Grant Agreement N° 215483

Title:	*Results of the First Validation*
Authors:	*UniDue, FBK, UPM, POLIMI*
Editor:	*Raman Kazhamiakin (FBK)*
Reviewers:	*Zsolt Nemeth (SZTAKI)*
	Rafique Haque (Tilburg)
Identifier:	*CD-IA-3.2.2*
Type:	*Deliverable*
Version:	*1.0*
Date:	*15 March 2010*
Status:	*Final*
Class:	*External*

Management Summary

In this deliverable we present the vision and strategy of the workpackage WP-IA-3.2. This strategy foresees a rigorous method to validate the Integrated Research Framework (IRF). This validation is split into two aspects: an internal verification and an external validation. Given the fact that this is the first validation round, this deliverable concentrates on the internal verification. Therefore, the objectives of the workpackage specified in the description of work are broken down into goals, which are operationalized as questions. Each question is further refined as metrics, which is used to query the IRF database. Based on these quantitative data, we derive a set of recommendations for future work on the IRF and in the S-Cube project.

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).

Members of the S-Cube consortium:

University of Duisburg-Essen (Coordinator)	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
Vrije Universiteit Amsterdam	Netherlands

Published S-Cube documents

All public S-Cube deliverables are available from the S-Cube Web Portal at the following URL:

http://www.s-cube-network.eu/results/deliverables/

The S-Cube Deliverable Series

Vision and Objectives of S-Cube

The Software Services and Systems Network (S-Cube) will establish a unified, multidisciplinary, vibrant research community which will enable Europe to lead the software-services revolution, helping shape the software-service based Internet which is the backbone of our future interactive society.

By integrating diverse research communities, S-Cube intends to achieve world-wide scientific excellence in a field that is critical for European competitiveness. S-Cube will accomplish its aims by meeting the following objectives:

- Re-aligning, re-shaping and integrating research agendas of key European players from diverse research areas and by synthesizing and integrating diversified knowledge, thereby establishing a long-lasting foundation for steering research and for achieving innovation at the highest level.
- Inaugurating a Europe-wide common program of education and training for researchers and industry thereby creating a common culture that will have a profound impact on the future of the field.
- Establishing a pro-active mobility plan to enable cross-fertilisation and thereby fostering the integration of research communities and the establishment of a common software services research culture.
- Establishing trust relationships with industry via European Technology Platforms (specifically NESSI) to achieve a catalytic effect in shaping European research, strengthening industrial competitiveness and addressing main societal challenges.
- Defining a broader research vision and perspective that will shape the software-service based Internet of the future and will accelerate economic growth and improve the living conditions of European citizens.

S-Cube will produce an integrated research community of international reputation and acclaim that will help define the future shape of the field of software services which is of critical for European competitiveness. S-Cube will provide service engineering methodologies which facilitate the development, deployment and adjustment of sophisticated hybrid service-based systems that cannot be addressed with today's limited software engineering approaches. S-Cube will further introduce an advanced training program for researchers and practitioners. Finally, S-Cube intends to bring strategic added value to European industry by using industry best-practice models and by implementing research results into pilot business cases and prototype systems.

S-Cube materials are available from URL: http://www.s-cube-network.eu/

Table of Contents

1 Introduction and Workpackage Vision

The aim of the WP-IA-3.2 "Integration Framework: Validation and Personalization" is to consolidate and validate the Integrated Research Framework (IRF) with the help of industrial case studies and different classes of users, in order to continuously support the coordination of the research activities in the S-Cube project. The validation, which will be conducted iteratively in different phases, will provide the specification for revisions and improvements of the Integrated Framework.

In this document, Deliverable CD-IA-3.2.2 "Results of the First Validation" we report the validation of the IRF as a whole. The document is organized as follows: in this section the vision and the roadmap of the workpackage WP-IA-3.2 until the end of the S-Cube project is outlined. This vision is refined into a strategy in Section 2. Section 2 also describes the validation method used here. Section 3 applies the method to validate the IRF as a whole. Important conclusions are summarized in Section 4.

According to the description of work[1], the workpackage WP-IA-3.2 has three major objectives:

- *Validation of the IRF*: The IRF should be validated as a whole with the help of scenarios, case studies and formal means. The goal of this validation is to improve the IRF.
- *Validation of the IRF building blocks*: Research results produced in joint research activities are the core building blocks of the IRF. These research results should be evaluated using standard research methods such as experiments, case studies, prototypes, demonstrators or formal proofs.
- *Customization of the IRF*: The IRF should be tailored according to the different classes of users defined in workpackage WP-IA-3.1.

These three objectives are detailed below.

1.1 Validation of the IRF

The validation goal of the entire the IRF is to deliver a *consistent*, *complete*, *useful* and *communicated* the IRF at the end of the S-Cube project.

By *consistent* we mean that the integrity of the IRF is ensured among all of its elements. It also means that the research results can be integrated in a way to realize a system engineering scenario. Consequently, the inputs and outputs of the research results produced need to be "compatible".

By *complete* we mean that there are no "disconnected" elements in the IRF. For instance, research challenges should always have related research questions at the end of the S-Cube project. In addition, there should be enough research results to realize a system engineering scenario.

By *useful* we mean that the IRF should help the S-Cube partners internally to organize their work towards integration. In particular, the IRF should support six workpackage and two activity leaders of the joint research activities to plan, organize and supervise their work. In addition, the IRF should also be useful to convey the S-Cube vision and its research results within the spreading of excellence activities. In other words, the IRF should be useful as marketing instrument.

By *communicated* we mean that the IRF should have an impact, for instance, on academic or industrial research agendas.

[1] Amendment #2; draft from the 5th of November 2009.

1.2 Validation of the IRF Elements

At the end of the S-Cube project, we would like to know two issues: first, the major research results produced during the project should be validated. This validation should possibly include the use of different validation methods (e. g. a formal proof and an experiment to validate an algorithm). Second, for each research result produced by S-Cube its validation status should be known, e. g. it should be known which results were validated and in the positive case (i.e., the outcome of the research result validation is positive) how strong this validation was. This aspect is very important in guaranteeing the credibility and high quality of the results provided by S-Cube.

Since validation is a resource-intense activity, validation experiences and validation data gained during the execution of the project should be shared between S-Cube partners.

1.3 Customization of the IRF

Finally, the customization has the goal to implement the user patterns defined in the task T-IA-3.1.3. The framework should be enriched with adaptation points, which allows tailoring it for different user types. The aspects related to the IRF customization will be addressed and presented in future deliverables of the WP-IA-3.1.

2 IRF Validation Strategy and Approach

The validation strategy for the IRF is based on the Goal Question Metric (GQM) method proposed by Basili and Rombach [3], which is combined with the process model proposed by Heinrich [4] who proposes to structure a validation along the following eight phases:

1) *Validation Object*: The first step is to determine, which object(s) should be evaluated. This could include processes (e. g. software development life cycles) and products (such as a software artefact, an algorithm, etc.). In our case, the object under evaluation is the entire IRF. The entire IRF will be validated as a product while some of its components (e. g. the life cycle) need to be validated as processes.
2) *Validation Goal*: The next step is the formulation of the goals of the validation. Goals describe the purpose of the validation—in our case the purpose is derived from the goals of WP-IA-3.1 and the overall goals of the S-Cube project. The purpose may be amended with a focus or view (perspective) of the study and the condition under which the validation is carried out. Both aspects are especially relevant for the validation of the IRF elements.
3) *Validation Criteria (Questions in Basili et al.*): The general evaluation goal is refined into evaluation criteria or questions, which allow quantifying the validation goals. Two aspects can be further distinguished here: criteria regarding the properties of the validation object (e. g. physical attributes of products, properties of a process) and quality properties (e. g. reliability of a software system, effectiveness of a process).
4) *Criteria Weighting:* The weighting of the criteria is only of interest if multiple criteria are to be aggregated in a quantitative manner. For the validation of the entire IRF, we will not define explicit weights for each criterion, thus, all criteria have the same weights.
5) *Metrics:* The validation criteria are further refined into metrics. The metrics allow collecting quantitative or qualitative data of the validation object(s).
6) *Method:* Based on the validation goals, the criteria and the metrics, the measurement method will be chosen.
7) *Data Collection*: Given the validation object and the metrics, the data will be collected with the prescribed method.
8) *Data Analysis*: Finally, the collected data are to be analysed and interpreted.

2.1 Validation and Verification of the IRF

Given the vision of WP-IA-3.2 (see Section 1, we can distinguish between validation and verification as follows (cf. Figure 1) :

- *Internal Verification*: The internal verification ensures that the IRF is useful for guiding integrative research in S-Cube. Three issues can be distinguished regarding this internal verification:
  - *Consistency check:* An internal verification of the IRF consistency ensures the overall quality and integrity of the integration framework. E. g., it will be analysed whether the structure of the IRF is consistent or whether the IRF is consistent with the knowledge model (WP-IA-1.1). For instance, the consistency would be violated if the IRF links to terms in the knowledge model, which do not exist (any more).

    The output of this verification is used to *correct* the integration framework.

  - *Gap Analysis:* Together with the verification of the IRF consistency, a gap analysis is performed by analysing the IRF elements themselves. This gap analysis ensures the completeness of the IRF. E. g., this verification will reveal research questions without research results This gap shows that important research results are still missing in the IRF. As a consequence, there are different ways to eliminate this gap – the modification of the research framework itself (as part of the work in WP-IA-3.1), to stimulate mobility in order to close this gap (as part of the work in WP-IA-2.1), or even to set up new collaborations with external bodies.

  - The output of this verification is used to *initiate a modification of the IRF* or *to trigger mobility activities*.

  - *Scenario-based Evaluation of the IRF*: The IRF will be verified with a SBA development scenario, e. g., with one concrete path through the IRF's life cycle. This verification ensures the consistency between the entire IRF and its elements or in other words, the compatibility of the S-Cube research results. For instance, this scenario-based verification will reveal whether the outputs of a requirements engineering technique (research result) can be used as inputs for a design technique later on.

    The output of this verification is used to *initiate a modification of the IRF* or *to trigger mobility activities*.

- *External Validation*: External validation ensures that the integration framework is useful outside the S-Cube project. Here we distinguish between external validation in academia and industry:
  - *External Validation in Academia*: The focus of this validation activity is on estimating how the IRF covers existing research frameworks in the service field and how the IRF can influence academic research agendas.
  - *External Validation in Industry:* The focus here is to estimate how much the S-Cube project and particularly its IRF can influence industrial research agendas. This analysis will be carried out as joint work with workpackage WP-IA-2.2.

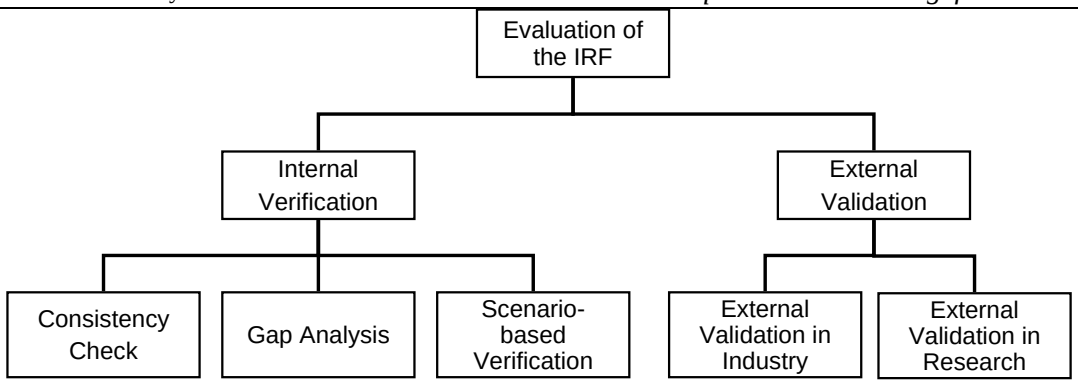

Figure 1: Internal Verification and External Validation (see also [2], p. 8).

2.2 Relation with other Integration Workpackages

For the overall strategy in WP-IA-3.2 it is important to understand the inputs and outputs needed and, therefore, to understand the relations and dependencies with the other integration workpackages. These dependencies are depicted in Figure 2 and include:

- *WP-IA-3.1 (Integration Framework: Baseline and Definition) – WP-IA-.3.2*: The most important relationship of WP-IA-3.2 is the one with WP-IA-3.1 since WP-IA-3.1 provides the main inputs to WP-IA-3.2 in form of the IRF and of its research questions and research results, which are to be validated. In turn, WP-IA-3.2 provides the relevant materials in terms of validation results, which either become part of the IRF (validation of the IRF elements) or lead to an improvement of the IRF (validation of the entire IRF).
- *WP-IA-2.2 (Alignment with European Industry Practices) – WP-IA-3.2*: WP-IA-3.2 uses the industrial case studies from WP-IA-2.2 to derive validation scenarios. These validation scenarios are in turn used for extending/refining the industrial case studies and pilot cases (cf. [1]).
- *WP-IA-2.1 (Mobility of Researchers) – WP-IA-3.2*: Once the first set of validation results is collected and the entire IRF is validated, the results will not only be used to enhance the IRF itself but also to stimulate work in the areas, which are not yet covered. One mean to achieve this coverage is to influence the mobility plan, which is developed in WP-IA-2.1.
- *WP-IA-1.1 (Convergence KnowledgeModel) – WP-IA-3.2*: The knowledge model provides the relevant glossary terms related to the validation results.

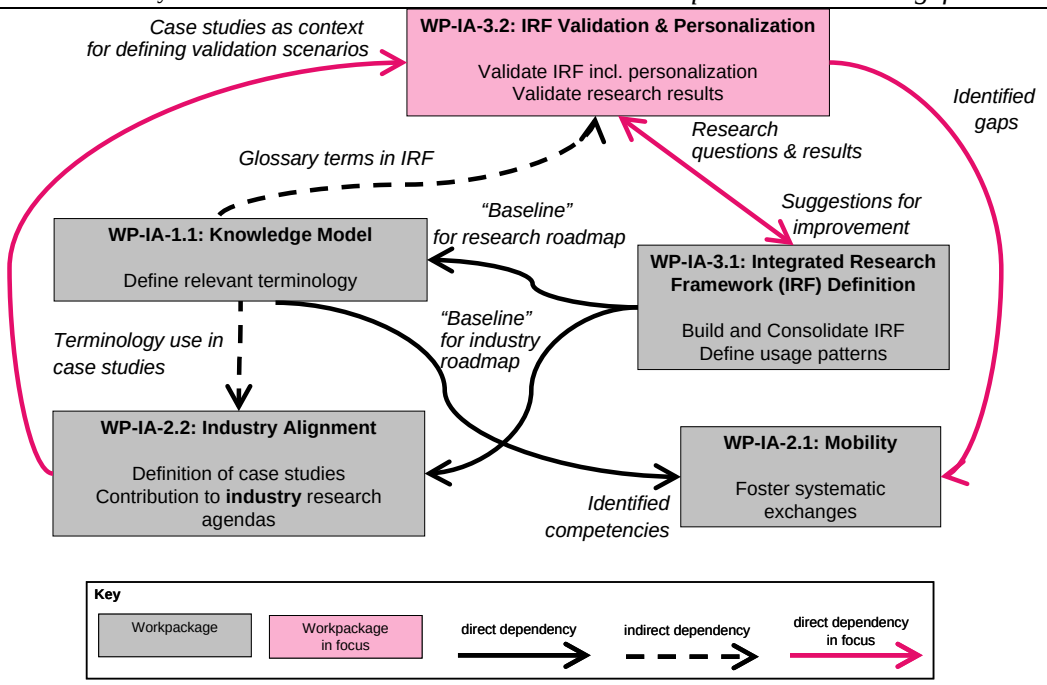

Figure 2: Relation between the Integration Workpackages

2.3 Roadmap and Timeline in IA-3

Since the validation object(s) such as the IRF and its element is produced in WP-IA-3.1, it is important to understand the intertwining with this workpackage as it has a direct impact on the timeline in WP-IA-3.2. This analysis has to consider two tasks in WP-IA-3.1, e. g. T-IA-3.1.2 and T-IA-3.1.3 and three tasks in WP-IA-3.2, e. g. T-IA-3.2.1, T-IA-3.2.2 and T-IA-3.2.3. Regarding those tasks, we can distinguish between the following three types of dependencies (cf. the vision description in Section 1):

1) *IRF validation dependency*: The validation of the IRF is implemented in three different cycles. The first validation of the IRF (deliverable CD-IA-3.2.2; delivery date month 24) is based on the definition level of the IRF (deliverable CD-IA-3.1.3; delivery date month 21). The second validation (deliverables PO-IA-3.2.3 and CD-IA-3.2.4, delivery date: month 36) is based on the consolidated and revised integration framework (deliverable CD-IA-3.1.5, delivery date: month 33). The third and final validation (CD-IA-3.2.5, delivery date: month 48) is based on the final consolidated version of the IRF (deliverable: CD-IA-3.1.7).

   Therefore, the planned time for validating the IRF is three months in each cycle. The remaining time is needed to develop the validation method (such as the validation scenarios in PO-IA-3.2.3) and to influence the integration activities according to the outcome of the validation.

2) *IRF element validation dependency*: Task T-IA-3.2.3 aims to validate all major research results. Since these validation results become part of the integration framework itself, they will be used for instance to determine the validation status of the IRF. Therefore, the validation results of the deliverables PO-IA-3.2.6 and PO-IA-3.2.7 become part of the IRF.

3) *Personalization dependency*: The second dependency between WP-IA-3.1 and WP-IA-3.2 regard the personalization of the IRF. For WP-IA-3.2 this basically means that the workpackage needs to implement the user patterns developed in T-IA-3.1.3 in the IRF, e. g. by extending the IRF with different user types. No additional deliverables are defined in WP-IA-3.2 and the personalization and customization results are reported in the same deliverables as the other validation results.

Deliverable PO-IA-3.1.4 (delivery date: month 27) provides the relevant input for validating the defined user patterns and methodologies (documented in CD-IA-3.2.4, delivery date: month 36). The second and final validation of those user patterns (documented in CD-IA-3.2.5, delivery date: month 48) will be based on deliverable CD-IA-3.1.6 (delivery date: month 39).

The dependencies between WP-IA-3.1 and WP-IA-3.2 are visualized in Figure 3.

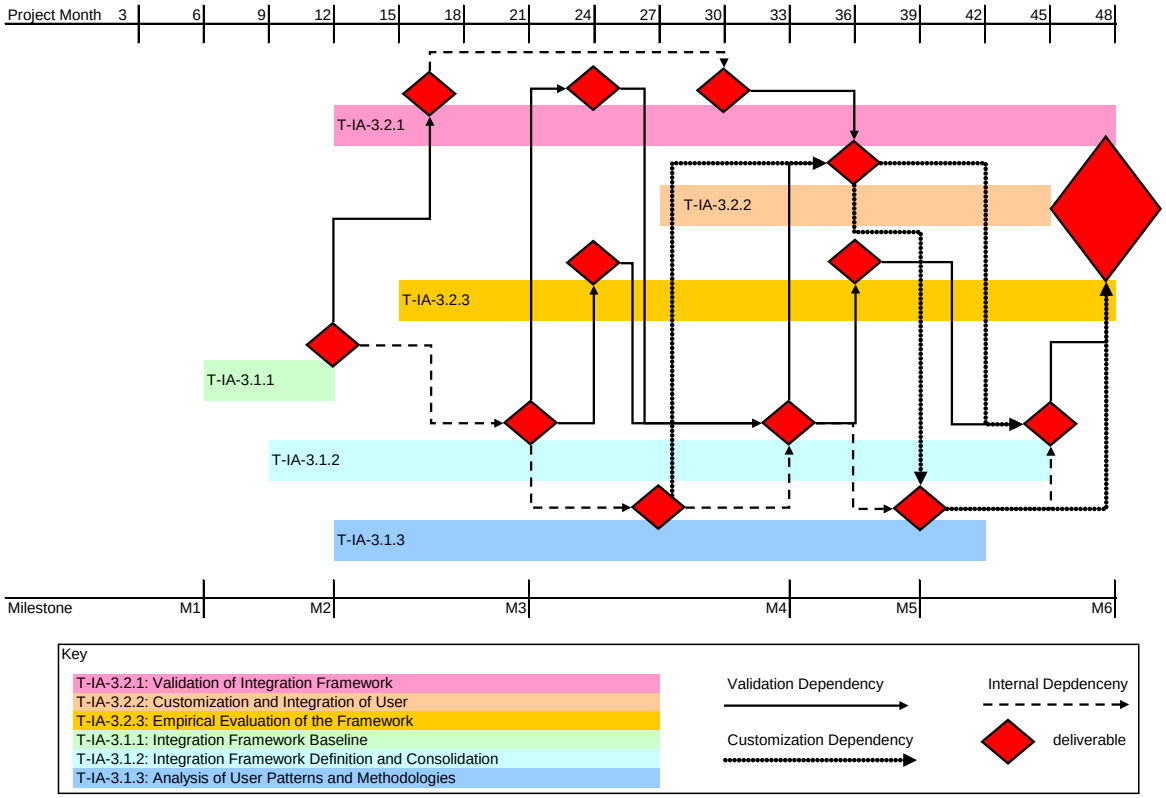

Figure 3: Intertwining between WP-IA-3.1 and WP-IA-3.2

Given the vision and strategy outlined before and the dependencies between the workpackages, the following timeline will be used for WP-IA-3.2 for the years 2–4:

- *Year 2*: Since the first version of the IRF was completed in month 21, the main focus in year 2 is on the consistency check of the IRF. In addition, the validation activities in year 2 will also concentrate on the gap analysis in order to provide input to the mobility program and to the JRAs to coordinate the research in years 3 and 4. Regarding task T-IA-3.2.3, the main focus is on collecting validation results, documenting them in a unique format and making them available via the IRF.
- *Year 3*: In the third year the verification activities will continue and will be extended by the scenario-driven verification. This scenario-driven verification will especially ensure the consistency of the different research results achieved. In addition, validation activities will also start in year 3. In addition, the validation status of the IRF elements will be analysed. This analysis may reveal gaps in the validation, which will in turn trigger validation activities. These validation activities will be executed in close collaboration with the two joint research activities.
- *Year 4*: In the final year the focus will be on external validation in close collaboration with WP-IA-2.2. (industry) and WP-SoE-1.2 (spread of excellence). The internal verification will be limited to those inconsistencies and gaps in the IRF, which will remain after the completion of the S-Cube project since this output cannot influence the S-Cube project anymore. Finally,

Task T-IA-3.2.3 will deliver a report regarding the validation status of the IRF elements. This report will contain the elements, which are validated including the validation results. The report will also contain those elements, which were not yet validated during the S-Cube project.

3 First Evaluation of the IRF

This deliverable contains the first evaluation of the IRF. Given the timeline, strategy and vision outlined above, the deliverable focuses on the internal verification of the IRF, especially on the consistency checks and the gap analysis. More precisely, the remaining deliverable follows the evaluation strategy with respect to the general usage of the IRF within the S-Cube project.

3.1 Validation Object – The IRF

The IRF (see Figure 4), as defined in Deliverable CD-IA-3.1.3 "First version of the Integration Framework", consists of four clusters of logically related elements.

- The "Views" cluster defines the IRF views introduced in the baseline document PO-IA-3.1.1. These views (i.e., Conceptual Research Framework, Life-cycle, Logical Run-time Architecture, and Logical Design Environment) and their elements characterize the key perspectives of the IRF and its reference model.
- The "Research" cluster characterizes the research objective, activities, and results undertaken by the S-Cube project. Here the Research Challenges define the long-term objectives of the research activities, while the Research Questions define the short-term goals addressed by the specific Research Results.
- The "Use Cases" cluster characterizes the case studies defined and described by IA-2.2 as well as the specific scenarios within those use cases to be used for the validation of the research results of the previous cluster.
- Finally the "Validation" cluster aims at capturing the aspects related to the validation of the research results obtained. As it is shown in Figure 4, the validation refers to the research question that is answered by the underlying research result, and is illustrated with the help of a particular validation scenario.

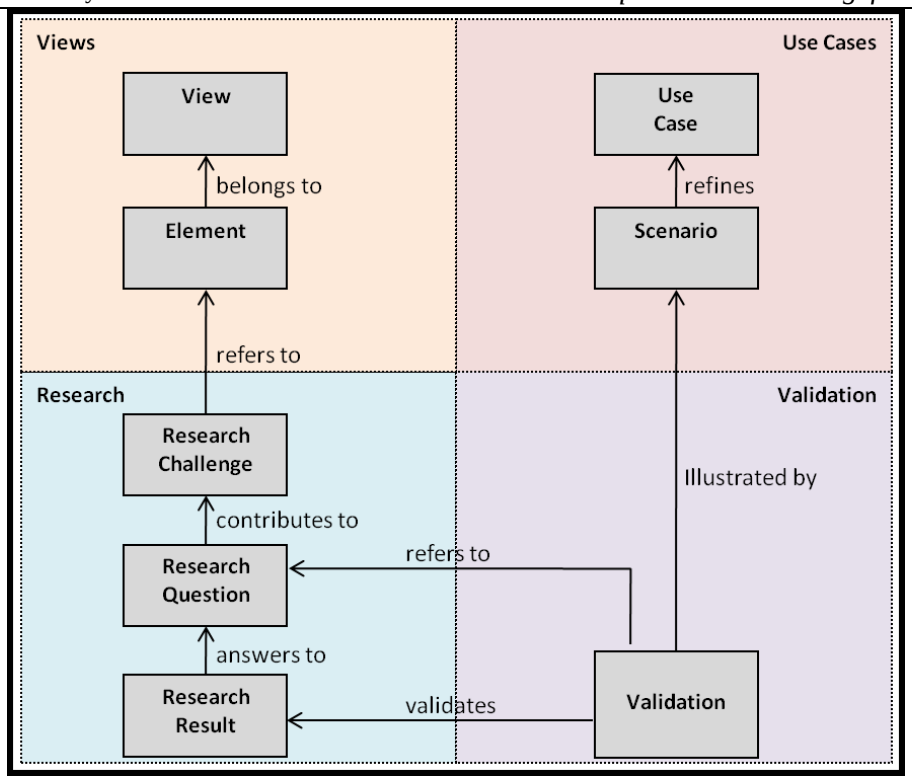

Figure 4: IRF Structure

The validation activities reported in the current document refer, therefore, to the consistency and correctness of the IRF structure, the relations between its elements, as well as the consistency of the specific IRF entries. Note that the IRF validation will also address the issue of the evolution of the IRF. That is, the new entries added to the IRF, how the existing elements are changed, the difference in the dynamics of different clusters, etc. This information is important in order to understand the usefulness of the IRF as a whole, as well as of its specific components; to identify the gaps in the research activities undertaken by the S-Cube.

3.2 Goals

According to the validation issues presented in Section 2.1, the general validation goal is refined into the following sub-goals:

G1 *IRF Complexity*: with this goal we control the complexity of the IRF, especially whether it grows substantially over time. This goal is important since a low IRF complexity bears the risk not to cover important aspects of SBAs while a too complex IRF comes at the risk of low understandability of the IRF.

G2 *Consistency*: The consistency check ensures the integrity of the IRF. Since the IRF is implemented as relational database [2], the consistency checks ensure the formal integrity of the IRF database. The consistency is important since an inconsistent IRF will lead to unpredictable and wrong results when working with the IRF.

G3 *Gap Analysis*: The gap analysis aims to reveal potential incompleteness of the IRF. It identifies elements, which are “left alone” in the IRF such as research challenges that are not refined into research questions. The gap analysis ensures that the unnecessary elements are removed from the IRF and missing elements are added (e. g. by means of research activities).

G4 *Validation Status*: The analysis of the validation status of the IRF shows, which research results are currently validated. This goal is important since valid research results are more credible than the ones that are not validated. In addition, this validation status may also influence the validation work of task T-IA-3.2.3.

G5 *Integration Status*: To demonstrate the interactions between the workpackages, the two joint research activities and to show the integration achieved in S-Cube, the integration status of the IRF is measured.

G6 *Consistency with Knowledge Model:* This analysis shows the consistency between the terminology used in the IRF and the terms defined in the knowledge model. This goal is important since it ensures that the knowledge model is actually used in the project. In addition, it may also reveal inconsistencies or incompleteness of the knowledge model.

3.3 Validation Questions

Given our method, the goals described in Section 3.1 are further refined into quantifiable validation questions (Table 1). In particular, the metrics reflect the validation issues as follows:

- To reflect the complexity of the IRF, we quantitatively measure all different components of the IRF, the relations within and among those components.
- To check the consistency of the IRF, we verify whether the IRF is structurally integrated and whether the relations between elements of the IRF are preserved at different levels. That is, whether the relations among concrete challenges are preserved by the relations between the associated questions (and vice versa), etc.
- To evaluate the completeness of the IRF, we study how the research questions, results, challenges, elements, and validations are distributed. In this way, we can evaluate which problems acquired more attention than the others, which problems are left apart and require further investigations, which results require further validations, and how the case studies and scenarios are involved in the process.
- To evaluate the status of validation, we have defined metrics to continuously track the progress of the S-Cube results validation, as well as how the validation covers different elements of the IRF.
- The set of metrics is also defined for the evaluation of integration status. Those metrics aim to reveal the existing and missing relations between questions, results, and challenges, the relations of those elements across workpackages and across joint research activities.
- To evaluate the consistency with the knowledge model we study both how much the terminology is exploited in the research (i.e., the terms actually used by the research questions, results, and validations) as well as the new terminology identified but not yet reflected in the KM (i.e., through identification of terms and keywords not appearing in the KM).

Table 1: Breakdown of the Goals into Validation Questions

No.	Validation Question	Goal
1	Number of elements per type in the IRF.	G1
2	Total number of elements in the IRF.	G1
3	Total number of all relations in the IRF.	G1
4	Check whether for each pair of related research results belonging to different research questions that there is also a relation between the research questions.	G2
5	Check whether for each pair of related research questions there at least one relation between research results related to those research questions.	G2
6	Do 4) and 5) for the pairs: research question-research challenge and research challenge-element.	G2
7	Number of Elements per View	G3
8	Number of research challenges per element incl. min/max/avg. (visualize as distribution curve)	G3
9	Number of research challenges per view (Aggregate; visualize as distribution curve).	G3
10	Number of research questions per research challenge only min/max/avg (visualize as distribution curve).	G3
11	Number of research questions per element and per view.	G3
12	Number of research results per research question.	G3
13	Number of research results, which belong to more than one view.	G3

14	Number of research results per challenge, element and view.	G3
15	Number of scenarios per case study.	G3
16	Number of validation result per scenario.	G3
17	Number of validation results per case study.	G3
18	Number of validations per research method.	G4
19	Percentage of validated research results and validated research questions.	G4
20	Number of validations performed per research question, research challenge, element and view.	G4
21	Number of relations between elements per element (including min, max & avg).	G5
22	Number of Research challenges, which are related to more than one view.	G5
23	Number of relations between research challenges per challenge (including min, max & avg).	G5
24	Number of research questions related to more than one view.	G5
25	Number of relations between research questions per question (including min, max & avg)	G5
26	Number of relations between research results per result (including min, max &avg).	G5
27	Number of cross-JRA challenges: Number of challenges belonging to more than one element in the view “Conceptual Research Framework” and one element is in the set {Adaptation & Monitoring; Engineering & Design; Quality Definition, Negotiation and Assurance} and another element is in the set {Business Process Management; Service Composition and Coordination; Service Infrastructure}.	G5
28	Number of cross-WP research challenges: Number of research challenges belonging to more than one element of the view “Conceptual Research Framework”.	G5
29	Number of cross-JRA questions: Number of research questions belonging to more than one element in the view “Conceptual Research Framework” and one element is in the set {Adaptation & Monitoring; Engineering & Design; Quality Definition, Negotiation and Assurance} and another element is in the set {Business Process Management; Service Composition and Coordination; Service Infrastructure}.	G5
30	Number of cross-WP research questions: Number of research questions belonging to more than one element of the view “Conceptual Research Framework”	G5
31	Number of cross-JRA research results: Number of research results belonging to more than one element in the view “Conceptual Research Framework” and one element in the set {Adaptation & Monitoring; Engineering & Design; Quality Definition, Negotiation and Assurance} and another element in the set {Business Process Management; Service Composition and Coordination; Service Infrastructure}.	G5
32	Number of cross-WP research results: Number of research results belonging to more than one element of the view “Conceptual Research Framework”.	G5
33	Number of KM terms referenced by the IRF as glossary or keyword terms.	G6
34	Number of keywords not included in the KM	G6

3.4 Data Collection

To answer the validation questions identified in Section 3.3, the corresponding quantitative metrics have been defined and evaluated on the IRF. As soon as the data model of the IRF defined in CD-IA-3.1.3 has been implemented as a relational database management system, the corresponding metrics may be realized as queries over the content of the database. More precisely, for each of the validation questions defined in Section 3.3 one (or more) SQL query has been defined in order to evaluate the corresponding validation question. In Appendix A1 the complete list of SQL queries corresponding to the validation questions is presented.

In the following, we describe the results of these SQL queries according to six different goals described in Section 3.2. This section, however, does not include an analysis of that data. The analysis is presented in Section 3.5.

3.4.1 Goal G1 – IRF Complexity

With respect to the IRF complexity (Goal G1), Table 2 reports the number of IRF elements of different nature, and also number of all relations in the IRF database.

Table 2: IRF complexity metrics (metrics 1, 2, 3)

Number of views	4
Number of Elements	32
Number of Challenges	24
Number of Research Questions	75
Number of Research Results	51
Number of Glossary Terms	140
Number of Keywords	52
Number of Case Studies	5
Number of Scenarios	16
Number of Validation Results	5
Total number of elements	675
Total number of relations	1154

3.4.2 Goal G2 – Consistency

As defined in Section 3.3, for the consistency check we are interested in knowing whether relations between research questions imply relations between research results (and vice versa) and whether relation between research challenges imply relations between research results (and vice versa; Metrics 4, 5 and 6). Table 3 lists the number of missing relations.

Table 3: Consistency Results (Metrics 4, 5, 6)

No.	Metric	Number of missing pairs
4	Missing relation between research questions which belong to related results.	3
5	Missing relations between research results which belong to related research questions.	17
6/1	Missing relations between research challenges which belong to related research questions.	14
6/2	Missing relations between research questions which belong to related challenges.	32
6/3	Missing relations between elements which belong to related challenges.	32
6/4	Missing relations between challenges which belong to related elements.	11

3.4.3 Goal G3 – Gap Analysis

In order to see the complexity of the IRF and to cross-check the plausibility of the more detailed results described below, Figure 5 depicts the number of all elements in the IRF together with the overall number of relations between those elements. It reflects the overall structure of the IRF, shows how it is fulfilled, and defines the relation of different components if the IRF with the information from Convergence Knowledge Model, in particular with the KM terms. The picture shows in quantitative manner also the relations between the components of the IRF, such as challenges referring to other challenges or to elements, results associated to research questions and to other results, etc.

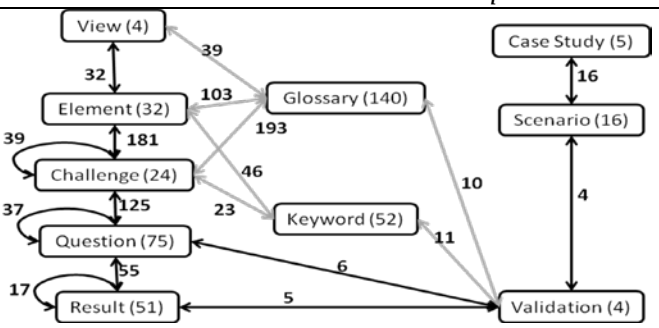

Figure 5: Relations between Elements (metrics 7, 8, 10, 12, 20, 15, 16)

Figure 6 shows the distribution of elements, challenges, research questions and results per view with average values of 8 elements per view, 17.25 challenges per view, 53 research questions per view, and 13 results per view respectively.

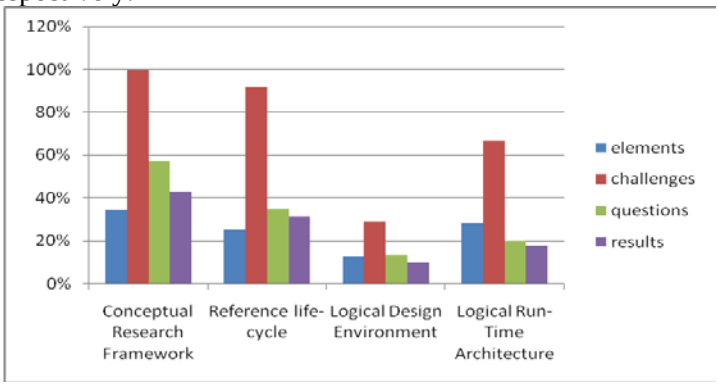

Figure 6: Distribution of Elements, Challenges, Research Questions and Results per View (metrics 7, 9, 11, 14)

Figure 7 shows the distribution of challenges, research questions and results per element with average values of 5.9 challenges per element, 6.7 research questions per element, and 3.5 research results per element respectively.

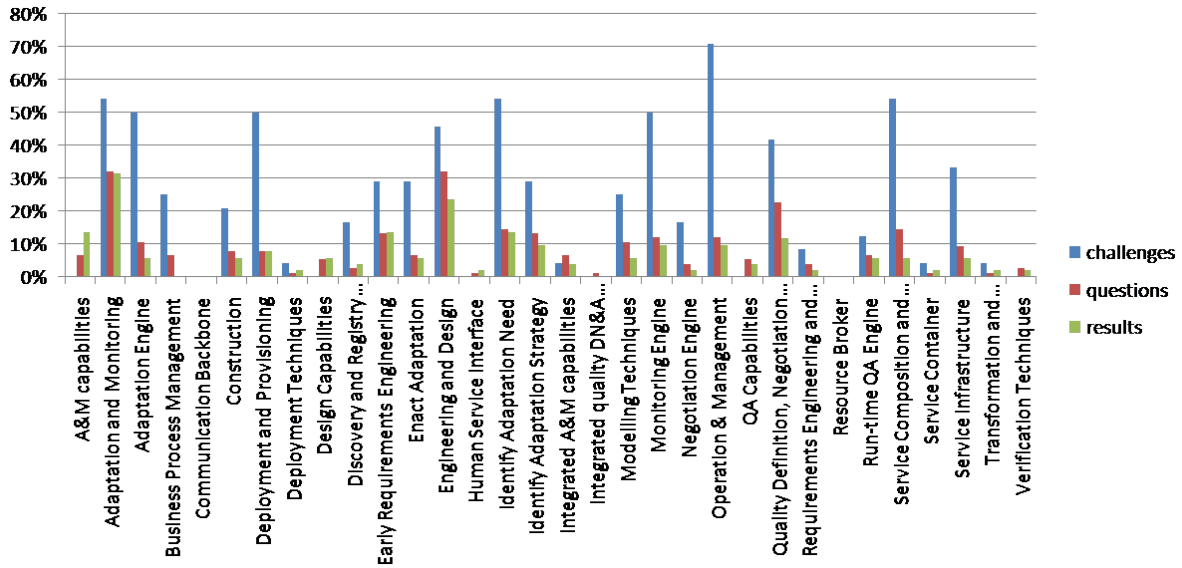

Figure 7: Distribution of Challenges, Research Questions and Results per Element (Metrics 8, 11, 14)

Figure 8 shows the distribution of the research questions and results per challenge with average values of 5.2 research questions per challenge and 3.7 research results per challenge respectively.

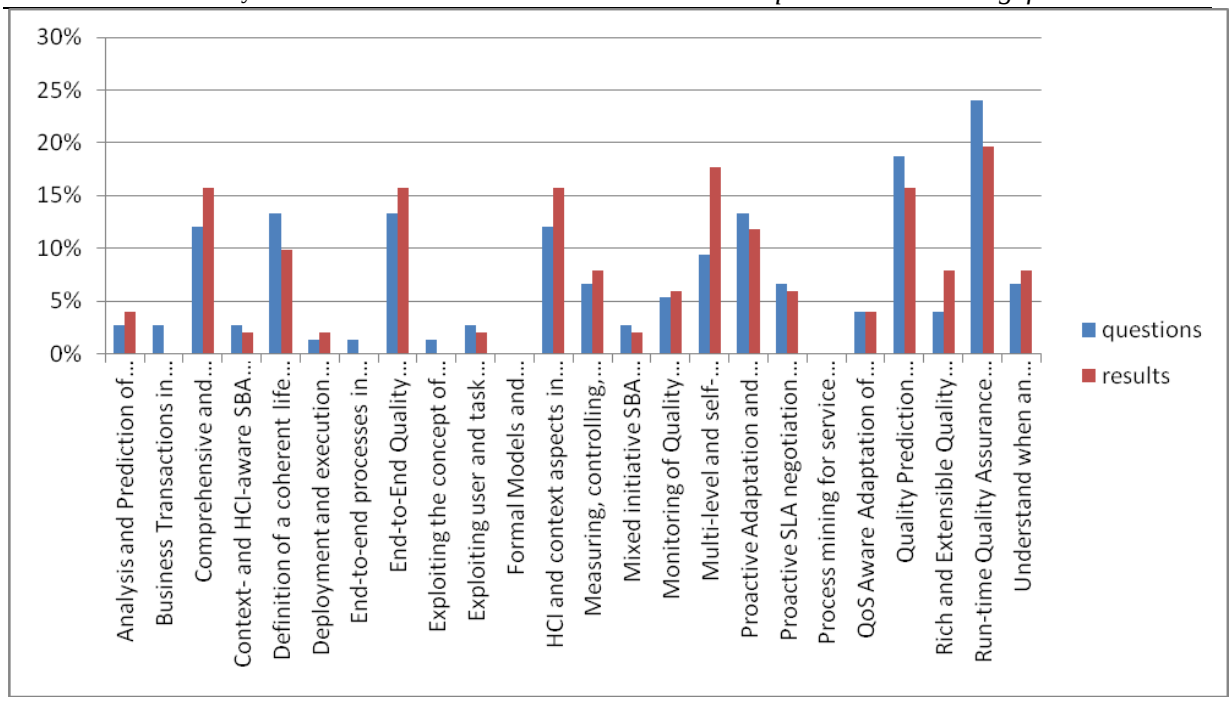

Figure 8: Distribution of Research Questions and Results per Challenge (Metrics 10, 14)

As of Metric 13 we found that 19 research results are related to more than one view.

3.4.4 Goal G4 – Validation Status

The results of the validation metrics are presented in Table 4.

Table 4: Validation Results (Metrics 18,19,20)

No.	Metric	Value
18	Number of case study-based validations	1
18	Number of experiment-based validations	3
18	Number of formal proof-based validations	1
19	Percentage of validated research results	10%
19	Percentage of validated research questions	8%
20	Number of validations performed per question (min, max, average)	0 / 1 / 0,07
20	Number of validations performed per challenge (min, max, average)	0 / 2 /0,45
20	Number of validations performed per element (min, max, average)	0 / 1 / 0,25
20	Number of validations performed per view (min, max, average)	1 / 3 / 2,5

3.4.5 Goal G5 – Integration

As for integration, Figure 9 shows the distribution of the number of relations between the elements per element.

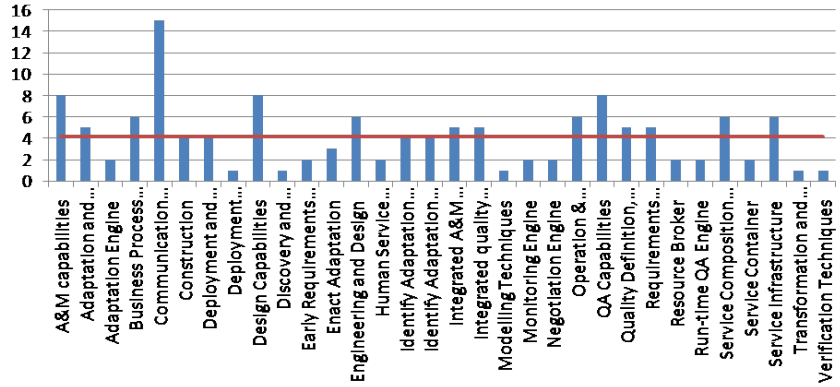

Figure 9: Number of Relations between Elements per Element (metric 21)

Figure 10 reports the distribution of the number of relations between the challenges per challenge.

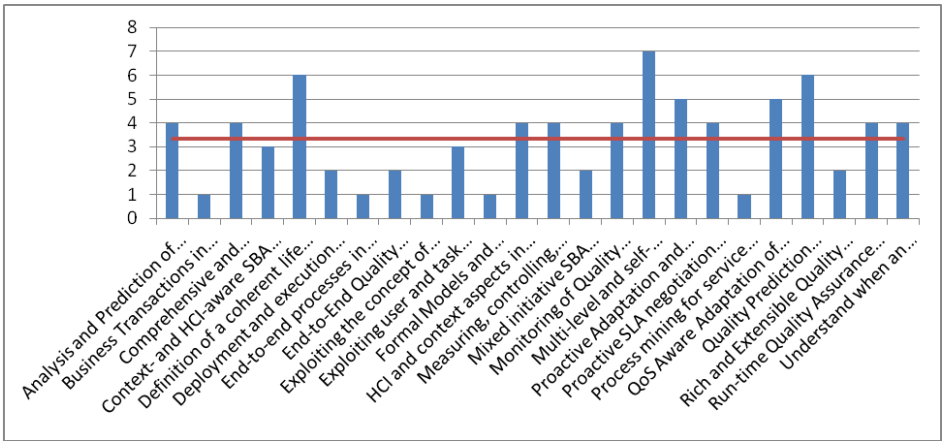

Figure 10: Relations between Challenges per Challenge (metric 23)

The other integration metrics are reported in Table 5.

Table 5: Integration metrics 21, 22, 23, 24, 25, 26 and 27-32

No.	Metric	Value
21	Number of research challenges, which are related to more than one view	22
22	Number of research questions, related to more than one view	30
23	Average number of relations between elements per element	4,18
24	Average number of relations between research challenges per challenge	3,33
25	Average number of relations between research questions per question	1,573333
26	Average number of relations between research results per result	0,733333
27	Number of cross-JRA challenges	12
28	Number of cross-WP challenges	22
29	Number of cross-JRA research questions	11
30	Number of cross-WP research questions	20
31	Number of cross-JRA research results	3
32	Number of cross-WP research results	11

3.4.6 Goal G6 – Consistency with the Knowledge Model

Finally, with respect to the consistency with the Knowledge Model (G6), the number of knowledge model terms referenced by the current IRF is 85, while the number of keywords appearing in IRF and not appearing in the KM is 44 (out of 52) (metrics 33, 34).

3.5 Data Analysis

In this Section we analyze the data collected in Section 3.4. This analysis also contains an interpretation and a recommendation for future work.

3.5.1 Goal G1 – IRF Complexity

At the current stage, the numbers obtained so far do not provide a deep insight into the complexity of the IRF. Indeed, these figures will only be significant when the next versions of the IRF will be evaluated, as they will show the growth and evolution of different parts of the framework. However, they already show that there is a need for more validations as the current number of validations is too small with respect to the research results obtained. That is, the partners should exploit more the pilot S-Cube case studies in order to evaluate their solutions on the common basis.

Another important trend resulting from the analysis of the complexity metrics is that the number of the research questions is higher than the number of the research results. This means that certain important research aspects are not addressed by the current research activities. In addition, the research questions with no associated results will be identified so that and reported back to the joint research activities for either working on these questions or evaluating their relevance. Again, this information will be critical for the next evaluations to understand how the research agendas of the partners are tailored towards those questions.

Finally, as Table 2 shows, the number of relations (1154) is greater than the number of elements (675). This characterizes the integration positively; this trend is expected to continue in the future evaluations. This means that the number of relations should grow more quickly than the number of elements in the IRF.

3.5.2 Goal G2 – Consistency

The results of the consistency metrics show that the relations between the pairs of elements, challenges, research questions and research results show inconsistencies. These inconsistencies have to be analyzed. Although related challenges may not in all cases lead to related research questions and related research questions may not always lead to related research results, the high numbers lead to the recommendation to carefully check these relations.

3.5.3 Goal G3 – Gap Analysis

Given the goal of the gap analysis and the associated metrics, one would expect that all challenges, research questions and results are related to the “Conceptual Research Framework” and to the “Reference Life Cycle” views and that challenges, questions and results are either related to the “Logical Design Environment” or to the “Logical Runtime Environment”. In addition, one would expect that the elements are equally distributed over the views, which means that the views have the same level of detail.

As Figure 6 indicates the distribution of challenges follow our expectations. However, questions and results are not distributed as expected. This indicates that there are missing links between research results and the framework views in the IRF database. These links have to be corrected. In addition, one can clearly see that the “Logical Design Environment” view is not frequently used. This might indicate that this view is not useful for the project. Further investigations are needed to clarify this issue.

For Figure 7 and Figure 8 we would have expected an equal distribution of challenges, research questions and results. As we can clearly see, there the following challenges have neither associated research questions nor associated research results:

- Formal Models and Languages for QoS-Aware Service Compositions (introduced by JRA-2.2)
- Process mining for service discovery (introduced by JRA-2.3)

The project – in particular the workpackages WP-JRA-2.2 and WP-JRA-2.3 – should check those challenges and either detail them to research questions, which can be answered during the lifetime of the project or delete the challenges from the IRF.

In addition, we have the following unanswered research questions in the IRF:

- Business Transactions in Service Networks (introduced by JRA-2.1)
- End-to-end processes in Service Networks (introduced by JRA-2.1)
- Exploiting the concept of service-based applications in the internet of things setting (introduced by JRA-1.1)

The finding of having unanswered research questions is consistent with the finding that the number of research questions (75) is greater than the number of research results (51). Given the fact that there are only three unanswered questions, the numbers also indicate that research results are related to more than one research questions, which is the basis for integrative research. Lastly, the finding clearly shows that there are still research questions to answer and the project should strive to produce research results associated to these questions.

Finally, the distribution of research questions and research results over the challenges shows a strong interest in quality-related questions. The project could use this strong interest to demonstrate integration, especially across the different technological layers.

3.5.4 Goal G4 – Validation Status

In the current stage of the project the result of validation presented in Table 4 indeed show low results and distribution of validations. Given the fact that the project produced 10 journal papers and around 10 good conference papers, the low number indicate that not all validation results were reported to the IRF. Therefore, the recommendation is 1) to cross-check the publications and the IRF and 2) to focus more on validation in the upcoming years.

3.5.5 Goal G5 – Integration

The values of the integration metrics (Figure 9, Figure 10 and Table 5) show that the elements of the framework are well connected. Such elements like “communication backbone” or such challenge like “multilevel and self-adaptation” show the highest level of connectivity due to their integrating nature. On the contrary some of the elements from the “Logical Design Environment” view are much less connected that also confirms the results of the gap analysis that this view is least used in the project. Also some challenges referring to BPM aspects (e.g. Business Transactions or End-to-end Processes in Service Networks), or some quite specialized challenges (e.g. Exploiting concepts of SBA in Internet of Things or Process Mining for Service Discovery) are rather isolated. Therefore, we need to analyze those isolated challenges and integrate them much better into the IRF.

Another important result of the analysis is that the two JRAs are well connected by challenges and research questions. This indicates that the interaction between the two JRAs was established. However, given the fact that only three results belong to both JRAs, we clearly see that the results of this cooperation are still pending.

3.5.6 Goal G6 – Consistency with the Knowledge Model

The results of the consistency with the knowledge model show two things: Firstly, roughly one third of the knowledge model terms are used in the IRF to tag elements, which is a good number. Secondly, 44 terms are used as keywords, which are currently not in the knowledge model. These 44 terms should be used as input to the knowledge model. The WP-IA-1.1 team should decide whether these terms become part of the knowledge model.

4 Recommendations and Conclusions

In this deliverable we introduced a rigorous method to validate the integrated research framework (IRF). This method foresees an internal verification for year 2, an internal verification combined with an external validation for year 3 and, finally, an external validation of the framework at the end of the project. The internal verification is based on the Goal Question Metrics approach, where the objectives of the description of work are refined into goals, operationalized as validation questions and measured by different metrics. The input of this verification was the IRF database and the metrics were defined as SQL queries to this database.

Based on the evaluation of the six goals, we can recommend the following actions to the project:

- Goal G1 – IRF Complexity: The IRF is already quite complex. In the future the project should strive not to add more elements to the IRF but rather to add relations between those elements since these relations indicate that the project produced integrated results.
- Goal G2 – Consistency Check: The results of the consistency check indicate that the relations between elements, challenges, research questions and research results should be carefully reviewed.
- Goal G3 – Gap Analysis: From the gap analysis we can deduce that the challenges are well related to the four different views while these relations are missing for research questions and research results. Especially the “Logical Design Environment” view is not widely used in the project, so it should be considered to remove this view from the IRF. In addition, we found challenges with no associated research questions and research questions without results. The relevant workpackages should review these challenges and questions to decide whether they are still relevant.
- Goal G4 – Validation Status: The validation status clearly shows that there is too little validation in the project. Given the fact that only 10% of the results are currently validated we need to determine the root causes for this problem. Especially, a consistency check between the publication database and the IRF should be performed.
- Goal G5 – Integration: The integration metrics shows that the elements in the IRF are generally well connected. Isolated elements should be investigated in the futures. However, the cross-JRA metrics indicate that effort was made to plan the cooperation between the two JRAs (reflected in cross-JRA challenges and research questions) but the results of this effort are still pending.
- Goal 6 – Consistency with the Knowledge Model: The metrics clearly show that the terms of the knowledge model are used in the IRF. The 44 keywords, which could not be related to the knowledge model, should be used as input to the knowledge model, e. g. it should be clarified whether these terms should become part of the knowledge model.

In the future we need to reduce the identified inconsistencies and problems of the IRF and the progress of these activities should be carefully monitored in the second validation of the framework.

References

[1] Mazza, V.; Pernici, B.: Report on Common Pilot Cases. S-Cube Deliverable CD-IA-2.2.4, 2009

[2] Gehlert, A.; Pistore, M.; Plebani, P.; Versienti, L. (Eds): First Version of the Integration Framework. S-Cube Deliverable CD-IA-3.1.3, 2009

[3] Basili, V.; Rombach, H.: The TAME Project: Towards Improvement-oriented Software Environments. In: IEEE Transactions on Software Engineering, 14 (1988) 6, S.758–773.

[4] Heinrich, L.: Bedeutung von Evaluation und Evaluationsforschung in der Wirtschaftsinformatik. In: Heinrich, L.; Häntschel, I. (Eds.): Evaluation und Evaluationsforschung in der Wirtschaftsinformatik: Handbuch für Praxis, Lehre und Forschung [Evaluation and Evaluation Research in Information Systems: Textbook for Industry, Teaching and Research]. Oldenbourg, 2000, p. 7–22.

A1 Validation Metrics in SQL Queries

1. Number of elements per type in the IRF

```
select 'Number of knowledge model items' as `Item`, count(*) as `Count`
from km
union select 'Number of views', count(*) from view
union select 'Number of elements', count(*) from element
union select 'Number of case studies', count(*) from casestudy
union select 'Number of research challenges', count(*) from challenge
union select 'Number of glossary items', count(*) from glossary
union select 'Number of keywords', count(*) from keyword
union select 'Number of research questions', count(*) from question
union select 'Number of research results', count(*) from result
union select 'Number of scenarios', count(*) from scenario
union select 'Number of validations', count(*) from validation
```

3. Number of all relations in the IRF

```
select 'Element <-> Keyword' as `Relation type`, count(*) as `Count` from
elementtokeyword
union select 'Element <-> Glossary', count(*) from elementtoglossary
union select 'View <- Element', count(*) from element
union select 'View <-> Glossary', count(*) from viewtoglossary
union select 'View <-> Keyword', count(*) from viewtokeyword
union select 'Challenge <-> Challenge', count(*) from challengetochallenge
union select 'Challenge <-> Glossary', count(*) from challengetoglossary
union select 'Challenge <-> Element', count(*) from challengetoelement
union select 'Challenge <-> Keyword', count(*) from challengetokeyword
union select 'Question <-> Question', count(*) from questiontoquestion
union select 'Question <-> Element', count(*) from questiontoelement
union select 'Question <-> Glossary', count(*) from questiontoglossary
union select 'Question <-> Keyword', count(*) from questiontokeyword
union select 'Question <-> Challenge', count(*) from questiontochallenge
union select 'Result <-> Result', count(*) from resulttoresult
union select 'Result <-> Element', count(*) from resulttoelement
union select 'Result <-> Glossary', count(*) from resulttoglossary
union select 'Result <-> Keyword', count(*) from resulttokeyword
union select 'Result <-> Question', count(*) from resulttoquestion
union select 'Case Study <- Scenario', count(*) from scenario
union select 'Case Study <-> Glossary', count(*) from casestudytoglossary
union select 'Case Study <-> Keyword', count(*) from casestudytokeyword
union select 'Case Study <-> Reference', count(*) from
casestudytoreference
union select 'Scenario <-> Glossary', count(*) from scenariotoglossary
union select 'Scenario <-> Keyword', count(*) from scenariotokeyword
union select 'Scenario <-> Element', count(*) from scenariotoelement
union select 'Scenario <-> Challenge', count(*) from scenariotochallenge
union select 'Scenario <-> Reference', count(*) from scenariotoreference
union select 'Validation <-> Glossary', count(*) from validationtoglossary
union select 'Validation <-> Keyword', count(*) from validationtokeyword
union select 'Validation <-> Scenario', count(*) from validationtoscenario
union select 'Validation <-> Reference', count(*) from
validationtoreference
union select 'Validation <-> Question', count(*) from validationtoquestion
union select 'Validation <-> Keyword', count(*) from validationtoresult
```

4. Check whether for each pair of related research results belonging to different research questions that there is also a relation between the research questions

```
SELECT distinct RR.* from (scube2.resulttoresult AS RR,
scube2.resulttoquestion AS RQ, scube2.resulttoquestion AS RRQ)  where
```

```
(RQ.question <> RRQ.question AND RR.result = RQ.result AND RR.relresult =
RRQ.result AND NOT EXISTS (SELECT * FROM scube2.questiontoquestion AS QQ
WHERE ((RQ.question=QQ.question AND RRQ.question=QQ.relquestion) OR
(RQ.question=QQ.relquestion AND RRQ.question=QQ.question))))
```

5. Check whether for each pair of related research questions there at least one relation between research result related to those research questions

```
SELECT distinct QQ.* from (scube2.questiontoquestion AS QQ,
scube2.resulttoquestion AS QR, scube2.resulttoquestion AS RQR) WHERE
(QQ.question=QR.question and QQ.relquestion=RQR.question) AND NOT EXISTS
(SELECT * FROM scube2.resulttoresult AS RR WHERE ((RR.result=QR.result AND
RR.relresult=RQR.result) OR (RR.result=RQR.result AND
RR.relresult=QR.result)))
```

6. Do 14) and 15) for the pairs: research question-research challenge and research challenge-element

```
SELECT distinct QQ.* from (scube2.questiontoquestion AS QQ,
scube2.questiontochallenge AS QC, scube2.questiontochallenge AS RQC)
where (QC.challenge <> RQC.challenge AND QQ.question = QC.question AND
QQ.relquestion = RQC.question AND NOT EXISTS (SELECT * FROM
scube2.challengetochallenge AS CC WHERE ((QC.challenge=CC.challenge AND
RQC.challenge=CC.relchallenge) OR (QC.challenge=CC.relchallenge AND
RQC.challenge=CC.challenge))))

SELECT distinct CC.* from (scube2.challengetochallenge AS CC,
scube2.questiontochallenge AS QC, scube2.questiontochallenge AS RQC) WHERE
(CC.challenge=QC.challenge and CC.relchallenge=RQC.challenge) AND NOT
EXISTS (SELECT * FROM scube2.questiontoquestion AS QQ WHERE
((QQ.question=QC.question AND QQ.relquestion=RQC.question) OR
(QQ.question=RQC.question AND QQ.relquestion=QC.question)))

SELECT distinct CC.* from (scube2.challengetochallenge AS CC,
scube2.challengetoelement AS CE, scube2.challengetoelement AS RCE)  where
(CE.element <> RCE.element  AND CC.challenge = CE.challenge AND
CC.relchallenge = RCE.challenge AND NOT EXISTS (SELECT * FROM
scube2.elementtoelement AS EE WHERE ((CE.element=EE.element AND
RCE.element=EE.relelement) OR (CE.element=EE.relelement AND
RCE.element=EE.element))))

SELECT distinct EE.* from (scube2.elementtoelement AS EE,
scube2.challengetoelement AS CE, scube2.challengetoelement AS RCE) WHERE
(EE.element=CE.element and EE.relelement=RCE.element) AND NOT EXISTS
(SELECT * FROM scube2.challengetochallenge AS CC WHERE
((CC.challenge=CE.challenge AND CC.relchallenge=RCE.challenge) OR
(CC.challenge=RCE.challenge AND CC.relchallenge=CE.challenge)))
```

7. Number of elements per view

```
select `view` as `View`, count(name) as `No. of elements`
from `element`
group by `view`
```

8. Number of research challenges per elements

```
select E.name as `Element`, count(distinct challenge) as `No. of
challenges` from element E left outer join challengetoelement CE on
E.name=CE.element
group by E.name
```

9. Number of research challenges per view

```
select E.`view` as `View`, count(distinct CE.challenge) as `No. of
challenges`
```

```
from element E left outer join challengetoelement CE on E.name=CE.element
group by E.`view`
order by 2 desc
```

10. Number of research questions per research challenge (min/max/avg)

```
select C.name, count(distinct QC.question) as qpc
      from challenge C left outer join questiontochallenge QC
                                  on C.name=QC.challenge
      group by C.name
```

11. Number of research questions per element and per view

```
select E.name as `Element name`, count(distinct QE.question) as `No. of
questions` from element E left outer join questiontoelement QE on
E.name=QE.element
group by E.name
order by 2 desc

select E.`view` as `View`, count(distinct QE.question) as `No. of
questions` from element E left outer join questiontoelement QE on
E.name=QE.element
group by E.`view`
order by 2 desc
```

12. Number of research results per research question

```
select Q.name as `Question`, count(distinct RQ.result) as `No. of results`
from question Q left outer join resulttoquestion RQ on Q.name=RQ.question
group by Q.name
order by 2 desc
```

13. Number of research results, which belong to more than one view

```
select * from (select RQ.result as `Result`, count(distinct E.`view`) as
`No. of views`
from resulttoquestion RQ left outer join question Q on RQ.question=Q.name
left outer join questiontoelement QE on Q.name=QE.question
left outer join element E on QE.element=E.name
group by `Result`) t where t.`No. of views`>1
```

14. Number of research results per challenge, element and view

```
select C.name as `Challenge`, count(distinct RQ.result) as `No. of
results`
from challenge C left outer join questiontochallenge QC on
C.name=QC.challenge
     left outer join resulttoquestion RQ on QC.question=RQ.question
group by C.name

select E.name as `Element`, count(distinct RQ.result) as `No. of results`
from element E left outer join questiontoelement QE on E.name=QE.element
     left outer join resulttoquestion RQ on QE.question=RQ.question
group by E.name

select V.name as `View`, count(distinct RQ.result) as `No. of results`
from  `view` V left outer join element E on V.name=E.`view`
      left outer join questiontoelement QE on E.name=QE.element
      left outer join resulttoquestion RQ on QE.question=RQ.question
group by V.name
```

15. Number of scenarios per case study

```
SELECT CS.name,COUNT(S.name) FROM scube2.casestudy AS CS LEFT JOIN
scube2.scenario AS S ON CS.name = S.casestudy GROUP BY CS.name
```

16. Number of validation result per scenario

```
SELECT S.name,COUNT(VS.validation) FROM scube2.scenario AS S LEFT JOIN
scube2.validationtoscenario AS VS ON S.name = VS.scenario GROUP BY S.name
```

17. Number of validation results per case study

```
SELECT CS.NAME, COUNT(VR.result) FROM (((scube2.casestudy AS CS LEFT JOIN
scube2.scenario AS S ON CS.name = S.casestudy) LEFT JOIN
scube2.validationtoscenario AS VS ON S.name=VS.scenario) LEFT JOIN
scube2.validationtoresult AS VR ON VS.validation=VR.validation) GROUP BY
CS.name
```

18. Number of validations per research method

```
SELECT method,COUNT(*) FROM scube2.validation AS V GROUP BY method
```

19. Percentage of validated research results and validated research questions

```
SELECT count(result)/(select count(name) from scube2.result)*100 FROM
scube2.validationtoresult AS VR

SELECT count(question)/(select count(name) from scube2.question)*100 FROM
scube2.validationtoquestion AS VQ
```

20. Number of validations performed per research question, research challenge, element and view

```
SELECT Q.name, COUNT(distinct VQ.validation) FROM scube2.question AS Q
LEFT JOIN scube2.validationtoquestion AS VQ ON Q.name = VQ.question GROUP
BY Q.name

SELECT QC.challenge, COUNT(distinct VQ.validation) FROM
((scube2.questiontochallenge AS QC JOIN scube2.challenge AS C ON
C.name=QC.challenge) LEFT JOIN scube2.validationtoquestion AS VQ ON
QC.question=VQ.question) GROUP BY QC.challenge

SELECT E.name,COUNT(distinct VQ.validation) FROM ((scube2.element AS E
LEFT JOIN scube2.questiontoelement AS QE ON E.name=QE.element) LEFT JOIN
scube2.validationtoquestion AS VQ ON QE.question=VQ.question) GROUP BY
E.name

SELECT E.view,COUNT(distinct VQ.validation) FROM (((scube2.element AS E
LEFT JOIN scube2.challengetoelement AS CE ON E.name=CE.element) LEFT JOIN
scube2.questiontochallenge AS QC ON CE.challenge = QC.challenge) LEFT JOIN
scube2.validationtoquestion AS VQ ON QC.question=VQ.question) GROUP BY
E.view
```

21. Number of relations between elements per element

```
select E.name as `Element name`, count(*) as `No. of relations`
from `element` E left outer join `elementtoelement` EE on
      (E.name=EE.element or E.name=EE.relelement)
group by E.name
order by 2 desc
```

22. Number of research challenges that are related to more than one view

```
select CE.challenge as `Challenge name`, count(distinct E.view) as `No. of
views`
from challengetoelement CE left outer join element E on
(CE.element=E.name)
group by CE.challenge
having count(distinct E.view)>1
order by 2 desc
```

23. Number of relations between research challenges per challenge

```
select C.name as `Challenge name`, count(*) as `No. of relations`
from `challenge` C left outer join `challengetochallenge` CC on
      (C.name=CC.challenge or C.name=CC.relchallenge)
group by C.name
order by 2 desc
```

24. Number of research questions, related to more than one view

```
select Q.name as `Question`, count(distinct E.`view`) as `No. of views`
from question Q left outer join questiontoelement QE on Q.name=QE.question
     left outer join element E on QE.element=E.name
group by Q.name
having count(distinct E.`view`)>1
order by 2 desc
```

25. Number of relations between research questions per question

```
select Q.name as `Question`, count(*) as `No. of relations`
from `question` Q left outer join `questiontoquestion` QQ on
      (Q.name=QQ.question or Q.name=QQ.relquestion)
group by Q.name
```

26. Number of relations between research results per result

```
select Q.name as `Question`, count(distinct RQ.result) as `No. of results`
from question Q left outer join resulttoquestion RQ on Q.name=RQ.question
group by Q.name
order by 2 desc
```

27. Number of cross-JRA challenges: Number of challenges belonging to more than one element in the view "Conceptual Research Framework" and one element is in the set {Adaptation & Monitoring; Engineering & Design; Quality Definition, Negotiation and Assurance} and another element is in the set {Business Process Management; Service Composition and Coordination; Service Infrastructure}

```
SELECT COUNT(DISTINCT CE.challenge) FROM scube2.challengetoelement AS CE
JOIN scube2.elementtoJRA AS EJ ON CE.element = EJ.element WHERE (EJ.jra=1
AND CE.challenge IN (SELECT CE.challenge FROM scube2.challengetoelement AS
CE JOIN scube2.elementtoJRA AS EJ ON CE.element = EJ.element WHERE
EJ.jra=2)) OR (EJ.jra=2 AND CE.challenge IN (SELECT CE.challenge FROM
scube2.challengetoelement AS CE JOIN scube2.elementtoJRA AS EJ ON
CE.element = EJ.element WHERE EJ.jra=1))
```

28. Number of cross-WP research challenges: Number of research challenges belonging to more than one element of the view "Conceptual Research Framework"

```
SELECT * FROM scube2.challengetoelement AS CE JOIN scube2.element AS E ON
CE.element = E.name WHERE LCASE(E.view) = LCASE('Conceptual Research
Framework') GROUP BY CE.challenge HAVING (COUNT(*) > 1)
```

29. 27Number of cross-JRA questions: Number of research questions belonging to more than one element in the view "Conceptual Research Framework" and one element is in the set {Adaptation & Monitoring; Engineering & Design; Quality Definition, Negotiation and Assurance} and another element is in the set {Business Process Management; Service Composition and Coordination; Service Infrastructure}

```
SELECT COUNT(DISTINCT CE.question) FROM scube2.questiontoelement AS CE
JOIN scube2.elementtoJRA AS EJ ON CE.element = EJ.element WHERE (EJ.jra=1
AND CE.question IN (SELECT CE.question FROM scube2.questiontoelement AS CE
JOIN scube2.elementtoJRA AS EJ ON CE.element = EJ.element WHERE EJ.jra=2))
OR (EJ.jra=2 AND CE.question IN (SELECT CE.question FROM
scube2.questiontoelement AS CE JOIN scube2.elementtoJRA AS EJ ON
CE.element = EJ.element WHERE EJ.jra=1))
```

30. Number of cross-WP research questions: Number of research questions belonging to more than one element of the view "Conceptual Research Framework"

```
select Q.name as `Question`, count(distinct EJ.jra) as `No. of WPs`
from question Q join questiontoelement QE on Q.name=QE.question
     join element E on QE.element=E.name
     join elementtojra EJ on E.name=EJ.element
where E.`view`='Conceptual Research Framework'
group by Q.name
having count(distinct EJ.jra)>1
```

31. Number of cross-JRA research results: Number of research results belonging to more than one element in the view "Conceptual Research Framework" and one element is in the set {Adaptation & Monitoring; Engineering & Design; Quality Definition, Negotiation and Assurance} and another element is in the set {Business Process Management; Service Composition and Coordination; Service Infrastructure}

```
SELECT COUNT(DISTINCT CE.result) FROM scube2.resulttoelement AS CE
  JOIN scube2.elementtoJRA AS EJ ON CE.element = EJ.element
  WHERE
    (EJ.jra=1 AND CE.result IN
      (SELECT CE.result FROM scube2.resulttoelement AS CE
         JOIN scube2.elementtoJRA AS EJ ON CE.element = EJ.element WHERE
EJ.jra=2))
    OR
    (EJ.jra=2 AND CE.result IN
      (SELECT CE.result FROM scube2.resulttoelement AS CE
         JOIN scube2.elementtoJRA AS EJ ON CE.element = EJ.element WHERE
EJ.jra=1))
```

32. Number of cross-WP research results: Number of research results belonging to more than one element of the view "Conceptual Research Framework"

```
select R.name as `Result`, count(distinct EJ.jra) as `No. of WPs`
from result R join resulttoelement RE on R.name=RE.result
     join element E on RE.element=E.name
     join elementtojra EJ on E.name=EJ.element
where E.`view`='Conceptual Research Framework'
group by R.name
having count(distinct EJ.jra)>1
```

33. Number of KM terms referenced by the IRF

```
SELECT COUNT(X.keyword) FROM ((
(SELECT keyword FROM scube2.challengetokeyword AS CK) UNION
(SELECT keyword FROM scube2.casestudytokeyword  AS CSK) UNION
(SELECT keyword FROM scube2.elementtokeyword  AS EK) UNION
(SELECT keyword FROM scube2.questiontokeyword  AS QK) UNION
(SELECT keyword FROM scube2.resulttokeyword  AS RK) UNION
(SELECT keyword FROM scube2.scenariotokeyword  AS SK) UNION
(SELECT keyword FROM scube2.validationtokeyword  AS VK) UNION
(SELECT keyword FROM scube2.viewtokeyword  AS VWK) UNION
(SELECT glossary FROM scube2.challengetoglossary AS CG) UNION
(SELECT glossary FROM scube2.casestudytoglossary AS CSG) UNION
(SELECT glossary FROM scube2.elementtoglossary AS EG) UNION
(SELECT glossary FROM scube2.questiontoglossary AS QG) UNION
(SELECT glossary FROM scube2.resulttoglossary  AS RG) UNION
(SELECT glossary FROM scube2.scenariotoglossary  AS SG) UNION
(SELECT glossary FROM scube2.validationtoglossary  AS VG) UNION
(SELECT glossary FROM scube2.viewtoglossary  AS VWG)
)AS X) JOIN scube2.km ON LCASE(X.keyword) = LCASE(km.name)
```

34. Number of keywords not included in the KM

```
SELECT COUNT(X.keyword) FROM ((
(SELECT keyword FROM scube2.challengetokeyword AS CK) UNION
(SELECT keyword FROM scube2.casestudytokeyword  AS CSK) UNION
(SELECT keyword FROM scube2.elementtokeyword  AS EK) UNION
(SELECT keyword FROM scube2.questiontokeyword  AS QK) UNION
(SELECT keyword FROM scube2.resulttokeyword  AS RK) UNION
(SELECT keyword FROM scube2.scenariotokeyword  AS SK) UNION
(SELECT keyword FROM scube2.validationtokeyword  AS VK) UNION
(SELECT keyword FROM scube2.viewtokeyword  AS VWK) )AS X) WHERE
LCASE(X.keyword) NOT IN (SELECT LCASE(name) FROM scube2.km AS KM)
```