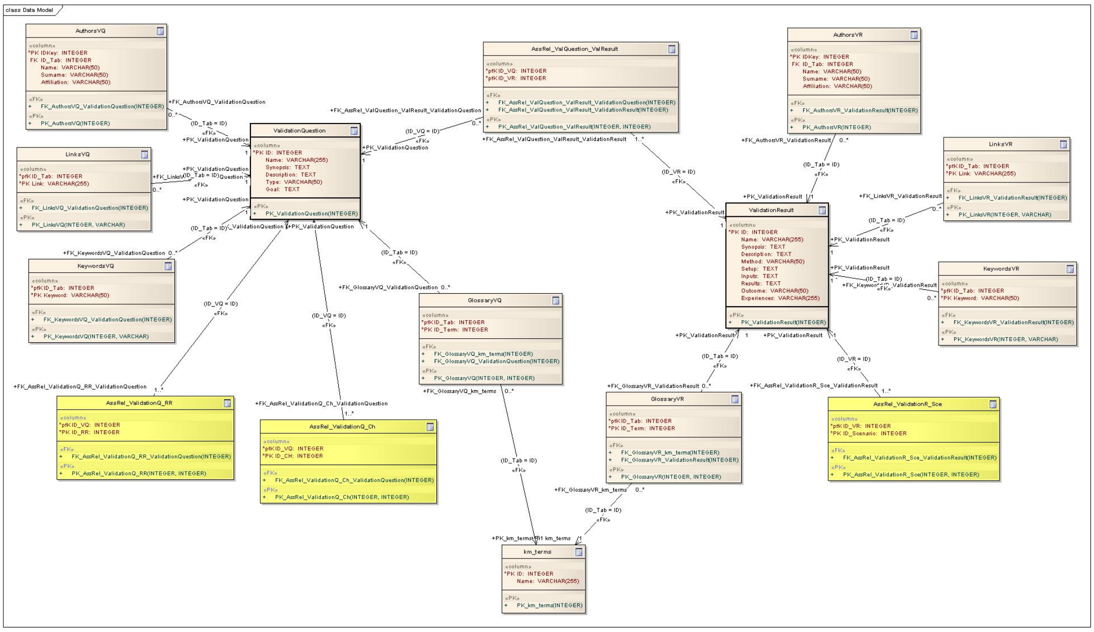


Figure 4.4: IRF validation data model.

4.2 User Interface

An authenticated user, by means of the web application, is able to input, browse, and update the IRF according to the following options that are viewed in the home page, grouped into three main functionalities:

- Create
  - Create a new View
  - Create a new Element
  - Create a new Challenge
  - Create a new RequestQuestion
  - Create a new ResearchResult
- List
  - List Views
  - List Elements
  - List Challenges
  - List ResearchQuestions
  - List ResearchResults
- Update
  - Update a View
  - Update an Element
  - Update a Challenge
  - Update a ResearchQuestion

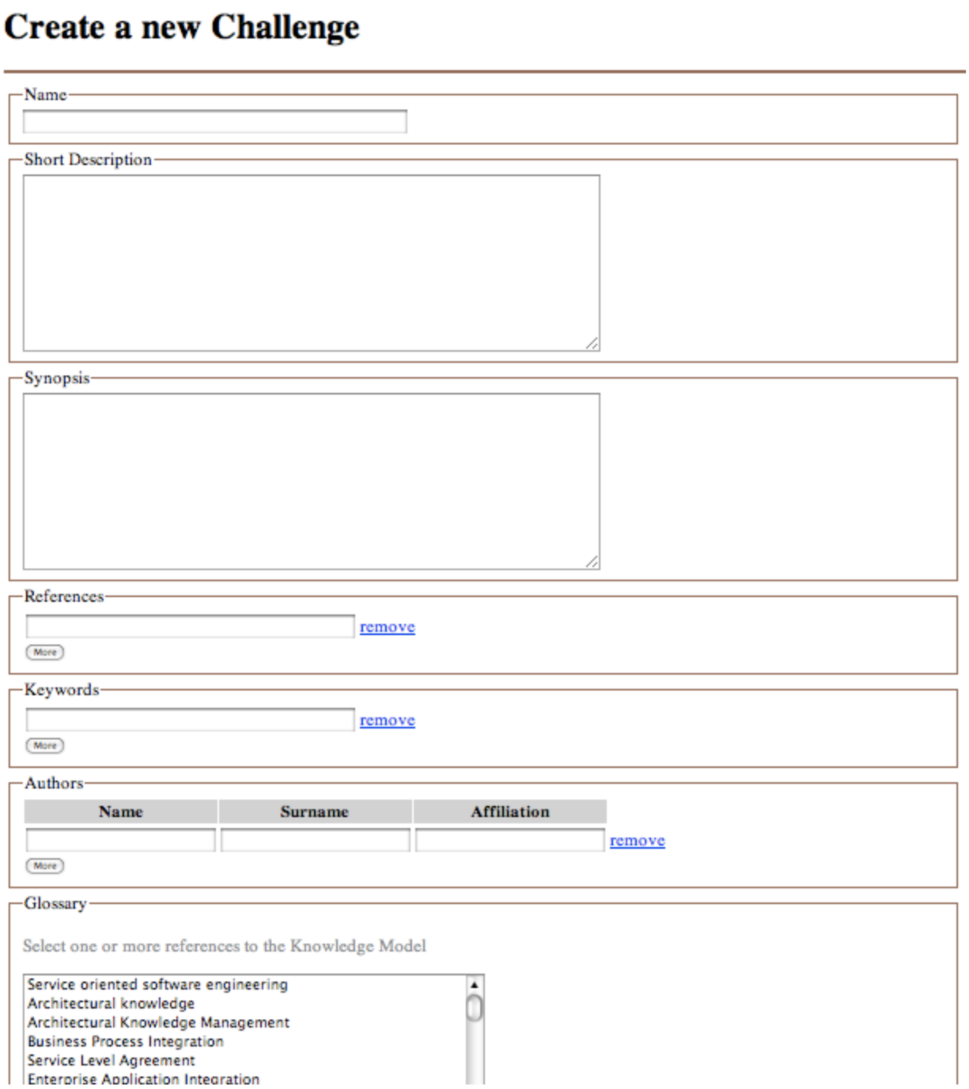


Figure 4.5: Portion of the page used to insert a new research challenge.

– Update a ResearchResult

About the first set of functionalities, i.e., 'New', the application shows a form through which the user can input all the information required to properly describe a specific element of the IRF. Figure 4.5 shows a portion of the form used to insert information about a new research challenge. Here, the 'references', the 'keywords', as well as the 'author' are multivalue fields where the user can insert free text. In addition, the field 'glossary' needs to be feed by a selection of one of more elements from the list shown in the form.

The same kind of form, pre-filled with the data already saved in the database, is used for updating the existing elements of the IRF (see Figure 4.6).

Finally, whenever the user wants to browse one of the existing IRF elements, the 'View' functionalities firstly show a table listing all the instances of the desired element (see Figure 4.7.a), then a table including all information about that element is shown (see Figure 4.7.b).

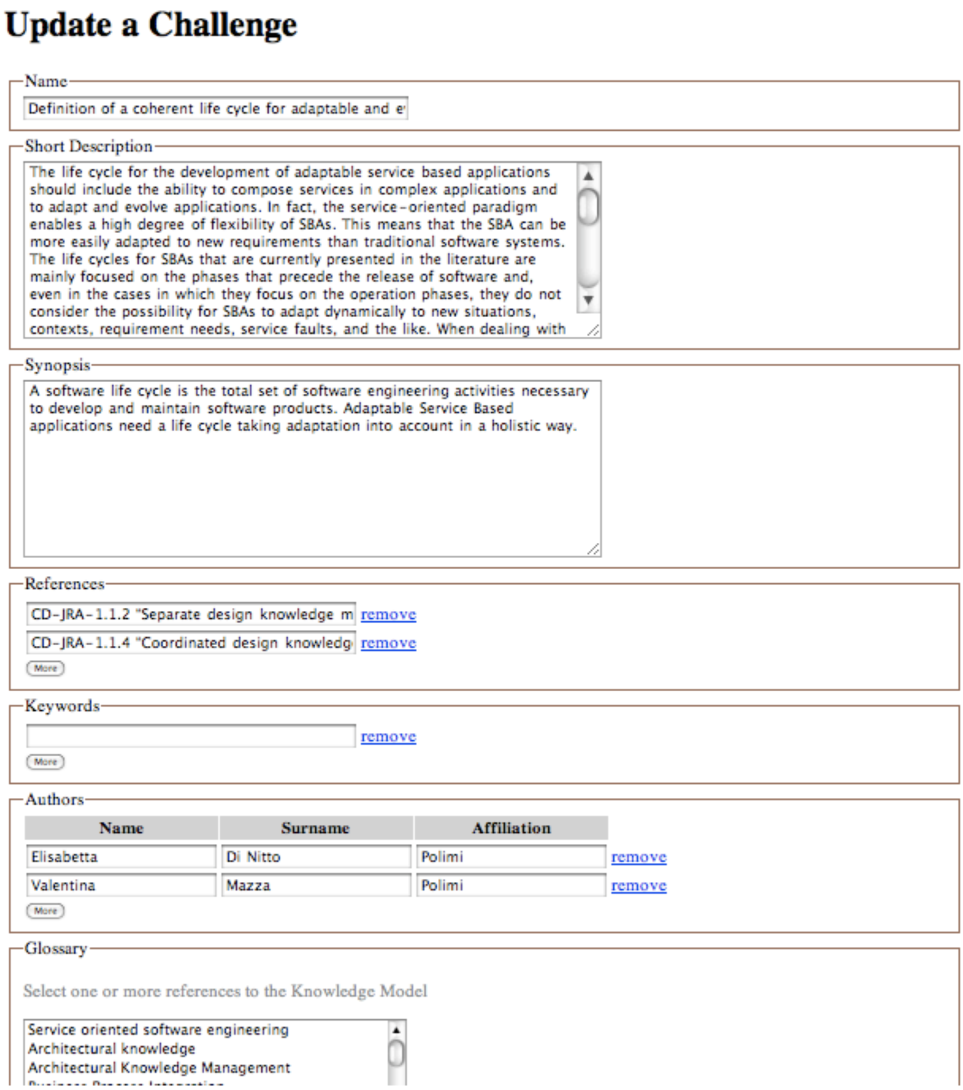


Figure 4.6: Portion of the page used to update a research challenge.

List Challenges

Name	Synopsis	Choose
Definition of a coherent life cycle for adaptable and evolvable SBA	A software life cycle is the total set of software engineering activities necessary to develop and maintain software products. Adaptable Service Based applications need a life cycle taking adaptation into account in a holistic way.	○

(a)

Definition of a coherent life cycle for adaptable and evolvable SBA Details

Name:

Definition of a coherent life cycle for adaptable and evolvable SBA

Description:

The life cycle for the development of adaptable service based applications should include the ability to compose services in complex applications and to adapt and evolve applications. In fact, the service-oriented paradigm enables a high degree of flexibility of SBAs. This means that the SBA can be more easily adapted to new requirements than traditional software systems. The life cycles for SBAs that are currently presented in the literature are mainly focused on the phases that precede the release of software and, even in the cases in which they focus on the operation phases, they do not consider the possibility for SBAs to adapt dynamically to new situations, contexts, requirement needs, service faults, and the like. When dealing with adaptation, on the one side, the requirements engineering phase can be shortened to enhance the time-to-market of the SBA as the missing or misunderstood requirements can later be implemented through adaptation of the running SBA. On the other side, the application has to be designed and developed in such a way that it is able to recognize an adaptation need and to act accordingly. Indeed, not only the application-specific requirements have to be elicited and addressed in the resulting implementation, but also the requirements for adaptation needs to be identified and have to result in a corresponding implementation.

Synopsis:

A software life cycle is the total set of software engineering activities necessary to develop and maintain software products. Adaptable Service Based applications need a life cycle taking adaptation into account in a holistic way.

References:

- CD-JRA-1.1.2 "Separate design knowledge models for software engineering and service based computing."
- CD-JRA-1.1.4 "Coordinated design knowledge models for software engineering and service-based computing."

Authors of the Challenge:

Elisabetta	Di Nitto	Polimi
Valentina	Mazza	Polimi

(b)

Figure 4.7: Portion of the page used to a) view the the list or the b) details of existing research challenges.

Chapter 5

Current contents of IRF

Based on the framework presented in Chapter 2, we have collected a first set of entities of the IRF. For the moment, this collection has covered only the Views and Research components of the IRF. Contributions to the Case Studies and the Validation parts are being collected as part of the activities of WP-IA-3.2.

As a first result of this effort, we have the following situation:

- 4 views;
- 31 elements of the views;
- 24 research challenges;
- 79 research questions;
- 56 research results.

We note that, for this first definition of the IRF, most of the effort has been devoted to the collection of the entities that define the IRF, both in terms of methodology, conceptual model, and web application, and in terms of actual entries that have been submitted by the partners to be part of the IRF. Moreover, for practical reasons, the research work-packages have been responsible of coordinating this collection (see Figure 5.1 for an analysis of the distribution of research challenges/questions/results among the different research activities).

	Research Challenges	Research Questions	Research Results		Research Challenges	Research Questions	Research Results
JRA-1.1	5	21	12	JRA-2.1	2	5	4
JRA-1.2	4	15	10	JRA-2.2	4	5	3
JRA-1.3	6	25	17	JRA-2.3	3	8	10
JRA 1	**15**	**61**	**39**	**JRA 2**	**9**	**18**	**17**

Figure 5.1: Numbers of research challenges/questions/results wrt the project research activities

Due to the bottom-up approach followed in this first effort, there is still some heterogeneity in the description of the collected research questions and results; this is also witnessed by the high variability among work-packages of the numbers reported in Figure 5.1. Moreover, a first analysis of the submitted entries reveals some overlaps as well as some possible synergies.

All these considerations will be further analyzed as part of the activities for the validation of the research framework (IA-3.2). The recommendations and guidelines coming out from this validation will help improving the quality of the IRF along the life of the project.

Chapter 6

Concluding remarks

The main objective of this deliverable is to describe the first version of the IRF that has been produced as outcome of the investigation of IA-3 during the first 21 months work. This objective is fullfilled by the definition of a conceptual model and by the development of a web application to properly describe and organize the research effort done in the S-Cube project.

Starting from the baseline of the IRF that provided a set of views corresponding to different perspectives to the IRF, we defined a conceptual model that includes the description of the research challenges, the research questions, and the research results. To maintain a strict connection to the other activities in the project, the model also includes references to the terms in the Knowledge Model (IA-1.1), to the Industrial Case Studies (IA-2.2), and to the Validation Framework (IA-3.2).

Based on this conceptual model, we also developed a web application to used by the project partners to insert the information about the research done in S-Cube. By accessing to this application, included in the S-Cube Web portal, it is possible to know which are the relevant areas of interest, how the research is conducted, and the results achieved so far.

As a future step, the gathered information needs to be properly validated. Such a validation will need to realize if there are any gaps and overlaps among the research questions and the research challenges. Moreover, we will identify if there are some research questions that do not have any results so far, or some results that are not yet validated. In this way, the project partners can have a comprehensive overview of the direction taken by the research activities.

Bibliography

[1] S-Cube Description of Work. `http://www.s-cube-network.eu`, 2007.

[2] Davor Gornik. UML Data Modeling Profile. Rational Software White Paper, 2002.

[3] S-Cube Consortium. CD-IA-1.1.2 - Separate knowledge models for functional layers, March 2009.

[4] S-Cube Consortium. CD-IA-2.2.2 - Collection of industrial best practices, scenarios and business cases, March 2009.

[5] S-Cube Consortium. CD-IA-3.1.1 - Integration Framework Baseline, March 2009.

[6] S-Cube Consortium. CD-IA-3.2.1 - Initial Definition of Validation Scenarios, October 2009.

[7] S-Cube Consortium. CD-IA-3.2.2 - Results of the First Validation (upcoming), 2010.