Requirements Engineering

SCRAM-CK: A collaborative requirements engineering process for designing a web based e-science toolkit

--Manuscript Draft--

Manuscript Number:	REJ507R2
Full Title:	SCRAM-CK: A collaborative requirements engineering process for designing a web based e-science toolkit
Article Type:	Original Research
Keywords:	requirements elicitation; requirements capture; user centered requirements engineering; Requirements evolution; collaborative design; design evolution; CK Theory; SCRAM
Corresponding Author:	Abraham Nieva de la Hidalga, PhD Cardiff University Cardiff, South Glamorgan UNITED KINGDOM
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Cardiff University
Corresponding Author's Secondary Institution:	
First Author:	Abraham Nieva de la Hidalga, PhD
First Author Secondary Information:	
Order of Authors:	Abraham Nieva de la Hidalga, PhD
	Alex R Hardisty
	Andrew C Jones, Dr
Order of Authors Secondary Information:	

SCRAM-CK: A collaborative requirements engineering process for designing a web based e-science toolkit

Abraham Nieva de la Hidalga
Research Associate; School of Computer Science & Informatics, Cardiff University, Queen's Buildings, 5 The Parade, Roath, Cardiff, CF24 3AA, Wales, UK; Tel. +44(0)2920 870 321
a.nieva@cs.cardiff.ac.uk
Alex Hardisty
Director of Informatics Projects; School of Computer Science and Informatics, Cardiff University, Queen's Buildings, 5 The Parade, Roath, Cardiff, CF24 3AA, Wales, UK; Tel. +44(0)2920 874 761
HardistyAR@cardiff.ac.uk
Andrew Jones
Senior Lecturer; School of Computer Science and Informatics, Cardiff University, Queen's Buildings, 5 The Parade, Roath, Cardiff, CF24 3AA, Wales, UK; Tel. +44(0)2920 875 537
Andrew.C.Jones@cs.cardiff.ac.uk

Abstract

This paper presents SCRAM-CK, a method to elicit requirements by means of strong user involvement supported by prototyping activities. The method integrates two existing approaches, SCRAM and CK Theory. SCRAM provides the framework for requirements management while CK Theory provides a framework for reasoning about design and its evolution. The method is demonstrated with the definition and refining of requirements for the BioVeL web toolkit. The objective of BioVeL is to allow scientists to understand, run, modify, and construct workflows for data analysis with minimal training using a web based interface. The proposed method is supported by prototyping activities for gathering user feedback, and refining requirements and design proposals. Using this method, the prototypes evolved from simple workflow execution enablers to include more complex functionalities for reviewing, modifying, and building workflows in later versions. This paper presents a contribution to the application of techniques for requirements engineering. SCRAM-CK is an amalgamated method that combines a user-centred continuous refinement approach with support for design evolution through prototyping. The paper also shows the influence of the requirements engineering process in the evolution of design proposals.

Keywords: requirements elicitation; requirements capture; user centered requirements engineering; requirements evolution; collaborative design; design evolution; CK Theory; SCRAM

SCRAM-CK: applying a collaborative requirements engineering process for designing a web based e-science toolkit

This paper presents SCRAM-CK, a method to elicit requirements by means of strong user involvement supported by prototyping activities. The method integrates two existing approaches, SCRAM and CK Theory. SCRAM provides the framework for requirements management while CK Theory provides a framework for reasoning about design and its evolution. The method is demonstrated with the definition and refining of requirements for the BioVeL web toolkit. The objective of BioVeL is to allow scientists to understand, run, modify, and construct workflows for data analysis with minimal training using a web based interface. The proposed method is supported by prototyping activities for gathering user feedback, and refining requirements and design proposals. Using this method, the prototypes evolved from simple workflow execution enablers to include more complex functionalities for reviewing, modifying, and building workflows in later versions. This paper presents a contribution to the application of techniques for requirements engineering. SCRAM-CK is an amalgamated method that combines a user-centred continuous refinement approach with support for design evolution through prototyping. The paper also shows the influence of the requirements engineering process in the evolution of design proposals.

Keywords: requirements elicitation; requirements capture; user centered requirements engineering; requirements evolution; collaborative design; design evolution; CK Theory; SCRAM; prototyping

1 Introduction

The Biodiversity Virtual e-Laboratory (BioVeL) is intended to meet the needs of the Biodiversity research community with tools for data analysis that will help understanding biodiversity in a rapidly changing environment. To achieve this, BioVeL customises, deploys and supports the myGrid Software Family (Taverna, myExperiment, BioCatalogue [13, 26, 35, 41]). BioVeL is particularly concerned with developing a sustainable infrastructure for supporting biodiversity e-Science, especially by fostering the development and reuse of scientific workflows. BioVeL aims to make these developments available to as wide a range of biodiversity scientists as possible, meeting their individual needs in an easy-to-use manner. However, the definition of workflows and their execution in environments like Taverna often demands a degree of computer literacy that does not match with the skills or interests of scientists [11, 12]. Moreover, the results of a previous project in the health domain indicated that user-centered design and genuine interdisciplinary approaches are essential to create solutions that are fit for purpose, sustainable and address the real needs of all stakeholders [29].

Given these concerns, a Requirements Engineering (RE) process to allow the participation of biodiversity scientists as co-designers was seen as the best strategy. The main characteristics of such an RE process are that it is user-centered, continuous, and supportive of design evolution and prototyping which is compatible with the agile development approach adopted for the BioVeL project. The RE process must be user-centered, including the biodiversity scientists as part of the design team to produce a system that matches their requirements closely. The RE process must be continuous because the matching between the requirements and the design must be reviewed constantly as the project progresses and new requirements are discovered or priorities change. The RE process needs to support design evolution because requirements and designs are expected to change as a result of the project findings and in response to prototype evaluation. Finally, the RE process needs to support prototyping activities because prototypes will be used as the means for practical evaluation of design decisions and also to stimulate exploration of new possibilities.

The SCRAM-CK RE process was developed to address these needs. SCRAM was selected because it is a user-centered requirements engineering method [36, 38] and we had some previous knowledge of its application to other projects. However, SCRAM is not specifically designed to monitor the evolution of designs or to foster the direct participation of users as co-designers. SCRAM expects designs to evolve in response to requirements refinements as an activity that happens in parallel [36]. In contrast, CK Theory provides a collaborative design method that supports the evolution of designs and encourages the collaboration of users and designers [15, 16, 34]. CK can be used to map the evolution of designs in response to changes in requirements, complementing SCRAM.

This article presents the SCRAM-CK process and describes how it has been used during the design, development and continuous refinement of the BioVeL workflow web toolkit. The paper is structured as follows. The theory backing SCRAM-CK is presented in Section 2. The application of SCRAM-CK in the development of the BioVeL workflow web toolkit is presented in Section 3, covering each stage of the process. Section 4 evaluates the results obtained by applying the method in practice. Section 5 provides a deeper analysis and discussion of the main issues discovered while implementing and using the method. Finally section 6 presents the conclusions and opportunities for future research.

2 SCRAM-CK

The primary measure of success of a software system is the degree to which it meets the purpose for which it was intended [24]. Requirements engineering is the process by which the requirements to meet that purpose are determined [6]. The main tasks of the requirements engineering process are: elicitation, modelling, analysis, validation/verification, and management.

Requirement elicitation involves understanding the needs of stakeholders and the contexts in which the to-be-developed software will be used. Requirement modelling involves creating representations of the requirements that are used to communicate and negotiate. Analysis involves determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders. Requirements validation and verification involves checking that a system meets requirements and specifications to fulfil its intended purpose. Requirements management is the process that organises the requirements engineering process, managing change, communicating and negotiating decisions with stakeholders [6].

Modern requirements engineering processes need to be interwoven into the software lifecycle from design and planning through to development, deployment and decommissioning [17, 37, 39]. The interweaving addresses changes in technology and changes in the nature of requirements. A robust and realistic software development process allows requirements engineers, designers, system architects, and developers to work concurrently and iteratively to describe the artefacts they wish to produce. Such a process allows developers to better understand problems through consideration of architectural constraints, so they can develop and adapt architectures based on requirements [25]. Development processes that facilitate fast, incremental delivery are essential for software systems that need to be developed quickly, with progressively shorter times-to-market [25].

The problem lies in selecting the adequate requirements engineering methodologies from amongst the many proposed in the literature. A comprehensive review of requirements engineering technologies found over 60 methods, techniques, and approaches supporting the different phases and activities of the requirements engineering process [28]. This abundance complicates selecting appropriate methodologies to apply to a particular problem. An alternative suggested in the literature is to select a method or combination of methods based on the characteristics of the project and the system being developed [2, 36].

The main factor to consider when selecting the requirements engineering method is that it should be appropriate to the type of system under development, and the expected discovery contexts. For interactive software such as the BioVeL web toolkit, the most commonly used techniques are based on scenarios and prototyping. Prototypes and scenarios are techniques that can be used to discover and describe tacit requirements and knowledge (See [2] for further discussion of the factors and techniques mentioned here). The use of scenarios facilitates requirements analysis and validation, and allows the inclusion of the target user group from an early stage in the development process. Scenario-based models are easiest for practitioners and non-technical stakeholders to use [6].

The Scenario Requirements Analysis Method (SCRAM) is one such approach designed to cover the entire requirements engineering process [36, 38]. SCRAM is designed as an agile method for user centered requirements which promotes the participation of users during the entire requirements engineering process. SCRAM, however, needs to be extended to facilitate the monitoring of the evolution of designs and to support an extended process that covers the entire software lifecycle, i.e. it is not confined to a single stage at the beginning of the project. Additionally, the participation of the user as a designer is not considered amongst the user roles in SCRAM. For these reasons a methodology that supported reasoning about design evolution with the inclusion of users as co-designers was required as a complement to SCRAM. CK Theory can be used to bridge this gap. CK Theory provides a framework for reasoning about design, the design process and the evolution of design proposals. Moreover, CK Theory encourages the collaboration of users as designers [15, 16, 34], thus appearing as a good candidate to complement SCRAM.

2.1 SCRAM – Scenario Requirements Analysis Method

SCRAM uses a combination of prototypes, scenario scripts, and design rationale to elicit and validate user requirements [36]. Prototypes and concept demonstrators provide a designed artefact that users can react to. Scenarios are used to situate the designed artefact in a context of use, thereby helping users relate the design to their work/task context. The designer’s reasoning is deliberately exposed to the user to encourage user participation in the decision process. The requirements are summarised on a whiteboard to identify dependencies and priorities.

SCRAM facilitates the management of the requirements engineering process by dividing the activities in four stages (Figure 1): (1) Initial Requirements Capture and Domain Familiarisation, (2) Storyboarding Design, (3) Requirements Exploration, and (4) Prototyping Requirements Validation.

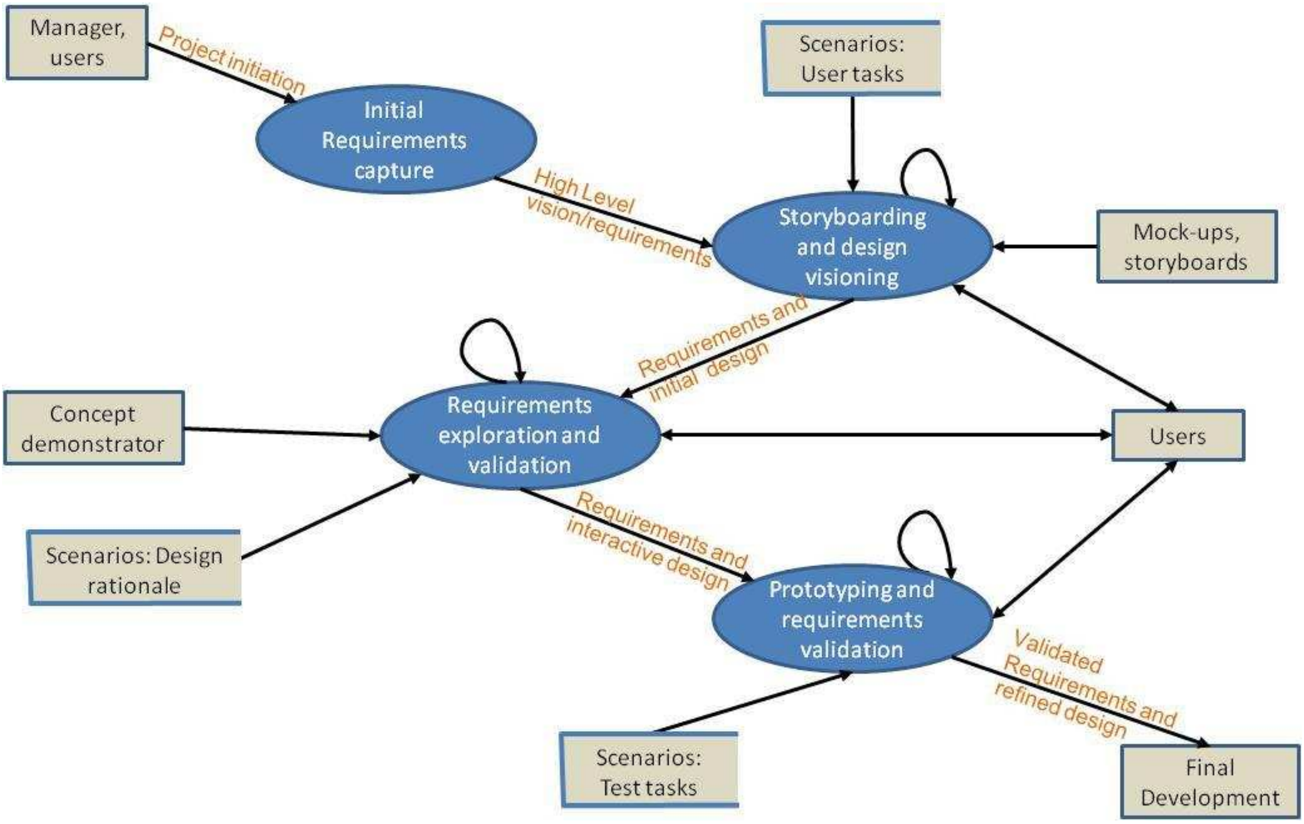

Figure 1 The SCRAM method (based on the diagram that appears in [36, 38])

Initial requirements capture is focused on the domain familiarisation activities. This stage is conducted by conventional interviewing and fact-finding techniques to gain sufficient information to develop a first concept demonstrator. In practice this takes 1-2 sessions with stakeholders.

Storyboarding and design visioning creates early visions of the required system that are explained to users in storyboard walkthroughs to get feedback on feasibility of the different designs being demonstrated.

Requirements exploration uses concept demonstrators and early prototypes to present more detailed designs to users in scenario-driven and semi-interactive demonstrations. This facilitates the analysis of the design and requirements validation activities.

Prototyping and requirements validation develops more fully functional prototypes and continues refining requirements until a prototype is agreed to be acceptable by the users.

2.2 CK Theory – Concept-Knowledge Theory

CK Theory is a framework for reasoning about design [15, 16]. CK Theory provides a definition of design that is independent from the domain in which it is used, integrating creative thinking and innovation [18]. CK Theory models the design process through iterative interactions and expansions of a concept space and a knowledge space.

The knowledge (K) space contains propositions that have a logical status for the designer. Having a logical status means that the designer assigns a degree of confidence to the logical status of a proposition (true, false or un-decidable) [15]. The concept (C) space contains propositions which do not have a logical status in K. This means that when a concept is formulated it is not possible to prove that it belongs to K. That is, the designer lacks evidence to judge if a given proposition can be incorporated into K [15].

CK Theory defines design as the processes by which a concept generates other concepts or is transformed into a knowledge proposition in K. CK Theory also defines the set of transformations that can occur for generating new concepts and deriving knowledge propositions from them. Design is viewed as the process mapping the expansions on the C-K spaces. There are four operations that describe these expansions: conjunction (C -> K), disjunction (K -> C), partition (C -> C), and expansion (K -> K). The following figure shows the dynamic behaviour of the operators.

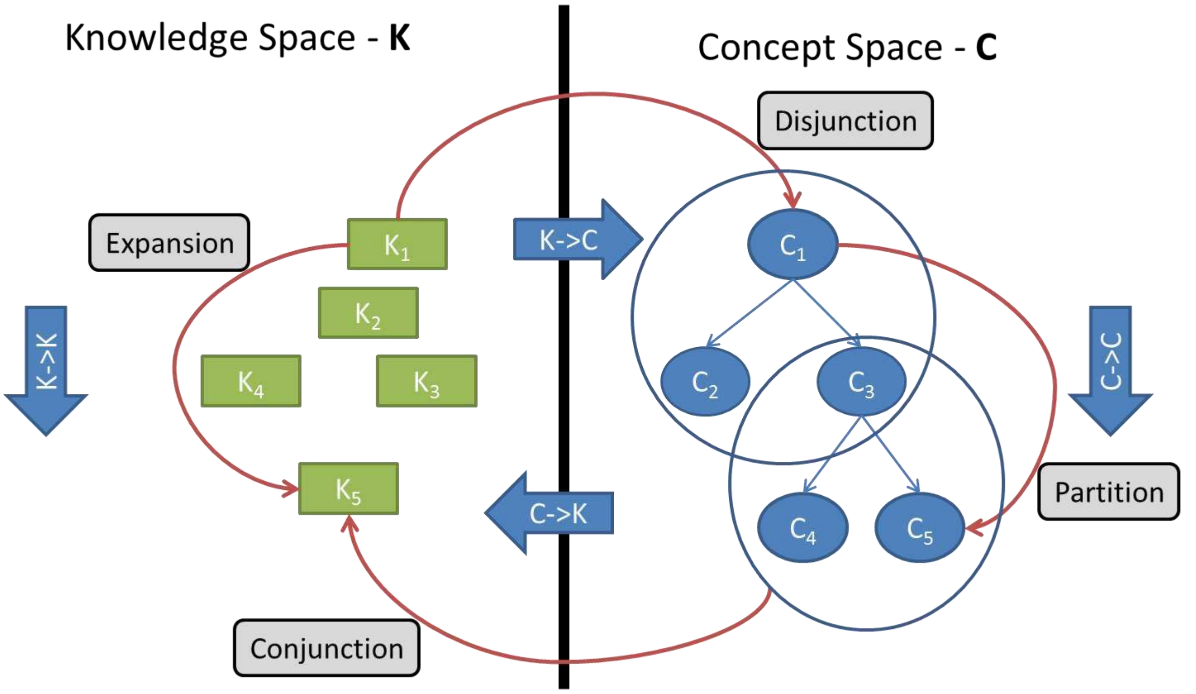

Figure 2 Dynamics of the design process according to CK Theory (derived from the "CK dynamics" diagram in [15, 18])

The design process begins by collecting the knowledge propositions around the design problem at hand, building K. Then disjunction is applied to identify the concepts which are contained in that knowledge, creating C. This is followed by further partitioning of C by subdividing, grouping, and organising the concepts. The next step involves trying to incorporate the new concepts in propositions of K, applying conjunction. This will cause the further expansion of the knowledge space. This expansion of C and K spaces continues throughout the design process.

Conjunction expands the knowledge space when different concepts are combined and rationalised to form new knowledge statements. The inverse operation is disjunction, where new concepts are derived from existing knowledge statements.

2.3 Integration of SCRAM-CK

The SCRAM-CK RE process was proposed to explore user preferences for light-weight web-based interfaces for executing, modifying and running workflows. The main drivers for the integration of SCRAM and CK theory were: (1) to allow the participation of users in the design process and (2) to drive the evolution of design. In this process SCRAM provides the framework for requirements management while CK theory deals with the uncertainties of design (why designs evolve). When a design process starts, the design team knows little in the way of concrete facts (K) but may have lots of ideas, assumptions and guesses about what the design should look like (C). CK is about pushing the design forward and making it evolve though a complete framework for reasoning about design.

Following SCRAM, the SCRAM-CK RE process is organized in four stages. These stages are described as follows.

1. **Initial requirements capture:** the activities during this stage are centered on domain familiarisation. Following the SCRAM method, at this stage conventional interviewing and fact-finding techniques are used to acquire sufficient information to develop an initial requirements list and to focus the design propositions. During this stage the CK Spaces are defined from domain knowledge. Users are not consulted directly for opinions but observed and interviewed about the domain.
2. **Storyboarding and design visioning:** the activities of this stage are geared towards the initial analysis of the requirements and a set of design alternatives, as advised by SCRAM. Different materials including: tutorials, walkthroughs, and mock-ups are used in an interactive session based on design games derived from CK Theory to facilitate explaining and discussing different aspects of the design alternatives. This facilitates the further expansion of the initial design space. During this stage the user is expected to take a more active role in the design process, particularly in the selection of design alternatives and in the ranking of requirements.
3. **Requirements exploration:** this stage requires the use of more advanced demonstrations to further validate and refine the requirements set. As suggested in SCRAM, more functional prototypes can be used as a concept demonstrator for discussing further design details with users in scenario-driven, semi-interactive demonstrations. CK theory will

support making explicit the mappings between the requirements set and the design decisions taken to fulfil those requirements. The users will participate in the validation of requirements and discussing how well they are served by the selected designs.

4. **Prototyping and requirements validation:** this is the final stage of SCRAM. This stage lasts longer than the previous stages and facilitates the seamless transition to production. The prototypes will be continuously updated to incorporate more functions in parallel with the continuous process of requirements refinement and validation. The prototypes are seen as the embodiment of the design space. The users will be required to participate in different roles as testers, validators and designers.

The application of the approach should be flexible, allowing regression to earlier stages if new features need to be explored further. For instance, if a new feature or an alternative interface needs to be analysed, story boards of alternative interfaces or simple demonstrators to explore new ideas can be implemented.

This process will continue until the end of the project, verifying requirements and usability cyclically with users. The expected final products from this process are a design document and advanced prototypes of the software – in our case, the BioVeL workflow web toolkit.

3 Application of SCRAM-CK

SCRAM-CK has been applied in the design and prototyping of the BioVeL web toolkit. This section describes the application of SCRAM-CK, outlining at each stage how the methodologies complemented each other and showing the evolution of the designs and requirements during the process.

The chronological presentation follows the staged application of the SCRAM-CK RE process. The participation of the users at every stage is reviewed to determine the actual role played in the process. This is also complemented by a summary of how the designs were evaluated and/or improved with the participation of the users.

3.1 Initial Requirements Capture

In this stage the design context is provided by the domain of application and the expected user base. The initial scenarios describe how the expected user-base will interact with the BioVeL software and the types of software provided, which is the perspective of the project sponsors and administrators. The scenarios were then validated by observing the work of actual users in the development of workflows and with initial interviews. At this stage, the only tools available for the users to inspect, edit and modify workflows were the myGrid software family (Taverna, myExperiment, and BioCatalogue). The effort needed for learning to use these tools requires a considerable investment of time from users.

3.1.1 Scenarios

BioVeL's expected user base is divided into three groups: (1) workflow experts, (2) workflow re-modellers, and (3) workflow consumers. This classification provides the initial scenarios:

- Workflow Experts typically need expressive, flexible tools for developing libraries of workflows that can be used by others. Such users' needs are satisfied directly by the myGrid software family (Taverna, myExperiment, and BioCatalogue). For this user group, BioVeL is adapting and customising the myGrid software family by promoting the creation of plugins and new services.
- Workflow Re-Modellers typically develop their own workflows, e.g., as variants of existing ones or based on libraries of workflow templates. These users do not need the full flexibility and complexity of a sophisticated workflow management system such as the Taverna Workbench. For this user group, BioVeL is deploying Taverna Lite, a web based application to simply assemble and adapt workflows using pre-defined templates, pre-existing workflows and components.
- Workflow Consumers typically use "pre-cooked, quality controlled" workflows through simple and sometimes data specific interfaces. For this user group, BioVeL is deploying Taverna Player and BioVeL Portal, web based applications for executing workflows and retrieving results.

3.1.2 Design Proposals

The application of SCRAM-CK in BioVeL is focused on addressing the needs of consumer and re-modeller users. For these user groups BioVeL aims to provide easy-to-use tools that make BioVeL services and workflows for data analysis and modelling readily accessible. The three applications that will cater for these users are the BioVeL Portal, Taverna Player, and Taverna Lite (Figure 3). These three applications form the core of the BioVeL web toolkit.

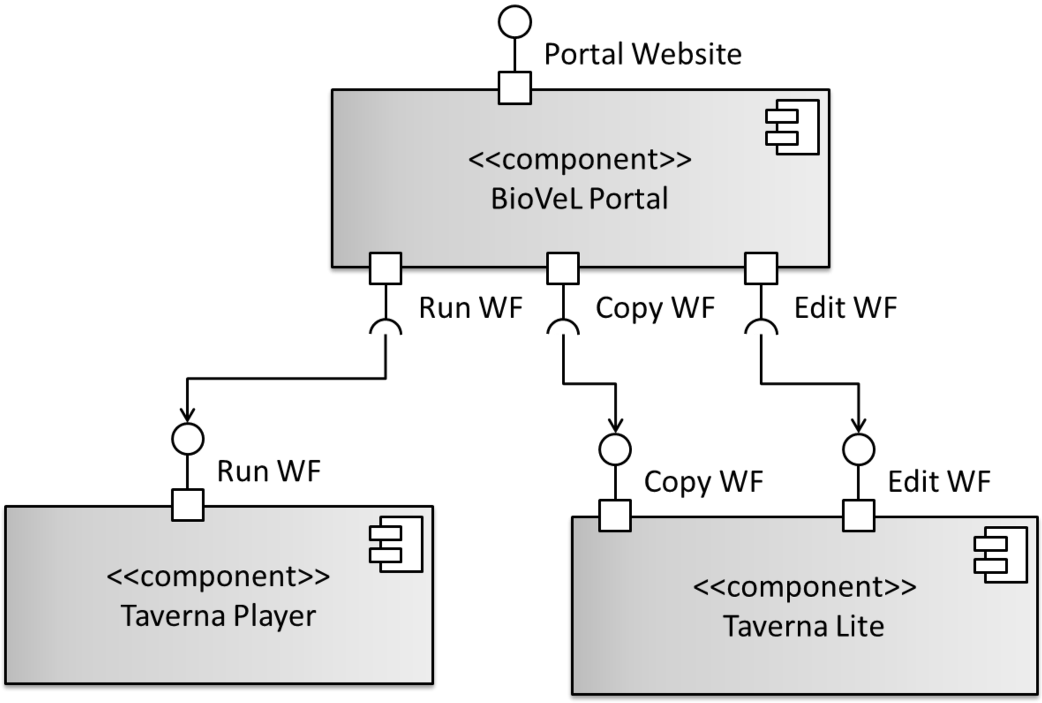

Figure 3 High Level Diagram of BioVeL Web Toolkit

3.1.3 Design Problems

The challenges associated with the introduction of scientific workflows to a research domain can be organised in four categories: facilitating the understanding of workflows, facilitating the use of the workflows, facilitating the adaptation of workflows for different needs and facilitating the construction of new workflows.

Understanding the design and operation of a workflow is a barrier for its use. Research suggests that workflow annotations help in describing the functionality of the workflows to reach a broader audience of potential users. Annotations also facilitate searching for workflows [10]. Workflow management systems (WFMS) facilitate the use of workflows. However, the complexity of WFMS is overwhelming and unintuitive for most new users [5]. WFMS also facilitate workflow construction and modification through graphical user interfaces. However, new users find the visual programming methods unfamiliar and complicated [42].

3.1.4 Design Space

The design proposals and the design problems for the adoption of workflows described above are represented using the CK diagram shown in Figure 4. The upper part of the diagram contains the knowledge propositions belonging to the Knowledge Space and the lower part of the diagram contains the corresponding propositions of the Concept Space.

The initial concept space can be expanded by applying partition (C -> C). In this case the different concepts evoke different methods for facilitating or overcoming each of the perceived barriers to workflow reuse. The first concept space contained five concepts (C_1 to C_5); further analysis of the documentation for the project objectives provided some ideas for the further expansion (C_6 to C_{15}).

The diagram shows the expansion of the initial concept space. Following CK theory, the next step involves trying to incorporate these new concepts in propositions of K, (conjunction operation explained in 2.2). This will cause the further expansion of the knowledge space. The propositions in K can then be mapped to requirements statements. For instance the proposition: “the WFMS must support displaying the workflow documentation to facilitate its understanding” is mapped to the requirement “Support improving workflow annotation”. However, the generation of knowledge statements such as these must be made in consultation with users, as discussed in the following sub-section.

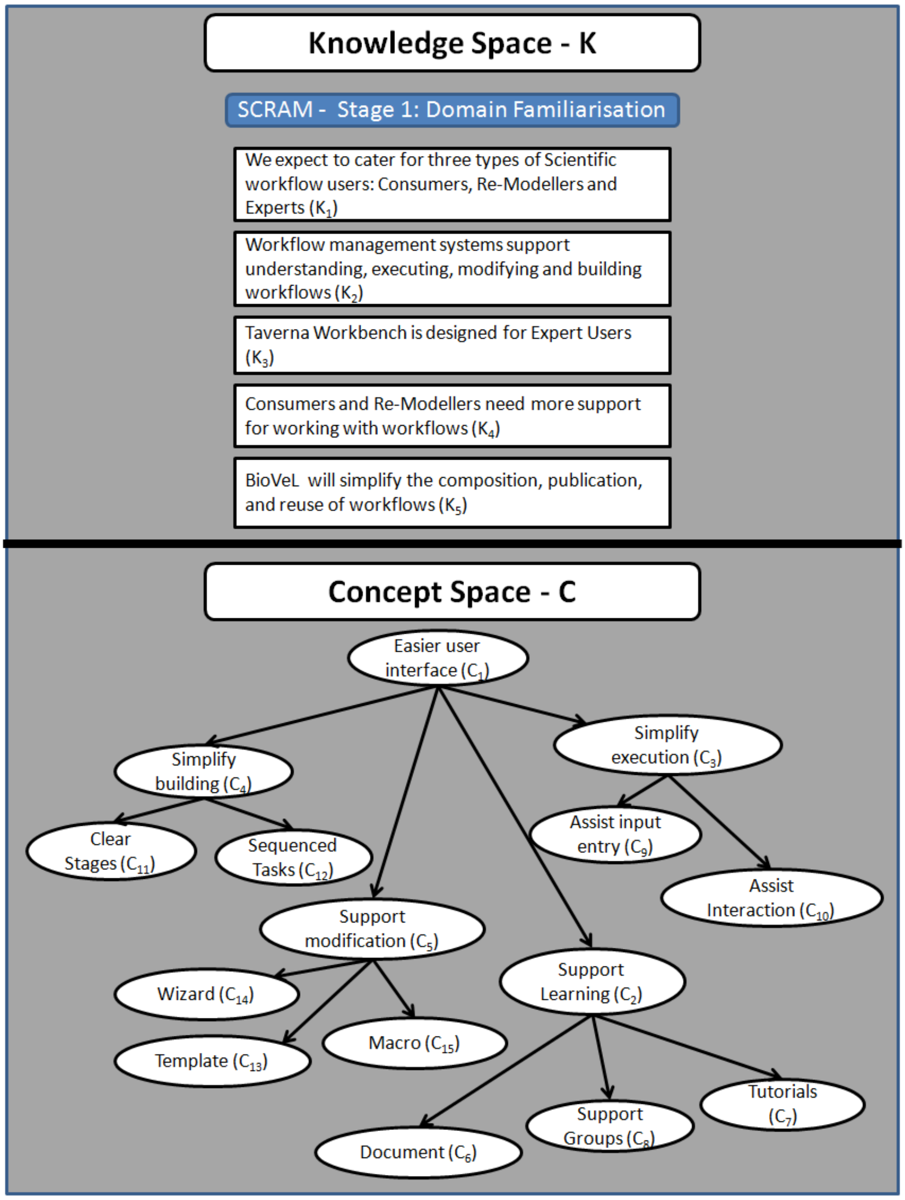

Figure 4 Stage 1 Domain Familiarisation. Initial Concept and Knowledge Spaces

3.1.5 Evolution of Design Space

Actual users' needs have to be compared to those envisioned by the mission statements for the project to separate real from perceived barriers for adoption. In the context of BioVeL, activities designed to allow this comparison are listed as follows.

- Analysing the work of the teams responsible for developing workflows.
- Reviewing the workflow building methods in BioVeL
- Observing online and face-to-face working sessions of the workflow building teams.

The most relevant issues when designing a workflow are those related to the input data formats, the expected transformations to be applied to that data and the expected outputs. Another relevant issue is the possibility of inspecting intermediate results when the analysis is not proceeding as expected, or in order to make choices about the next steps. The scientist may not have time or need for learning to configure a complex tool every time an analysis task is performed. From the scientist's perspective, the WFMS is mainly a tool for analysing data, rather than a workflow building tool. These concerns are reflected in the partition of the concept space presented expanding the branch from C_2. Figure 5 shows the derived concepts and the new set of knowledge propositions.

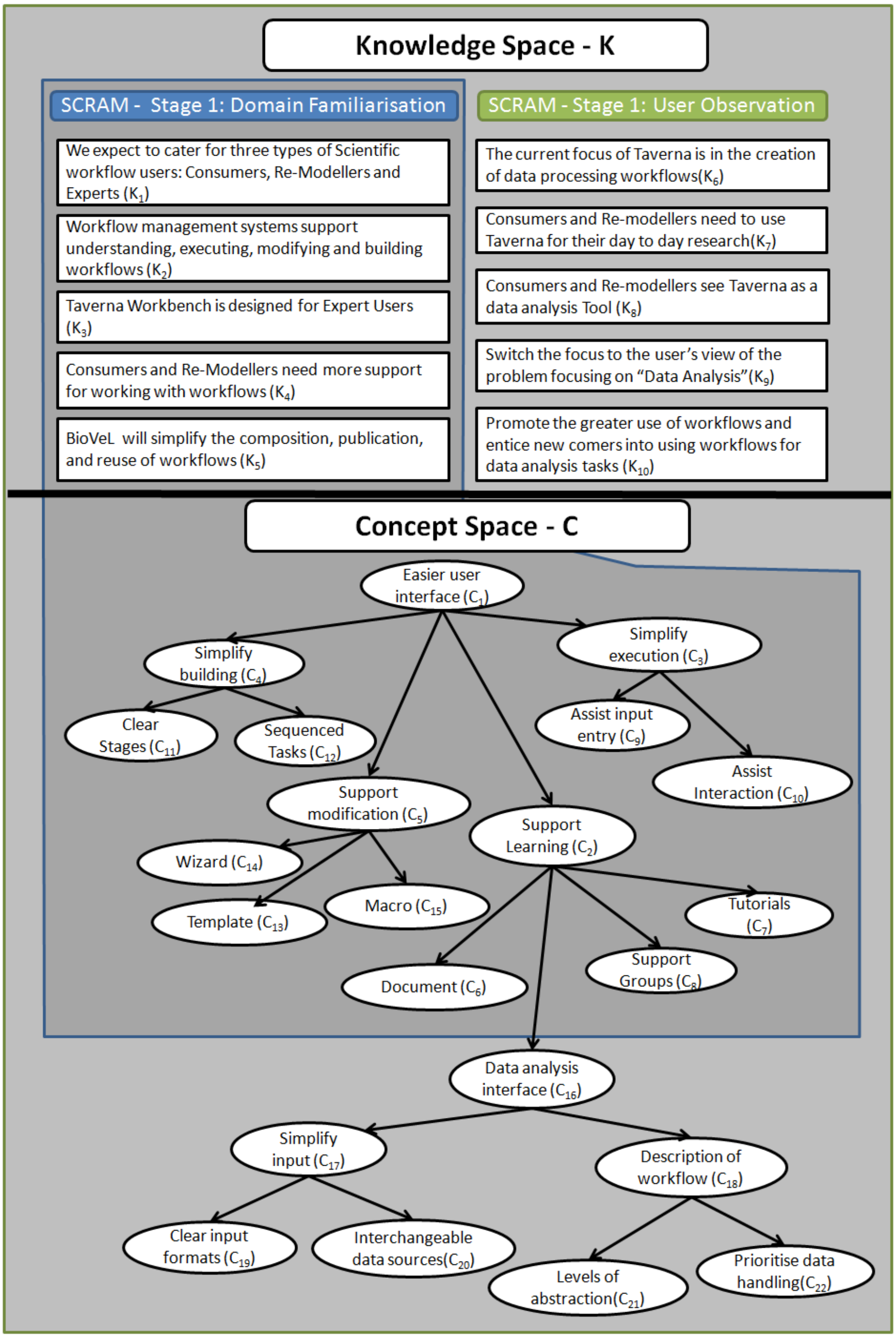

Figure 5 Stage 1 Domain Familiarisation and User Observation. Expansion of the Concept and Knowledge Spaces by Partition and Conjunction.

3.1.6 User Participation

The interpretation, observation and consultation activities helped focusing the initial stage of the requirements engineering process. The most revealing activities were those involving observing real users at work in three different settings. Firstly, users are observed indirectly by reviewing their work products and the tools they use in their day to day work. Secondly, they are observed as they participate in discussion between different service teams. Finally, they are observed and interviewed when working in online and face-to-face sessions. It was observed that BioVeL users have adopted several techniques for their workflow building process from specification, to design, building, testing, documenting and publishing. At this stage, the participation of the user is passive, users have not articulated their actual needs or discussed the possible solutions.

3.1.7 Requirements

The requirements gathered during the initial requirements capture and domain familiarisation stage suggest implementing a system that: emphasises the relevance of experimental data; describes the analyses that can be performed with a workflow clearly; and allows customisation of inputs. The most important features at this stage are those supporting understanding and running workflows.

1. Supporting understanding workflows
    a. Describe the analysis being performed
    b. Describe the required input and expected output data
    c. Describe intermediate processing of data
2. Support running workflows
    a. Facilitate finding a workflow
    b. Facilitate providing input data and retrieving results
3. Support modifying workflows (allow different input formats)
4. Support building workflows

3.2 Storyboarding and Design Visioning

The second stage of SCRAM deals with the further refinement of the requirements and requires greater user involvement. A focus group session was organised to explore different scenarios overcoming the perceived barriers to scientific workflow adoption (3.1.3 above). The objective of the focus group session was to analyse scientific workflow composition and usage scenarios with users.

3.2.1 Workflow Usage Scenarios

Seven scenarios were designed for displaying alternatives for workflow use ranging from complex workflow management systems to simple interfaces that only allow the selection and execution of a workflow (Table 1).

Table 1 Workflow usage scenarios

Scenario	Demonstrator	Emphasis
Development: Designed to show how a complex management system supports building, modifying and testing new workflows.	Used a modified version of the workflow development tutorial[1] for Taverna Workbench [13, 26, 35, 41]	Execution, building, and modification
Remodelling: Designed to show how social networking tools for sharing workflows and services can speed up the work of scientists by facilitating reuse of existing workflows as examples.	Used a modified version of the workflow development tutorial[2] to demonstrate searching for workflows, testing and modifying using Taverna and myExperiment [13, 26, 35, 41]	Learning, execution and modification
Execution: Designed to demonstrate the reuse of workflows via a user interface that hides the complexity of the underlying execution system.	MetFlow[3] portal and the Taverna Server Demonstrator[4] were used to show workflows published ready for execution with little or no configuration needed for running them.	Learning and execution
Data Centric: Designed to demonstrate how data analyses are performed by applying predefined functions and how the modifications are recorded for automatically building a workflow.	Demonstrated with the Galaxy 101 tutorial to show how workflow creation can be driven by data analysis [14]	Learning, execution, and building
Template: Designed to demonstrate assisted workflow composition. Users build a workflow by selecting services to fill activity slots on a workflow template	Demonstrated with the User Assisted Composition Tool [22, 23]	Building and modification
Example: Designed to demonstrate selection of workflows from a controlled repository and assisting in their execution and modification	Story Board supported with a mock-up application, a variation of the template approach [22, 23]	Learning, execution, and modification
Wizard: Designed to demonstrate the assisted step by step building of the workflow from a repository of workflow components	Story Board supported with a mock-up application, a variation of the template approach [22, 23]	Learning, execution, building and modification

[1] Original tutorial can be found at http://www.taverna.org.uk/documentation/taverna-2-x/tutorials/

[2] Original tutorial can be found at http://www.taverna.org.uk/documentation/taverna-2-x/tutorials/

[3] Metflow was a test project that enabled a simple web interface for running workflows. Source forge still has some references at: http://sourceforge.net/apps/mediawiki/metware/index.php?title=MetFlow, but the site http://msbi.ipb-halle.de/MetFlow/ used in May 2012 has now been decommissioned.

[4] The Taverna Server Demonstrator was a small web application used to demonstrate Taverna server up to version 2.3. GitHub still hosts the code at: https://github.com/myGrid/Taverna-Server-Demonstrator-Interface

The first four scenarios represent proposals to use/adapt existing software while the remaining three are new alternatives aimed at facilitating workflow building, reuse, and customisation.

3.2.2 Usage Design Proposals

The seven scenarios provide alternatives aimed at meeting the requirements gathered during the initial stage of SCRAM-CK. Some can deal with all the requirements while others just address some subset. The coverage of the requirements is highlighted in the last column of Table 1.

3.2.3 Usage Design Problems

The users who participated have been mostly using the Taverna WFMS to run workflows developed by other people. However, they expressed interest in learning to adapt and modify the workflows themselves. Thinking about the requirements for the development of the BioVeL web toolkit, a list of important features was produced and partitioned into two groups: current Taverna features to safeguard and features to improve.

The most important features to safeguard in the web toolkit are: flexibility, transparency, re-use and libraries. The group agreed that the flexibility for including different types of sources and analysis tools in workflows makes the WFMS an appealing research tool. For the user group, transparency means being able to "drill down" to inspect full details of workflows and "runs" as and when needed. For the users, the idea of reusing not only scientific results but also the tools for obtaining those results is a major selling point. Finally, having libraries of workflows and services that can be adapted to suit the user needs was highlighted as an incentive for using the web toolkit. The features that the users would like to see improved in the web toolkit include: error reporting and error handling, supporting the use of large data sets, run-time feedback, allowing some workflow customisation, improved output presentation, and simplified workflow execution.

3.2.4 Evolution of the Design Space

In this stage, most of the earlier requirements were confirmed as significant for the users. However, some new issues were also discovered. For this reason the design space needed was updated as shown in Figure 6.

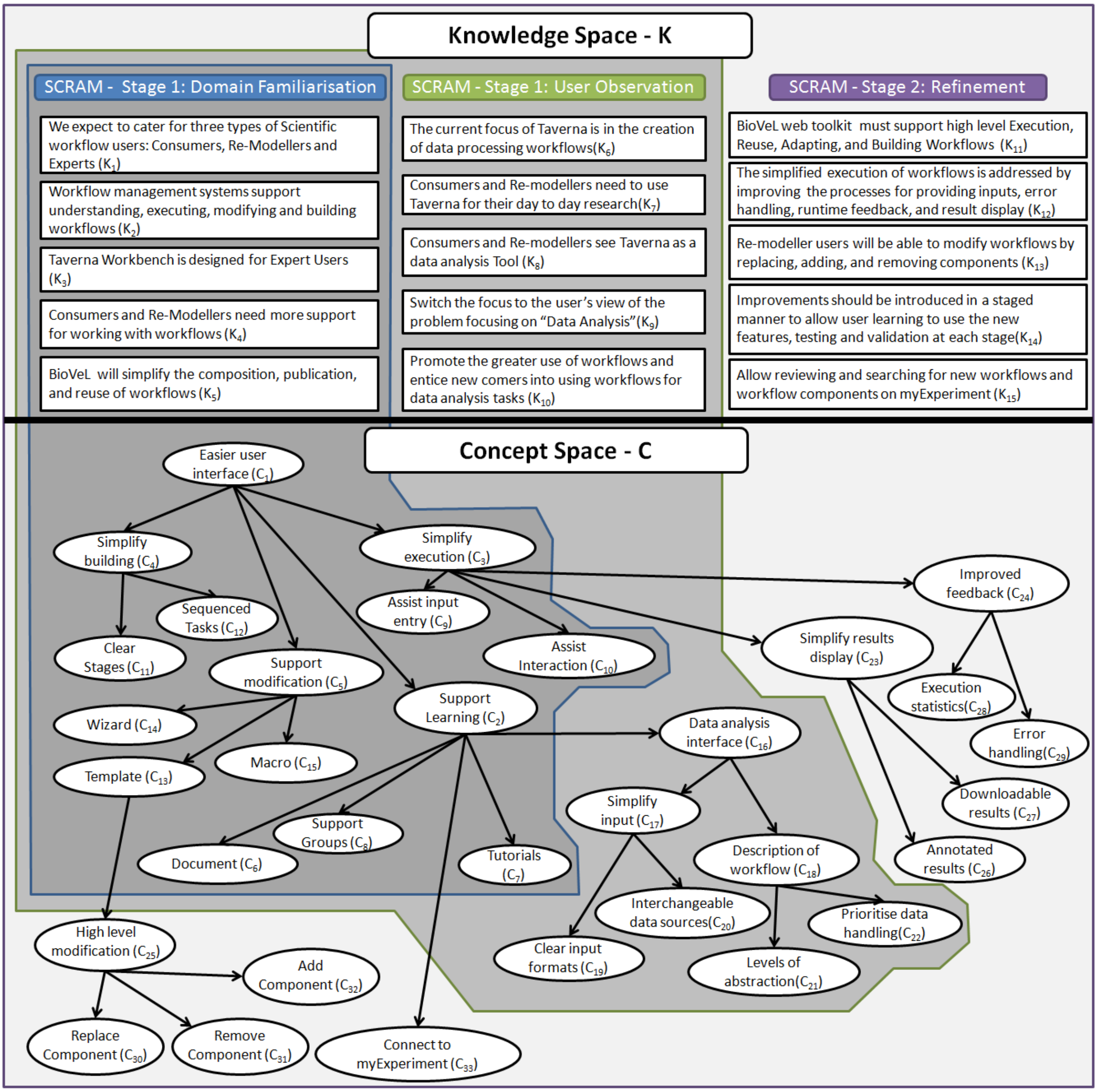

Figure 6 Stage 1 Domain Familiarisation and User Observation and Stage 2 Requirements Refinement. Third Expansion of the Concept and Knowledge Spaces by Partition and Conjunction.

The diagram in Figure 6 enables analysis of the entire structure of the design spaces and illustrates the interplay of CK and SCRAM. The diagram shows the evolution of the concept and knowledge spaces within the stages of the SCRAM-CK process, as is the case between the two expansions reported on stage one. Using this type of diagram also highlights the evolution of design across the stages of SCRAM.

3.2.5 User Participation

The participation of users during this stage was increased with the focus group session. The users who participated in the focus group session had already been using Taverna to run workflows developed in collaboration with their IT partners. The analysis of scenarios helped to facilitate their participation. The first two scenarios served as the catalyst for voicing their concerns, particularly those related to workflow execution and implementation. The remaining scenarios helped to introduce different alternatives with different levels of support for inspecting, executing, modifying and building. However, users expressed interest in learning to adapt and modify the workflows themselves, but stressed that the main issues to solve were those related to execution.

3.2.6 Refined Requirements List

The initial list of requirements was enriched with the results from this session. The ideas that were generated during the session were compiled and organised as the additional requirements for providing an easy-to-use web-based interface to learn, execute, build and modify workflows (Table 2).

Table 2 Refined requirements

Requirement	Learn	Execute	Build	Modify
Simplified workflow execution		X		
Presentation of results		X		
Downloading Results		X		
User management	X	X		
Workflow execution feedback	X	X		
Workflow execution statistics	X	X		
Implement connectivity with myExperiment	X	X		
Allow uploading sample input sets	X	X		
Allow downloading workflows	X	X		X
Implement error handling	X	X		
Customise inputs		X		X
Customise outputs		X		X
Implement template based customisation		X	X	X

Table 2 presents the requirements in the order of priority assigned by the users who participated in the focus group session (highest priority first). Users were informed that the intention of the session was to produce a requirements list to guide the implementation of the BioVeL workflow web toolkit. Although all the selected features integrate with each other for supporting the development of a web based WFMS, the group placed the highest value on supporting execution and facilitating documenting of the workflows for others to use.

3.3 Requirements Exploration

The third stage of SCRAM is aimed at further refining and validating the requirements for the BioVeL web toolkit. To accomplish this, three activities were carried out: an online survey on usability of Taverna, the development of a basic prototype to incorporate the first set of requirements, and the initial evaluation of the prototype by users.

3.3.1 Exploratory Scenario

The scenario for this stage matches the requirements derived during the second stage. In this case the prototype is presented as a potential solution for facilitating publishing and running workflows online. The main features to stress here are: ease of use, no need to configure any software, the possibility of downloading workflows for further inspection and the possibility of storing and managing results.

3.3.2 Exploratory Design

The priorities of the requirements list derived from the focus group session (Table 2 above) indicated the urgent need for a simple workflow execution tool. This was also supported by the feedback obtained from the responses to the usability survey. A prototype interface for running workflows was developed to further test the validity of the requirements and their importance to other users.

The prototype (Figure 7) was derived from a modified version of the Taverna Server Demonstrator[5] (TSD). The prototype included extended capabilities to manage users, upload workflows, preserve run results for longer than their expiry time on the server, and storing server credentials.

3.3.3 Prototype Evaluation

The basic functionality was exposed to a larger number of BioVeL partners during the annual project meeting. The demonstration was attended by scientists and IT technicians, partners of the BioVeL project. The prototype was demonstrated running actual BioVeL workflows. The observations from the participants coincided with the requirements previously discovered.

The users suggested new potential uses for the prototype in different scenarios such as facilitating training sessions with new workflow users; facilitating the presentation to third parties; an alternative for running and testing workflows; and as a tool for introducing students to use workflows. Enabling these scenarios required deployment of an improved version of the prototype online.

[5] https://github.com/myGrid/Taverna-Server-Demonstrator-Interface

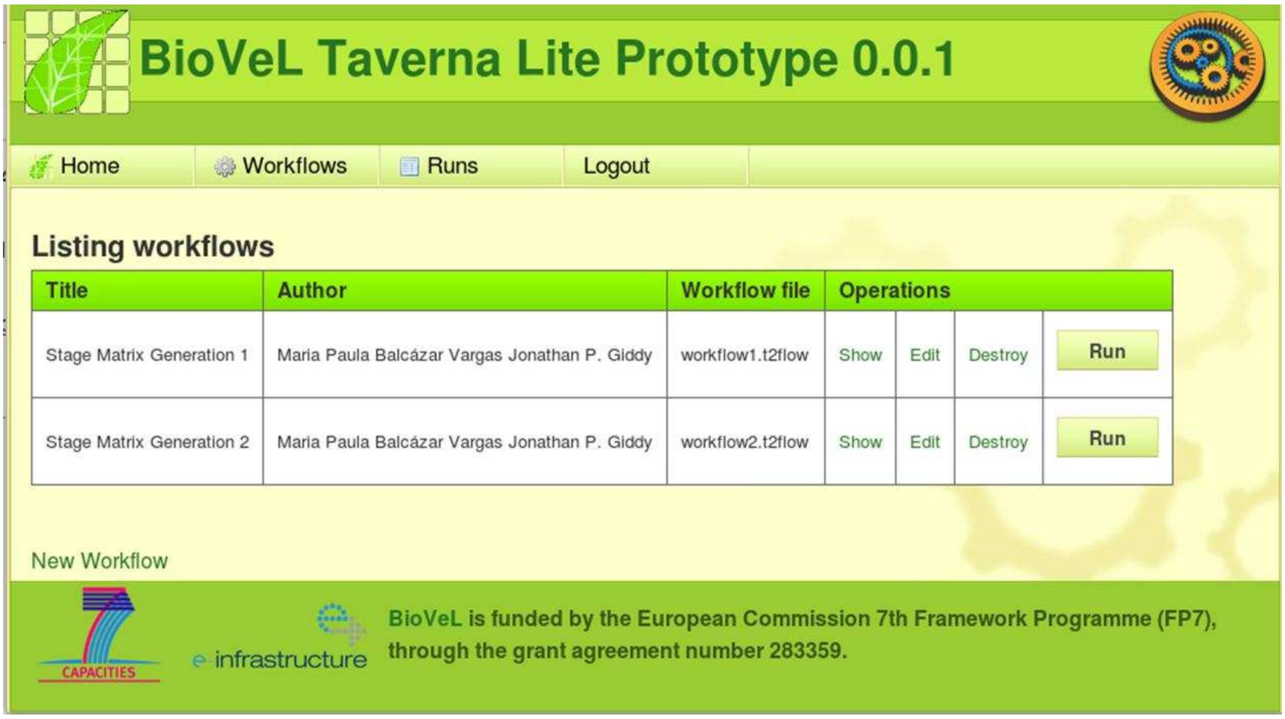

Figure 7 BioVeL Portal Prototype 0.0.1

3.3.4 Evolution of the Design Space

Most of the ideas brought forward during the exploratory evaluation session correspond to new concepts not contemplated before. These ideas are integrated into the design space as shown in Figure 8.

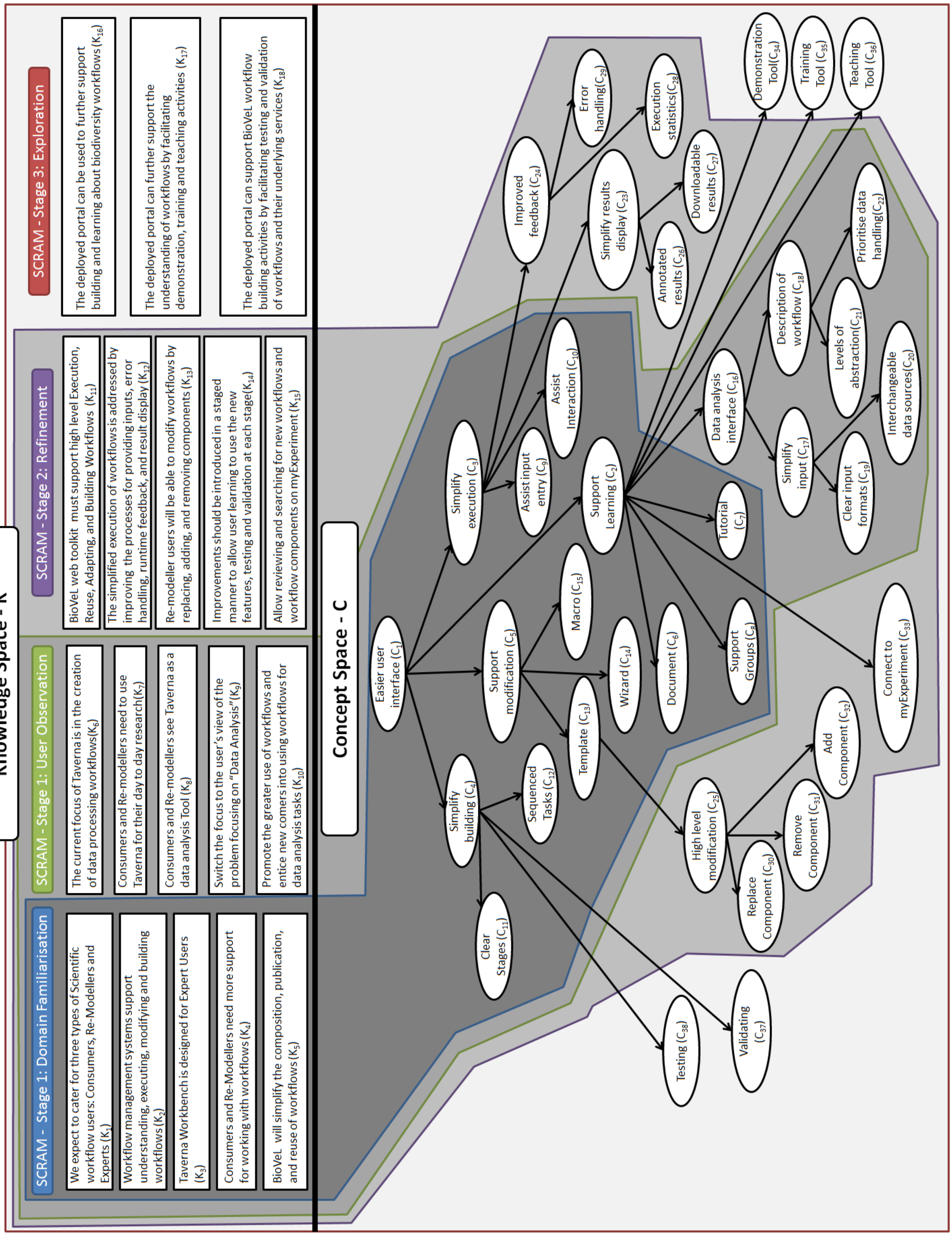

Figure 8 Stage 3 Exploration. Evolution of Design Spaces at the Third Stage of SCRAM-CK

3.3.5 User Participation

The users' role as designers was more prominent at this stage. This is evidenced by the new scenarios proposed by users which had not been contemplated before. Additionally, users participated in making a crucial design decision by requesting the deployment of the portal prototype. This design decision also influenced the prioritisation of the prototyping activities for the different components of the BioVeL web toolkit, to make them match the requirements priorities.

3.3.6 Revision of Requirements

During this stage only one requirement was added to the list from the previous stage, the deployment of the portal (Table 3). The ideas that were generated during the session were compiled and organised as the initial requirements for providing an easy-to-use web-based interface to learn, execute, build and modify workflows.

Table 3 Requirements Prioritisation and Mapping to Web Toolkit Components

Requirement	BioVeL Portal	Taverna Player	Taverna Lite
Simplified workflow execution		X	
Presentation of results		X	
Downloading Results	X		
User management	X		
Deployment of Prototype	X	X	
Workflow execution feedback		X	
Workflow execution statistics	X	X	
Implement connectivity with myExperiment	X		
Allow downloading workflows	X		
Implement Error Handling	X	X	X
Allow uploading sample input and outputs		X	X
Customise inputs			X
Customise outputs			X
Implement template based customisation			X

3.4 Prototyping and Requirements Validation

The prototyping and requirements validation phase was the longest of all the stages in the application of SCRAM-CK. This stage started with the deployment of Prototype 0.0.1 in October 2012 and concluded in January 2014, with the deployment of the last prototype in the series. During this period the versions of the prototype evolved from an easy to use interface for running Taverna workflows to an easy to use interface that included some workflow customisation and building functionalities. Each prototype in this series was deployed, evaluated and used constantly for demonstrations, workshops, training sessions, conferences and advance reviews.

3.4.1 Prototyping Scenarios

The deployment of the prototype and its successive refinements has facilitated the testing and evaluation using scenarios defined by BioVeL and scientists and IT experts. The four user defined scenarios are demonstration, outreach, development, and teaching.

Demonstration Scenario: the portal can be used to demonstrate the advances in the development of BioVeL workflows. This scenario has been used for formal reviews of the progress of the project.

Outreach Scenario: the portal can be used as an example of BioVeL working practices in presentations to third parties which are interested in collaborations with BioVeL.

Development Scenario: the portal can be used to facilitate testing and validation of workflows being developed within BioVeL.

Teaching Scenario: the portal can be been used in training sessions and workshops with potential new users.

3.4.2 Influence on the Design of Prototypes

The implementation of features scheduled in the previous stage (Table 3) has been carried out in parallel with the evaluation and use of the prototypes in the scenarios described above. These demonstrations have generated valuable feedback in the form of recommendations for usability improvements, new features to implement, and remarks on priorities for better supporting Biodiversity research (Appendix A).

3.4.3 Evolution of Design Space

The diagrams for representing the expansion of the Concept and Knowledge spaces have worked well for tracking the changes in the previous stages. However, the expansion of the design spaces is harder to monitor at this stage because the spaces grow faster and the increased number of concepts is harder to accommodate in simple diagrams. At this point a better tool for monitoring the evolution of the concept space and visualising it is required. Some type of notation could also be developed or adapted to cope with this.

3.4.4 User Participation

During this stage, the participation of users was more natural and spontaneous. User participation was monitored and demonstrated from the feedback, testing, usage, and demonstrating activities, many of which were started by their own initiative. At this stage, the project had become increasingly user-centered and user-driven. As a result, users participated more actively in reviewing the needs for the implementation of the new version of the BioVeL portal that eventually replaced the previous prototypes.

4 Evaluation

The goals of a RE method can be used as the frame to evaluate the method [3]. This type of evaluation is qualitative, not statistical nor experimental. The results of the evaluation help to determine how well the RE process meets the wider criteria in the RE domain and how well the RE process meets its own objectives [3].

4.1 Structuring the Evaluation

The evaluation of SCRAM-CK is structured around domain specific and general RE objectives. The evaluation process involves selecting the scope of the evaluation, identifying the main objectives to be evaluated, prioritising the criteria to be analysed, defining the data to be analysed, and comparing the results of applying the process to the target problem against other methodologies.

4.1.1 Scope of the Evaluation

It would be extremely complex to try to analyse all general aspects of SCRAM-CK as a RE method. Additionally, the main objectives of implementing SCRAM-CK have themselves a high degree of complexity. Accordingly, the evaluation presented here focuses mainly on whether SCRAM-CK has met the needs for BioVeL; that is, requirements gathering and enabling co-design supported by prototyping activities. Structuring the evaluation in this way focuses on the method and its contribution to support the RE process in the context of the BioVeL project. Additional issues, such as tool support, notation for requirements description, or production of design documents are not included because they are parallel activities which do not impact directly upon the evaluation of the method nor interfere/conflict with the methodologies being applied in the development and prototyping activities.

4.1.2 Evaluation Criteria

This evaluation of SCRAM-CK is structured in three stages. The first stage considers whether it covered the needs of the RE envisaged for BioVeL. As explained earlier, the ideal RE method for BioVeL needed a continuous process to facilitate user participation, design evolution, and prototyping. The second stage of the evaluation considers the extent to which SCRAM-CK covers the more general objectives of RE methodologies. These objectives include Pertinence, Correctness, Traceability, and Understandability [3]. Finally, the evaluation covers the comparison against other methods using the previously established domain specific and general criteria of the two previous evaluation exercises.

4.1.3 Prioritisation of Objectives

In addition to the scale for measuring the achievement of each method, it is also necessary to determine the priorities of each of the domain specific and general objectives with which the method is evaluated. In this case there are two groups of objectives: the domain specific objectives that motivated the development and use of SCRAM-CK and the general RE objectives which any RE method must aim to fulfil. From the viewpoint of the BioVeL project, the weight of the first set of objectives is greater because the main goal of implementing SCRAM-CK is to support a specific set of goals. From the viewpoint of RE practice, the general RE objectives are crucial to determine if the new method offers some form of advantage to other practitioners over existing methods.

4.1.4 Data Sets

The data to be interpreted is composed of the feedback from users testing, using and reporting bugs or problems with the different prototypes. The data is interpreted by the researchers and then grouped according to the objectives being evaluated. There is a risk of biasing the study as the users will focus more on the issues that are immediately urgent for their own needs while neglecting the ones that they deem secondary.

4.2 Evaluation in Line with BioVeL Objectives

The evaluation of SCRAM-CK needs to consider whether it achieved the initial prerequisites of a continuous RE process to facilitate user participation, design evolution, and prototyping.

4.2.1 User-Participation

In the context of producing useful tools for a scientific community, the users have participated in the design of the BioVeL web toolkit as designers. The prioritisation of requirements and the selection of most suitable interfaces was user-led. Users participated as designers by guiding the production of prototypes, selecting the functionalities to be supported and suggesting additional functionalities that the technical people had not considered or presented as options. Users have also been very clear when pointing out the most important shortcomings of the existing technologies from their point of view, and they also suggested similar tools to use as examples.

There are three key points supporting the assertion that the SCRAM-CK is user-centered:

a) Prototyping and release of the applications making up the BioVeL web toolkit was delivered and user tested earlier than expected. This was a direct response to early demonstrations in which users requested a simple method for showcasing their advances in Workflow building.

b) Testing prototypes occurred in settings that pushed the software capabilities to their limits very rapidly. This fostered the revision of the underlying structures supporting the prototypes, from the kind of machines selected for deployment to the considerations and decisions about the distribution of supporting software (servers, applications and supporting services).

c) The prototypes served to promote the BioVeL toolkit with third parties outside the project. The opinions and needs of those third parties were also incorporated, when possible, into the requirements set, given that they represented the needs of the wider biodiversity community, beyond the limits of the project's consortium.

In the first case, the prototyping activities for Taverna Player and the BioVeL portal were brought forward six and twelve months respectively. This accelerated the refinement both of the tools and of their integration into prototypes that evolved through seven design-develop-test-deploy iterations. These prototypes reached a level of maturity that facilitated their use as design models for the actual construction of the Player and Portal Applications. This process has helped far more than if we had just delivered the requirements list and written design specifications.

In the second case, users have participated as testers of the prototypes, using workflows relevant to their respective research specialities. Testing has promoted the inclusion of functionalities not contemplated in the early stages of the requirements engineering process. The prototypes turned out to be a very useful tool for testing and demonstrating new workflows. This validates the initial request for easier means to run workflows as a high priority.

Finally, the users have been scheduling separate workshops to showcase their workflows using the prototypes. This was done with the intention of disseminating their work amongst peers, but has had an additional positive effect as a source of further third party testing in new scenarios. The users have been reporting the results of those workshops and bringing back new sets of requirements based on actual demands of their respective biodiversity specialities.

The Requirements Engineering process reported here ran for twenty two months, from April 2012 to February 2014, i.e. until the end of the prototyping cycle. Different documenting and monitoring activities tracked the evolution of the requirements process and also the evolution of design propositions. Throughout this process, the requirements list was not considered an immutable set of clauses that needed to be abided to but guidelines that were enhanced in response to the needs of the users and developers.

4.2.2 Evolution of Requirements and Designs

The mapping between requirements and design decisions can be traced using the CK diagrams. In this project the requirements fall within the concept space and the design decisions in the knowledge space. This shows that although many requirements have been discovered, the prototyping activities only implement those requirements that are part of design decisions. The design evaluation process also evolved from paper prototypes and mock-ups, to functioning prototypes up to advanced prototypes deployed using Amazon Web Services[6]. The different stages in which the method was applied, and the diagrams showing the expansion of the design space, provide a wider view of the support that SCRAM-CK provides for the interweaving in the evolution of design and requirements.

4.2.3 Prototyping

Rapid software prototyping can be viewed as an evolutionary process in which early prototypes are continuously refined to adequately define and validate user requirements [27]. In addition to being a medium for requirements refinement and validation, early prototyping activities provide added value as a medium that facilitates user engagement, exploring emerging opportunities, negotiating the reconfiguration of roles and resources and early identification of emerging problems that could impact usability [40]. These effects were observed to different degrees within the requirements gathering process carried out as part of the BioVeL research into the development of easier user interfaces for online execution and modification of scientific workflows.

The series of prototypes provided users with a solid base for discussion not only of initial requirements but also of relevant usability issues. There were more than 15 online sessions (“Play Days”) in which the evolving prototypes were the medium for demonstration and discussion. Usability is an important non-functional requirement [20]. In addition to usability, the use of SCRAM-CK facilitated the discussion of other non-functional requirements (NFR) such as reliability, speed, responsiveness and dependability of the system at an early stage. For instance, users quickly pointed out to the use and placement of buttons to facilitate data input and control, suggesting not only color schemes but also the relocation of buttons on the page to improve user

[6] This was the platform where the BioVeL web toolkit was first deployed (http://aws.amazon.com/)

experience. Users also discovered the need for improving the response time after submitting a job and the use of state messages to reassure the user that the job was being processed.

The prototypes also served as an early integration test for the components – a test which otherwise would have been made later in the project. The late detection of integration issues had the potential of affecting target dates and delaying dissemination activities. In this sense, early testing of scalability and resourcing was beneficial. For instance, the resources needed for multiuser execution pointed out to the need for restructuring the underlying infrastructure, leading to a revised infrastructure dependent on diverse server programs such as Google Refine, R Server, Taverna Server, WebDav and Atom. For the initial prototypes the small AWS machines were sufficient for single user testing but once the number of users started to increase, so did the problems of limited hardware; which prompted early identification of the need to improve the underlying resourcing architecture, providing a more robust architecture for the deployment platform.

In terms of usability testing, the users coped well with the limitations of the prototypes, but also identified several significant shortcomings. As a result, they came up with examples and suggestions that made possible the redefinition of the user interface to better support their needs.

For these reasons, we may conclude that the experience of applying SCRAM-CK to the design and prototyping activities for delivering easier user interfaces for online workflow execution and modification was not an added cost to the project, but an essential activity which benefited the whole project in more than one way, and which avoided higher re-work costs later in the project. However, the findings about NFR must be taken into consideration in future applications of the model, especially if the prototype is expected to evolve into a fully-fledged production system.

4.3 Evaluation against RE Objectives

The evaluation of SCRAM-CK as a general purpose RE method needs to consider whether it achieved the goals that most RE methods aim to cover. These goals, as mentioned before, are pertinence, correctness, traceability and understandability.

4.3.1 Pertinence

Pertinence is a major RE objective and it refers to the avoidance of redundant requirements [3]. Using CK diagrams allowed structuring the design space in a way that made visible the relationships between the concepts in the design space and the statements that supported the requirements lists.

This helped avoid the presence of redundant requirements. For this reason this RE objective is deemed to be fully achieved in this case.

4.3.2 Correctness

Correctness is an RE objective that refers to the completeness, consistency and validity of the requirements produced [3]. The requirements for the production of easier user interfaces for executing, inspecting and modifying workflows were derived iteratively from several interactions with users and developers. The requirements appear to be complete as they cover most of the needs of users. However, measuring completeness is not straightforward, as different users have different requirements. They reached a general consensus on the main features but each pressed for different aspects that favoured their work. For instance, while most of the users did agree on the need for an easier medium to run workflows for testing, validating and demonstration purposes, not all agreed on the urgency of advanced features for modifying workflows. This also affected consistency and validity directly since the priorities assigned to different requirements sometimes clashed with user expectations. Within the constraints of the project this objective was partially fulfilled but there is no reason to believe that the method cannot completely fulfil this objective in other circumstances.

4.3.3 Traceability

Traceability involves tracking the relationships between requirements and their sources to clearly identify the origin of the requirements in order to manage requirements easily [3]. Traceability was used to rank and to validate the requirements at each stage. If a requirement was common to a greater group of users, then the ranking was higher and similarly, the requirements that had less demand were pushed down in the priorities list. The use of interviews, meetings, and the project's bug tracking system facilitated the direct identification of the user groups that were behind specific requirements. This helped in always being able to identify the source and rationale for each requirement. This objective is fully achieved.

4.3.4 Understandability

Understandability is an important RE objective in all life cycle of the requirements that facilitates the discussion of requirements with all stakeholders [3]. In this case, requirements were expressed in simple terms as statements in a ranked list. The paper prototypes, prototype interfaces, interview reports, video, and responses to surveys were used to clarify the details of the requirements. This helped not only to keep the requirements list clear and concise, but also to minimise the amount of documentation produced. The staged implementation of requirements and the agile development techniques applied also helped in keeping the requirements fresh and understandable to all stakeholders. This objective can therefore be considered to be fully achieved

4.4 Alternative Theories

The existence of several alternative RE methods leads to the question: is it appropriate to propose yet another method? First, it should be noted that any method selected would have had to be adapted to match the specific goals of BioVeL. After considering various alternatives, four other RE methods were considered good candidates to address the requirements for BioVeL: GORE, COSMOD-RE, ScenIC, and ARID. This section describes these methods and provides a brief comparison to SCRAM-CK in terms of their coverage of the RE process envisaged for BioVeL. The coverage of general RE objectives is not analysed here because that would have required more experience with using each of the methods.

4.4.1 COSMOD-RE

COSMOD-RE (sCenario and gOal based SysteM develOpment methoD) [30] advocates co-design which is defined as mixing of requirements and architecture design. In the COSMOD-RE process goals, scenarios, and architecture are continuously refined. COSMOD-RE is structured in five stages. However, the user participates in the initial stage for requirements specifications. The remaining stages involve making adjustments to reconcile requirements and architectural needs, with little participation from users. COSMOD-RE organisation mirrors the organisation of a layered architecture. The concerns of users are mostly addressed in the design of the interface layer and then mapped down through the different layers through to the hardware level.

4.4.2 GORE

Scenarios and Goal-Oriented Requirements Engineering (GORE) can be used to achieve a stable set of requirements [9]. There are several approaches that aim at combining GORE methods with scenarios in requirements engineering. Goal decomposition guides the discovery of requirements by means of continuous decomposition of higher order goals into more specific ones. GORE techniques do not address the active role of users in the design process and therefore GORE is not, in itself, a suitable methodology for projects where co-design is an important element.

4.4.3 ScenIC

ScenIC is a requirements engineering method for evolving systems [31]. ScenIC was derived from the Inquiry Cycle [34] model of requirements refinement and uses goal refinement and scenario analysis as its primary methodological strategies. This method can be used during the entire requirements process to analyse and validate requirements. The problem is that it does not provide a structure for conducting the process itself, suggesting just a constant repetition of the verification cycle. ScenIC does not address the changing nature of requirements as the project progresses and it does not envision the use of different strategies at different stages. Users are interviewed and observed for requirements elicitation but they do not take an active role in the process.

4.4.4 ARID

Active Reviews for Intermediate Designs (ARID) [7], is a method developed for evaluating a portion of a system, for instance the user-interface. ARID is a hybrid method that allows reviewing a design in its pre-release stages. ARID provides early insight into the design's viability, and would allow for discovery of errors, inconsistencies, or inadequacies. It is user centered, allowing users to participate not only in the evaluation of an artefact but also in the specification of the success criteria. However, ARID does not address analysis of the actual design alternatives or the evolution of the design, leaving them as additional activities after the evaluation process.

5 Discussion and Conclusions

It would be inaccurate to claim that all the positive results achieved have been the product of the SCRAM-CK process. The process has also been helped by the continuous exposure of the scientists to other design/development processes in the project. Firstly, they have received training on workflow building, which has made them aware of many of the needs for clearly designing and describing a procedure that requires automation. Secondly they have participated in the definition, selection and/or design of many of the processing steps that are integrated in their workflows. Thirdly they collaborate closely with developers that have been helping them in designing and building workflows for their respective research areas. These activities have been more rigorous because one of the objectives of BioVeL is to produce repositories of high quality workflows. As a result, the scientist users within BioVeL have demonstrated a better grasp of the concepts related to software design and development than one might normally expect of end-users. This section will elaborate on some of the main issues found during the RE exercise applying SCRAM-CK. An extended discussion about the prototyping activities is also included.

5.1 Main Issues

The use of SCRAM-CK has supported the development of the BioVeL web toolkit. SCRAM provided the general structures for the requirements engineering activities, while CK Theory has provided the logical underpinnings to map the evolution of design in different stages of the process. However, there are issues related to the actual mechanisms for monitoring evolution, documentation of changes and implementation time that need to be addressed in future if the approach is used again.

The documentation required for the application of SCRAM-CK is minimal. Apart from the diagrams and the requirements lists, the issues were easily recorded by users themselves in the project issue tracking system (JIRA). This facilitated a fast turnaround whilst also providing sufficient documentation of issues and their solutions. The analysis of (JIRA) issues helped in determining whether an issue reported was a new feature, a request for improvement, or a bug.

The diagrams for representing the expansion of the concept and knowledge spaces work well for small design tasks. The diagrams are presented in this article together in the different figures in order to illustrate the evolution of the design space. However, as the presentation of stages advances, the diagram complexity increases rapidly; complicating the building and maintenance of the diagrams. When viewed as a whole they are hard to follow and maintain. However, they can be kept separated by stages, types of concerns addressed or design session, to help readability and to facilitate interpretation.

Time constraints were particularly challenging. The initial proposal of the method and its early implementation in practice were carried out within four months. Similarly, the prototyping phase started from month five. More time is required to incorporate and apply the methods with support tools and notations for verifying requirements consistency and completeness. Nevertheless, the application of SCRAM and CK Theory method integrated easily with the agile development practices already in place within the project.

Another issue related to time constraints is the speed with which the proposed process was implemented and executed. This leaves room for questioning the selection of methods and theories. However, we based our selection on previous experience in similar projects and on sound guidance for selecting requirements engineering methodologies [2, 6, 9, 17, 24, 25, 28, 29, 39, 43]. The complementary review of technologies suggested that our candidate methods were a good match to the requirements of the project and the practical application confirms this.

The constant maintenance and implementation of the requirements list has been used as the foundation for the implementation strategy. The simple requirements list is not the optimal tool for describing the requirements in full. However, design decisions and requirements can be mapped using the CK diagrams in combination with this list. To complement this, the development teams have actively used the BioVeL project's issue tracking system (JIRA) where features are logged as improvements to the products under development.

5.2 Conclusions and future work

Virtual e-Science platforms (or Virtual Research Environments) are intended to support scientific discovery by enabling the cooperative work of diverse users over long periods of time, and bringing together of distributed resources [43]. The development of such platforms requires scalable methods of requirements analysis that: document the needs of vastly different user groups; continue to document changing needs over time; coordinate investigation at multiple sites of use; design for large distributed entities; and absorb transformative changes in practice [43]. An agile requirements engineering process, coupled with strong user engagement that can adapt to changing circumstances as project and users' needs unfold, is key to meeting these challenges [35]. The SCRAM-CK process presented in this paper is an approach that may be used to face those challenges.

The BioVeL project has implemented a requirements review process that is intertwined not only in the design and development processes of the software products, but also into the wider scientific support activities of the project. The requirements review process fostered the inclusion and engagement of end users in the design processes in a way that allowed them to take ownership of the resulting product and steer its design and development activities. This is a particularly important factor in fostering the uptake of new technologies that have the potential to change working practices.

The prototypes of the BioVeL web toolkit were developed using the SCRAM-CK process. This process produced working prototypes, which were effective tools for discussing and enhancing the requirements as these were implemented as features. The activities showed BioVeL as a viable e-Science platform, meeting the need for a tool for workflow inspection, execution and modification. The current version of the BioVeL workflow web toolkit is a set of Ruby on Rails applications deployed at http://portal.biovel.eu. The source code is openly available on github[7] [8] [9]. BioVeL also provides an Ansible[10] configuration management module to facilitate the rapid deployment portal instances.

[7] https://github.com/BioVeL/portal

[8] https//github.com/ myGrid/taverna-player

[9] https://github.com/myGrid/taverna-lite

[10] https://github.com/BioVeL/ansible-playbooks

The prototyping phase ended in February 2014. The prototypes evolved in controlled stages increasing functionalities. Each new version of the prototype supported more complex workflow manipulation operations. The last prototype delivered included the ability to modify workflows through Taverna Lite..

As a result of this controlled evolution, the software became increasingly robust, due to the fast feedback loop and the debugging and testing activities. At present the prototype versions have been decommissioned and a new version of the portal is being supported. This version is stronger from the beginning thanks to feedback and design insights gained through the development, deployment, and use of the prototypes supported by the SCRAM-CK approach.

In future evaluation exercises, the mapping of the expansions to the concept and knowledge spaces needs to be refined to better support the derivation of requirements. It will be interesting to analyse further the applicability of design games for scenario evaluation, particularly when different levels of customisation need to be implemented; as well as trying to apply alternative workshop facilitation techniques [4] and design review methods [7].

The application of SCRAM-CK suggests that the use of prototypes can also help in fleshing out other NFR at an early stage. This is an alternative to the approaches that promote determining the NFR after the functional requirements are defined [19, 33]. In this case using an ontology based NFR elicitation framework, such as ElicitO [1], in combination with SCRAM-CK could provide a more complete set of requirements. This would be an important enhancement with the potential of easing the transition from prototyping to development of high quality production applications.

6 Acknowledgements

The BioVeL project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no. 283359. More information at http:://www.biovel.eu.

The authors would like to thank the members of BioVeL who have actively participated in the requirements review activities reported in this paper, including our scientific partners: María Paula Balcazar Vargas, Sarah Bourlat, Päivi Lyytikäinen-Saarenmaa, Matthias Obst, Gerard Oostermeijer, Elisabeth Paymal, Hannu Saarenmaa, and Saverio Vicario; and our technical partners: Jonathan Giddy, Carole Goble, Robert Haines, Vera Hernandez, Robert Kulawik, Cherian Mathew, and Alan Williams.

The authors would also like to thank all the participants in the different training and dissemination activities, who have provided invaluable comments and feedback.

The authors also thank the anonymous reviewers for their feedback that helped to improve the overall quality of the manuscript.

7 References

1. Al Balushi, T. H., Sampaio, P. R. F., Loucopoulos, P. (2013). Eliciting and prioritizing quality requirements supported by ontologies: a case study using the ElicitO framework and tool. Expert Systems, 30(2), 129-151.
2. Alexander I., Beus-Dukic, L. (2009) Discovering requirements: how to specify products and services. Chichester : Wiley, 2009. 457 p.
3. Al-Subaie, H. S. F., Maibaum, T. S. E. (2006). Evaluating the Effectiveness of a Goal-Oriented Requirements Engineering Method. Fourth International Workshop on Comparative Evaluation in Requirements Engineering (CERE), September 2006. Pages 8-19. DOI: 10.1109/CERE.2006.3
4. Azadegan, A., Papamichail, N., Sampaio, P. (2013). Applying collaborative process design to user requirements elicitation: A case study. Computers in Industry, In Press. doi:10.1016/j.compind.2013.05.001
5. Bhagwanani, S. (2005). An Evaluation of End-User Interfaces of Scientific Workflow Management Systems. Master Thesis. North Carolina State University.
6. Cheng, B. H. C., Atlee, J. M., Joanne, M. (2007). Research Directions in Requirements Engineering Research Directions in Requirements Engineering. Proceeding FOSE '07 Future of Software Engineering (pp. 285–303). Minneapolis, MN: IEEE Computer Society Washington.
7. Clements, P. C. (2000). Active Reviews for Intermediate Designs (CMU/SEI-2000-TN-009). (pp. 1–25). Software Engineering Institute, Carnegie Mellon University. Retrieved from http://www.sei.cmu.edu/library/abstracts/reports/00tn009.cfm
8. Davis, F. D. (1989) Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. MIS Quarterly, 13:3, 319-340. (http://hcibib.org/perlman/question.cgi?form=PUEU)
9. Delnooz, C., Vrijnsen, L. (2003). Experiences with scenarios and Goal-Oriented RE. LACS2003, Landelijk Architectuur Congres.
10. Garijo, D., Alper, P., Belhajjame, K., Corcho, O., Gil, Y., Goble, C. (2012). Common motifs in scientific workflows: An empirical analysis. 2012 IEEE 8th International Conference on E-Science, 1–8. doi:10.1109/eScience.2012.6404427
11. Gil, Y., Deelman, E, Ellisman, M., Fahringer, T., Fox, G. (2007). Examining the challenges of scientific workflows. Computer.
12. Goble, C., De Roure, D. (2009). The impact of workflow tools on data-centric research. Data Intensive Computing: The Fourth Paradigm of Scientific Discovery, Microsoft Research, 2009

13. Goble, C., Wroe, C., Stevens, R. (2003). The myGrid project: services, architecture and demonstrator Grid Services and Architec-. In S. Cox (Ed.), Proceedings of UK e-Science All Hands Meeting 2003 (p. 8). Southampton: University of Southampton.
14. Goecks, J., Nekrutenko, A., Taylor, J. (2010). Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome biology, 11(8), R86. doi:10.1186/gb-2010-11-8-r86
15. Hatchuel A., Le Masson P. Weil B. (2004), CK Theory in Practice: Lessons from Industrial Applications, 8th International Design Conference, D. Marjanovic, (Ed.), Dubrovnik, 18th-21st May 2004: 245–257.
16. Hatchuel, A and Weil, B (2003), A new approach of innovative design: an introduction to CK Theory. Proceedings of the international conference on engineering design (ICED'03), Stockholm, Sweden, pp 109–124
17. Jarke, M., Loucopoulos, P., Lyytinen, K., Mylopoulos, J., Robinson, W. (2011). The brave new world of design requirements. Information Systems, 36(7), 992–1008. doi:10.1016/j.is.2011.04.003
18. Le Masson, P., Hatchuel, A., Weil, B. (2009). Design Theory and Collective Creativity: A Theoretical Framework to Evaluate KCP Process. Proceedings of ICED 09, the 17th International Conference on Engineering Design, Vol. 6, Design Methods and Tools (pt. 2), Palo Alto, CA, USA, 24.-27.08.2009 (277-288)
19. López, C., Codocedo, V., Astudillo, H., Cysneiros, L. M. (2012). Bridging the gap between software architecture rationale formalisms and actual architecture documents: An ontology-driven approach. Science of Computer Programming, 77(1), 66-80.
20. Loucopoulos, P., Sun, J., Zhao, L., Heidari, F. (2013). A Systematic Classification and Analysis of NFRs.
21. Marx, V. (2013). Biology: The big challenges of big data. Nature, 498, 255–260.
22. Mehandjiev, N., Lecue, F., Wajid, U., Namoun, A. (2010). Assisted Service Composition for End Users. 2010 Eighth IEEE European Conference on Web Services (pp. 131–138). IEEE. doi:10.1109/ECOWS.2010.30
23. Mehandjiev, N., Namoune, A., Wajid, U., Macaulay, L., Sutcliffe, A. (2010). End User Service Composition: Perceptions and Requirements. 2010 Eighth IEEE European Conference on Web Services (pp. 139–146). IEEE. doi:10.1109/ECOWS.2010.29
24. Nuseibeh, B., Easterbrook, S. (2000). Requirements Engineering : A Roadmap. In C. Ghezz, M. Jazayeri, A. L. Wolf (Eds.), Proceedings of the 22nd International Conference on on Software Engineering, ICSE 2000 (Vol. 1, pp. 35–46). Limerick Ireland.
25. Nuseibeh, B. (2001). Weaving Together Requirements and Architectures. Computer, 34(3), 115–119.
26. Oinn, T., Li, P., Kell, D. B., Goble, C., Goderis, A., Hull, D., Stevens, R., et al. (2007). Taverna / my Grid : Aligning a Workflow System with the Life Sciences Community. In I. J. Taylor;, E. Deelman, D. B. Gannon, M. Shields (Eds.), Workflows for e-Science (pp. 300–319). London: Springer London. doi:10.1007/978-1-84628-757-2

27. S. P. Overmyer (1991). Revolutionary vs. Evolutionary Rapid Prototyping: Balancing Software Productivity and HCI Design Concerns. Center of Excellence in Command, Control, Communications and Intelligence (C3I), George Mason University, 4400 University Drive, Fairfax, Virginia 1991.
28. Parviainen, P., Tihinen, M. (2007). A survey of existing requirements engineering technologies and their coverage ¨. International Journal of Sortware Engineering and Knowledge Engineering, 17(6), 827–850.
29. Peirce, S. C., Hardisty, A. R., Preece, A. D., Elwyn, G. (2011). Designing and implementing telemonitoring for early detection of deterioration in chronic disease: defining the requirements. Health informatics journal, 17(3), 173–90. doi:10.1177/1460458211409717
30. Pohl, K., Sikora, E. (2007). COSMOD-RE: Supporting the Co-Design of Requirements and Architectural Artifacts. 15th IEEE International Requirements Engineering Conference (RE 2007) (pp. 258–261). Ieee. doi:10.1109/RE.2007.25
31. Potts, C. (1999). ScenIC: a strategy for inquiry-driven requirements determination. Proceedings IEEE International Symposium on Requirements Engineering (pp. 58–65). IEEE Comput. Soc. doi:10.1109/ISRE.1999.777985
32. Potts C., Takahashi K., Anton A. (1994). Inquiry based requirements analysis. IEEE Software, March 1994, 21-32.
33. Rago, A., Marcos, C., Diaz-Pace, J. A. (2014). Assisting requirements analysts to find latent concerns with REAssistant. Automated Software Engineering, 1-34.
34. Segrestin, B. (2009). Collaborative innovation capabilities: developing platforms through design games, International Product Development Conference (Enschede).
35. Stevens, R. D., Robinson, A. J., Goble, C. A. (2003). myGrid: personalised bioinformatics on the information grid. Bioinformatics, 19(Suppl 1), i302–i304. doi:10.1093/bioinformatics/btg1041
36. Sutcliffe, A. G. (2002), User-Centered Requirements Engineering. Springer-Verlag London.
37. Sutcliffe, A. G. (2009). On the Inevitable Intertwining of Requirements and Architecture. (K. Lyytinen, P. Loucopoulos, J. Mylopoulos, B. Robinson, Eds.)Lecture Notes in Business Information Processing, 14(Design Requirements Engineering: A Ten-Year Perspective), 168–185.
38. Sutcliffe A. G., Ryan M. (1998). Experience with SCRAM, a SCenario Requirements Analysis Method, ICRE 1998, 164
39. Swartout, W., Baker, R. (1982). On the Inevitable Intertwining of Specification and Implementation. Communications of the ACM, 25(7), 438–440.
40. Ure J., Rakebrandt F. , Lloyd S., Khanban A., Procter R., Anderson S., Hanley J., Hartswood M., Pagliari C., Mckinstry B., Tarling A., Kidd G., Corscadden P. (2009). Giving Them Something to Hate: Using Prototypes as a Vehicle for Early Engagement in Virtual Organizations Social Science Computer Review November 2009 27: 569-582, first published on June 12, 2009
41. Wolstencroft, K., Haines, R., Fellows, D., Williams, A., Withers, D., Owen, S., Soiland-Reyes, S., et al. (2013). The Taverna workflow suite: designing and executing workflows of Web Services on the

desktop, web or in the cloud. Nucleic acids research, 41(Web Server issue), W557–61. doi:10.1093/nar/gkt328

42. Yeltayeva, K. (2012). Usability Study of the Taverna Scientific Workflow Workbench. Master Thesis. University of Manchester.
43. Zimmerman, A., Nardi, B. A. (2006). Whither or Whether HCI : Requirements Analysis for Multi-Sited Cyber-infrastructures. Conf. Human Factors in Computing Systems, CHI ’06 Extended Abstracts on Human Factors in Computing Systems, (pp. 1601–1606). ACM Press.

8 Appendix A: Further Enhancements

User feedback has been taken into account for the further enhancement of the BioVeL toolkit. Table 4 shows the mapping of the deployed versions of the toolkit along with the review events where they have been presented. The first column indicates the version number and the date of deployment. The second column lists the review events, including location and date. The third column indicates features incorporated in the version being used at each event.

Table 4 Prototype Progress and Dissemination

Version	Review event	Main features
Prototype 0.0.1 05/09/2012	Annual Project Meeting, Aix-en-Provence, France, 05/09/2012 Demonstration for researchers from the Humboldt Institute of Colombia, Cardiff University, 16/10/2012	Added a Homepage Handle User-Sessions Workflow Management (add, delete, etc.) Run Management (add, delete, etc.) Add Server Credentials
Beta 0.3.1 04/11/2012	EC Review, Brussels, 08/11/2012 Online Demonstration for Centre for Ecology & Hydrology, 14/11/2012	Deployed on Amazon cloud server Get details from workflow file Connectivity to R-Server Submit files as inputs Improved formatting of results
Beta 0.3.2 15/11/2012	Service team meeting demonstrations during November – December 2012 and January 2013	Improved run management Enabled workflow interaction Link to myExperiment Added announcements
Beta 0.3.3 28/11/2012	Play Day 2 – Testing BioVeL Portal DRW and ENM 30/01/2013 Play Day 3 – Testing BioVeL Portal DRW and ENM 31/01/2013	Improved workflow upload Improved workspace separation Enabled cancelling runs Added result descriptions
Beta 0.3.4 31/01/2013	Play Day 4 – Testing BioVeL Portal DRW and POPMOD 26/02/2013 Population Modelling Workshop 07/03/2013 to 08/03/2013	Enabled anonymous running Improved look and feel
Beta 0.3.5 05/03/2013	Play Day 5 – ENM Training at Gothenburg, 15/03/2013	Connection to myExperiment Improve user management Improve presentation of tables and tabs Improve results formatting Enabled workflow download
Beta 0.3.6 07/03/2013	Presentation on the Biodiversity Informatics Group Seminar, Cardiff, 28/03/2013 Play Day 12 – Browser and Stress Test of BioVeL Portal, 13/05/2013 to 17/05/2013 Play Day 13 – Functional Test BioVeL Portal, 21/05/2013	Annotation of code
Beta 0.3.7 30/05/2013	Play Day 14 – Biome-BGC Test on BioVeL Portal, 31/05/2013 Biome-BGC workshop, Budapest, 06/06/2013 to 07/06/2013	Customisation of input examples Customisation of output examples Customisation of error messages

The more advanced versions of the prototype were implemented, deployed and demonstrated to users, but they were not released for further testing and validation. Instead of this, the functionalities will be implemented into the final production version of the portal when and if required.

8.1 Review of Prototype Versions

The prototype has been used and evaluated during different review events, which can be grouped according to the scenarios described in section 3.4.1.

In the demonstration scenario context, the portal has been used to demonstrate the advances of the project and its different products on two occasions. The first demonstration occurred during the presentation of advances to the member of BioVeL during the second annual project meeting. The second demonstration was performed during a formal EC review of project progress.

In the outreach scenario context, the portal has been demonstrated to scientists from different British, European and American Institutions to showcase BioVeL products and to promote and support the adoption of virtual research environments in the Biodiversity research community.

In the workflow development scenario, the prototype has gradually become the medium for testing advances on workflow development in scheduled "play-days". The play-days are online working sessions in which a workflow is tested by different users and developers. Previously, these involved the use of the Taverna workbench but now they are mostly carried out on the Portal. This prompted the further strengthening of the toolkit features for supporting different users and managing run results.

In the teaching scenario, the portal has been used in training sessions and workshops. In addition to this the prototype has also been used in Workshops for population modelling, niche modelling and ecosystems functioning.

In addition to these scenarios, the portal has been increasingly used by BioVeL partners for their day to day use, supporting the production of research papers and reports for environmental agencies.

8.2 Requirements Review

The implementation of the features addressing the requirements gathered during different stages of the SCRAM-CK process has been interleaved with the activities for validation and refinement of those same requirements. This has facilitated providing a useful test platform that has allowed further enhancement of the toolkit thanks to actual feedback from workflow developers and users. Table 5 shows the mapping of the requirements to the deployed and planned versions of the toolkit. The final columns indicate the components of the BioVeL Workflow Web Toolkit that were modified for the implementation of each feature.

Table 5 Requirements Implementation Roadmap

Features	Version	Status	BioVeL Portal	Taverna Player	Taverna Lite
Simplified workflow execution	Prototype 0.0.1	Released		X	
Presentation of results	Prototype 0.0.1	Released		X	
Downloading Results	Prototype 0.0.1	Released	X		
User management	Prototype 0.0.1	Released	X		
Workflow execution feedback	Beta 0.3.2	Released		X	
Workflow execution statistics	Beta 0.3.2	Released	X	X	
Implement connectivity with myExperiment	Beta 0.3.5	Released	X		
Allow downloading workflows	Beta 0.3.6	Released	X		
Error Handling	Beta 0.3.7	Released	X	X	X
Allow uploading sample input and outputs	Beta 0.3.7	Released		X	X
Customise inputs	Beta 0.4.2	Deployed			X
Customise outputs	Beta 0.4.3	Deployed			X
Component View of Work Flow Structure	Beta 0.5.1	Deployed			X
Select Intermediate outputs	Beta 0.5.2	Deployed			X
Implement Component Swapping	Beta 0.5.3	Deployed			X
Workflow Template View	Beta 0.6.1	Deployed			X
Template based customisation	Beta 0.6.2	Deployed			X
Save workflows to myExperiment	Beta 0.6.3	Pending	X		X

Figure 1 The SCRAM method

Click here to download high resolution image

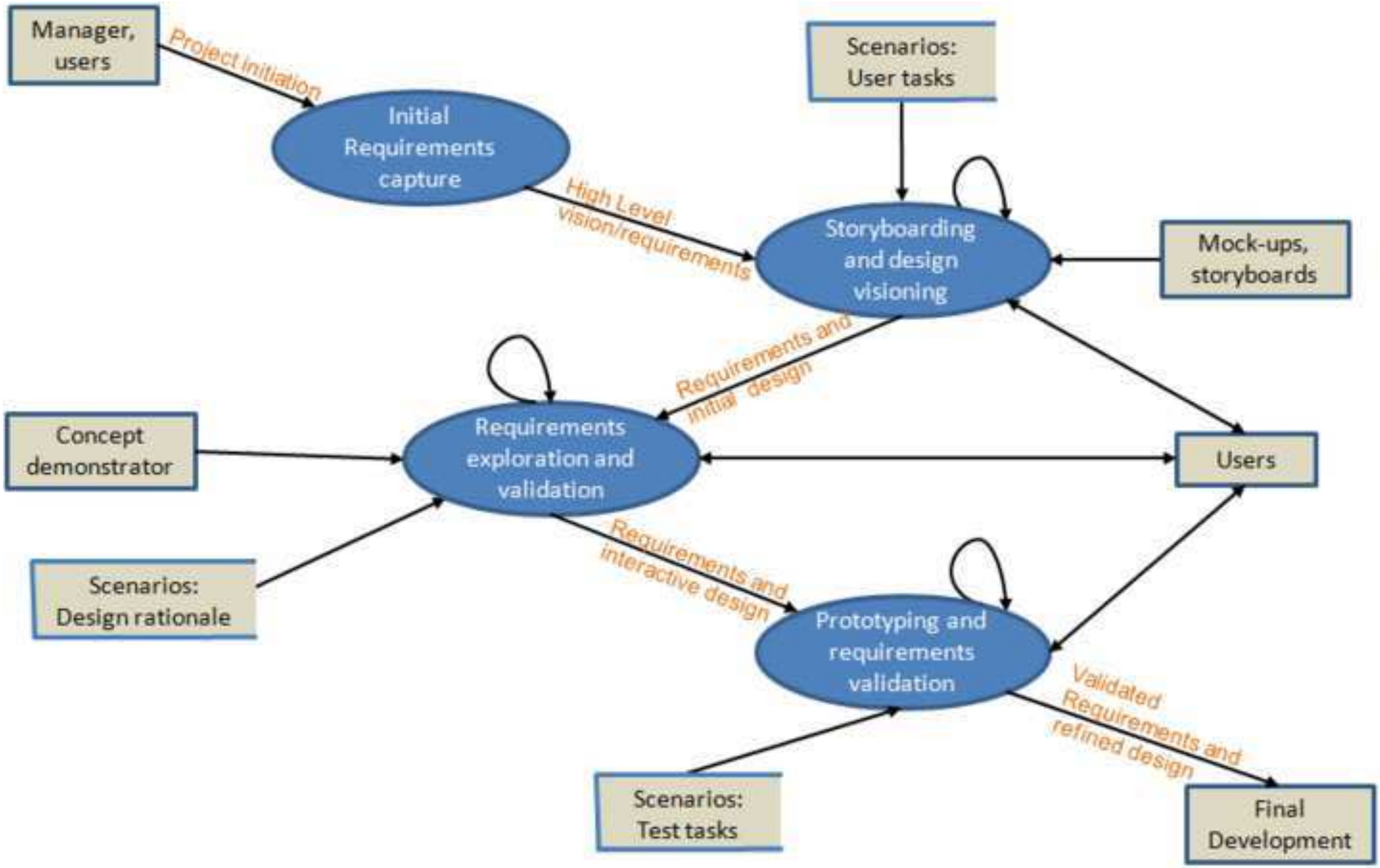

Figure 2 Dynamics of the design process according to CK Theory
Click here to download high resolution image

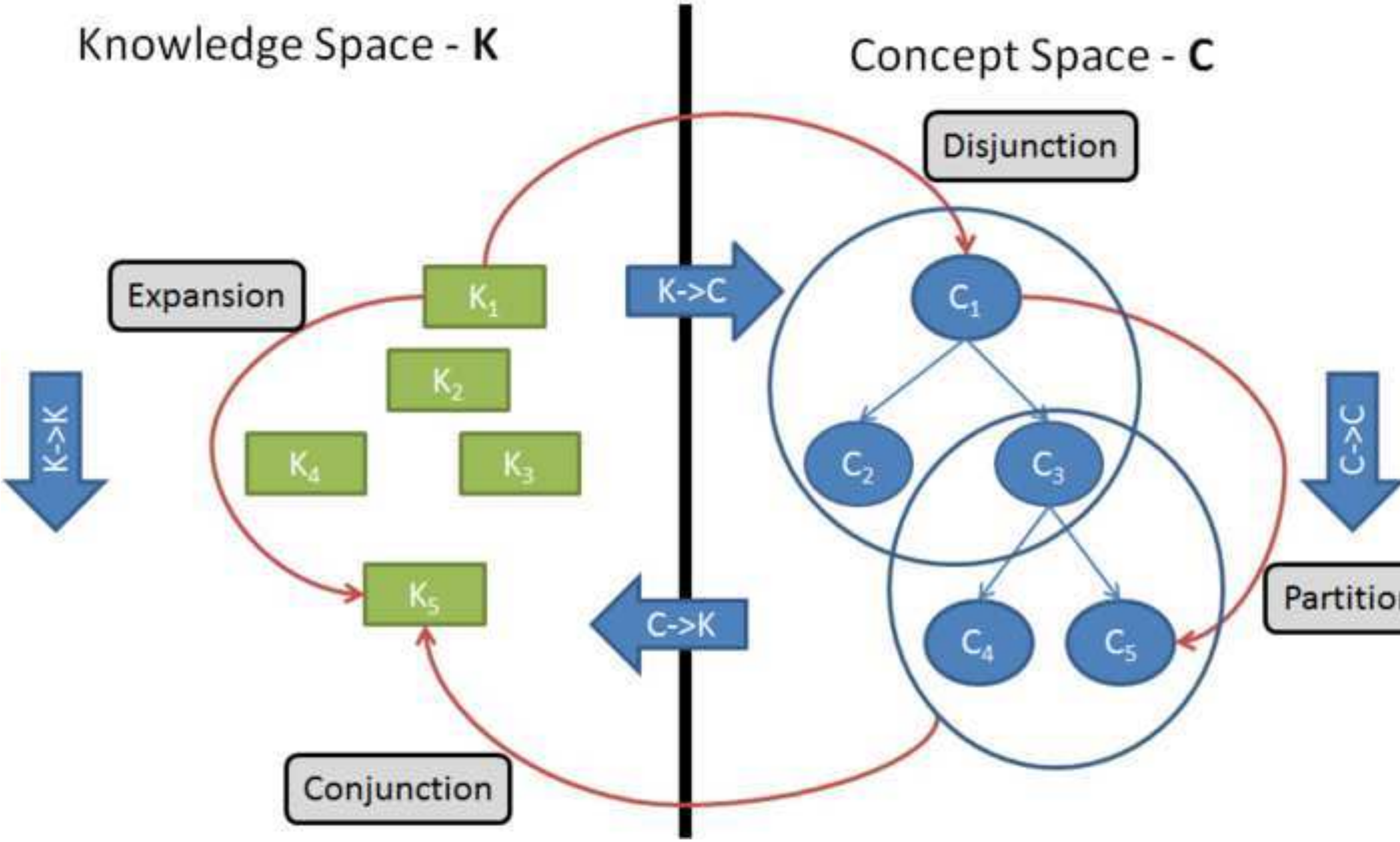

Figure 3 High Level Diagram of BioVeL Web Toolkit

Click here to download high resolution image

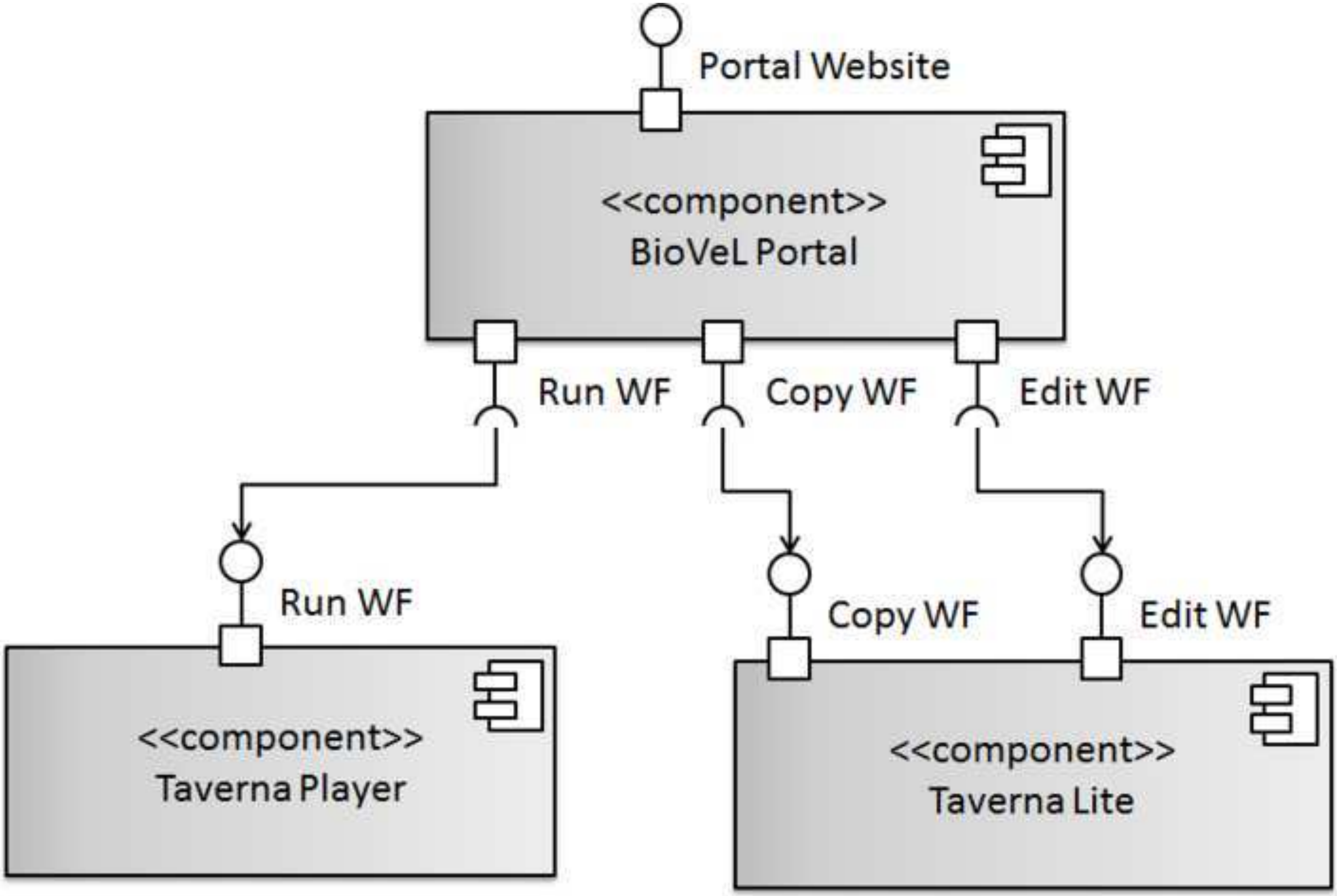

Figure 4 Stage 1 Domain Familiarisation.

Click here to download high resolution image

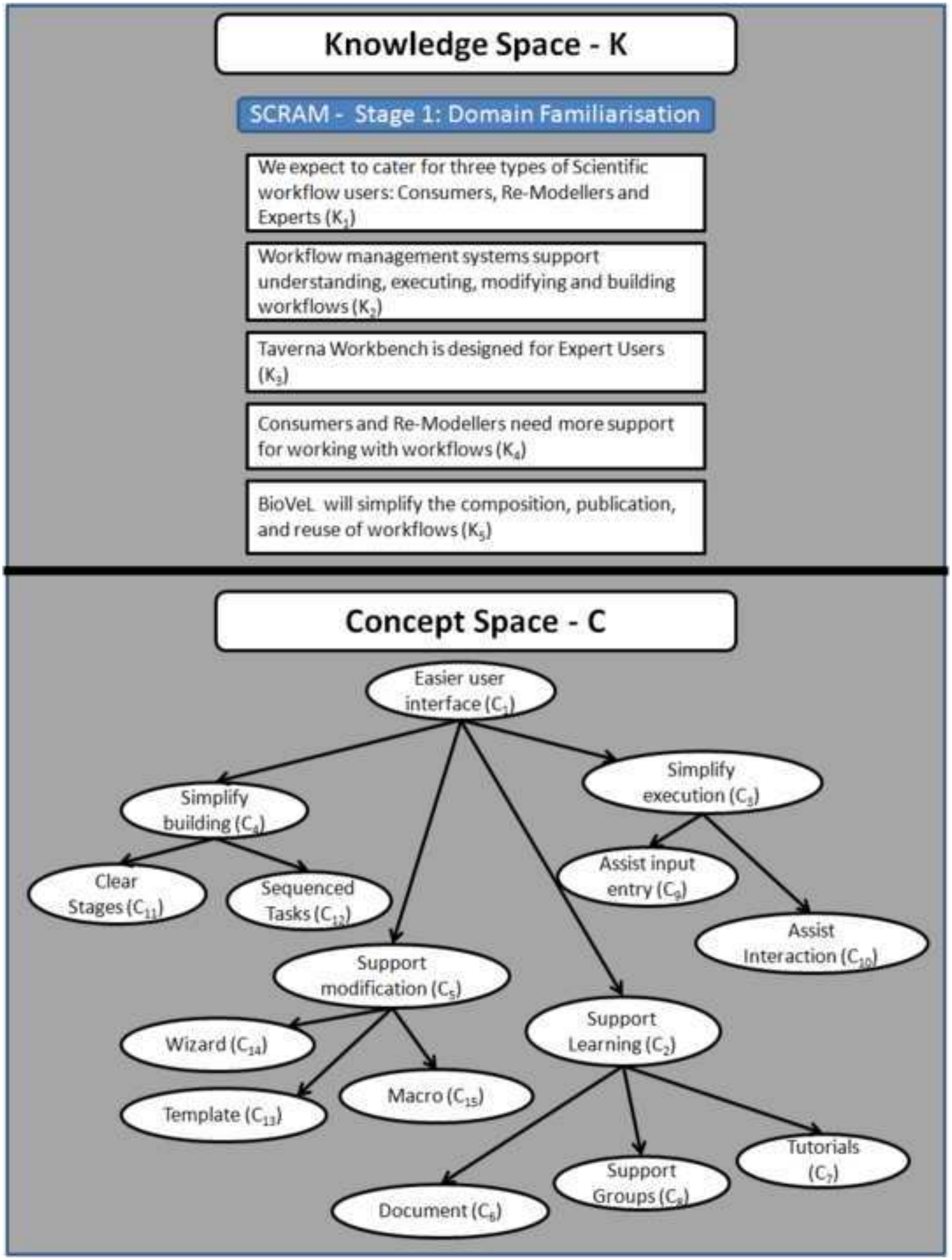

Figure 5 Stage 1 Domain Familiarisation and User Observation

Click here to download high resolution image

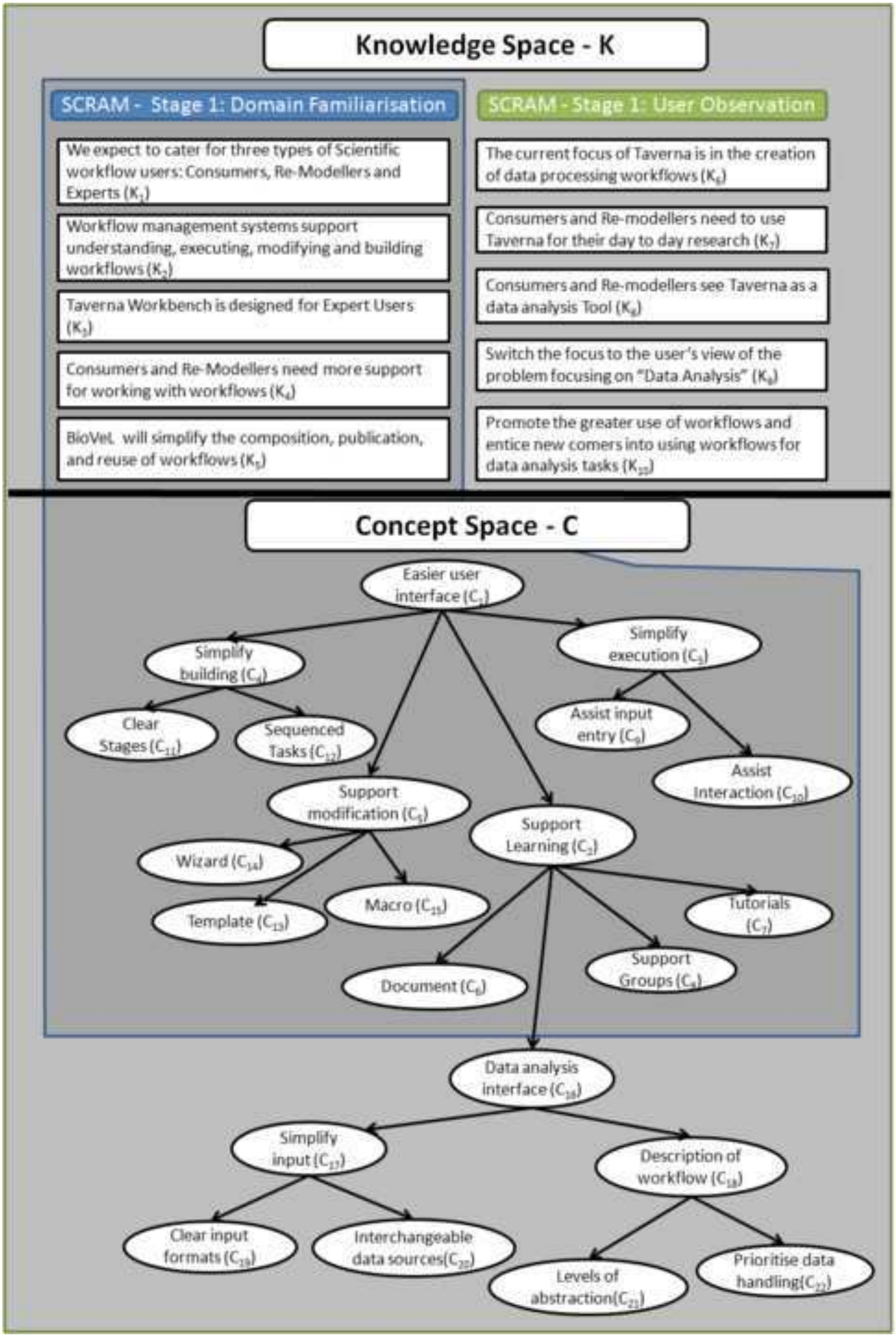

Figure 6 Stage 1 and Stage 2

Click here to download high resolution image

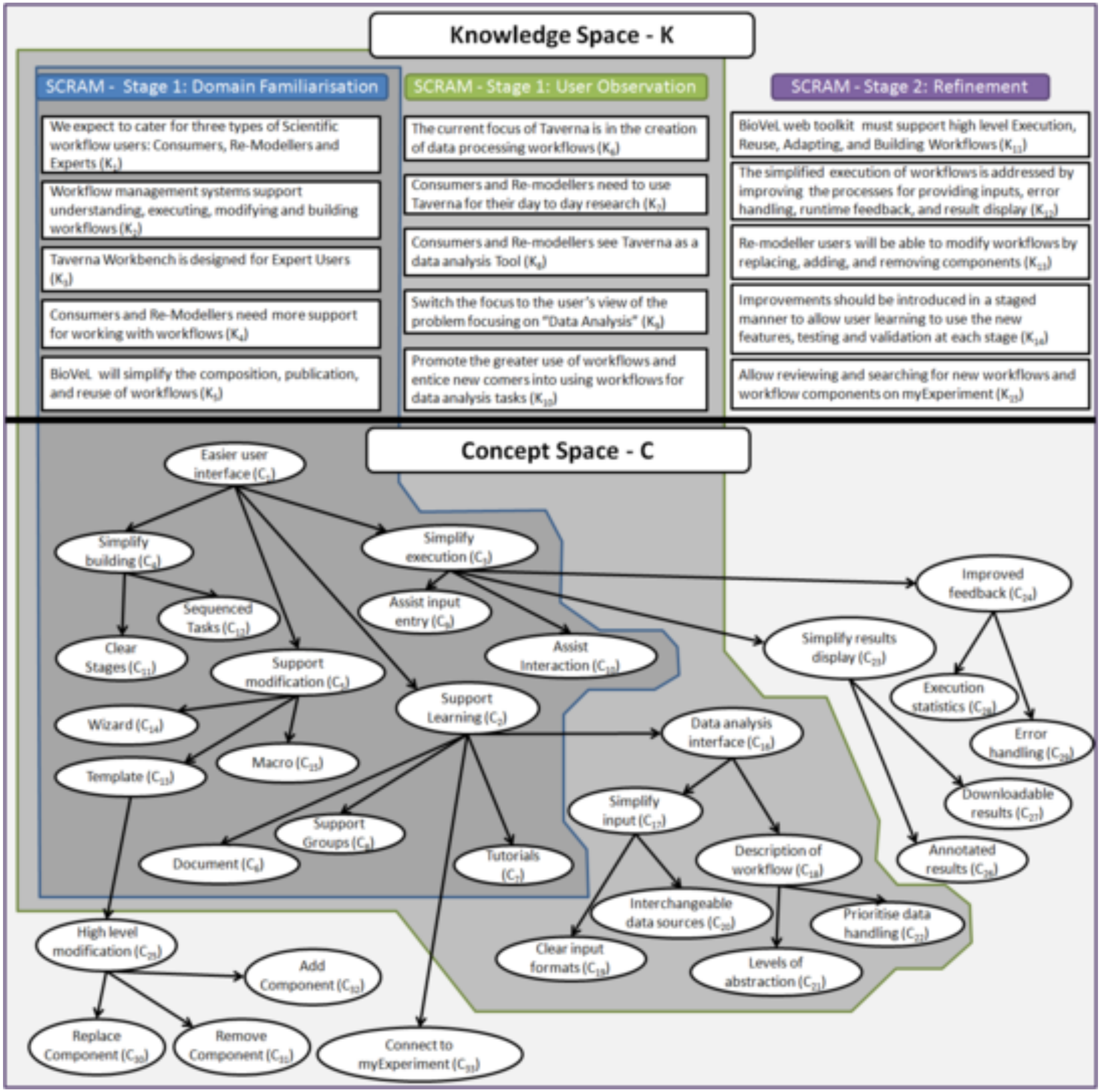

Figure 7 BioVeL Portal Prototype 0.0.1
Click here to download high resolution image

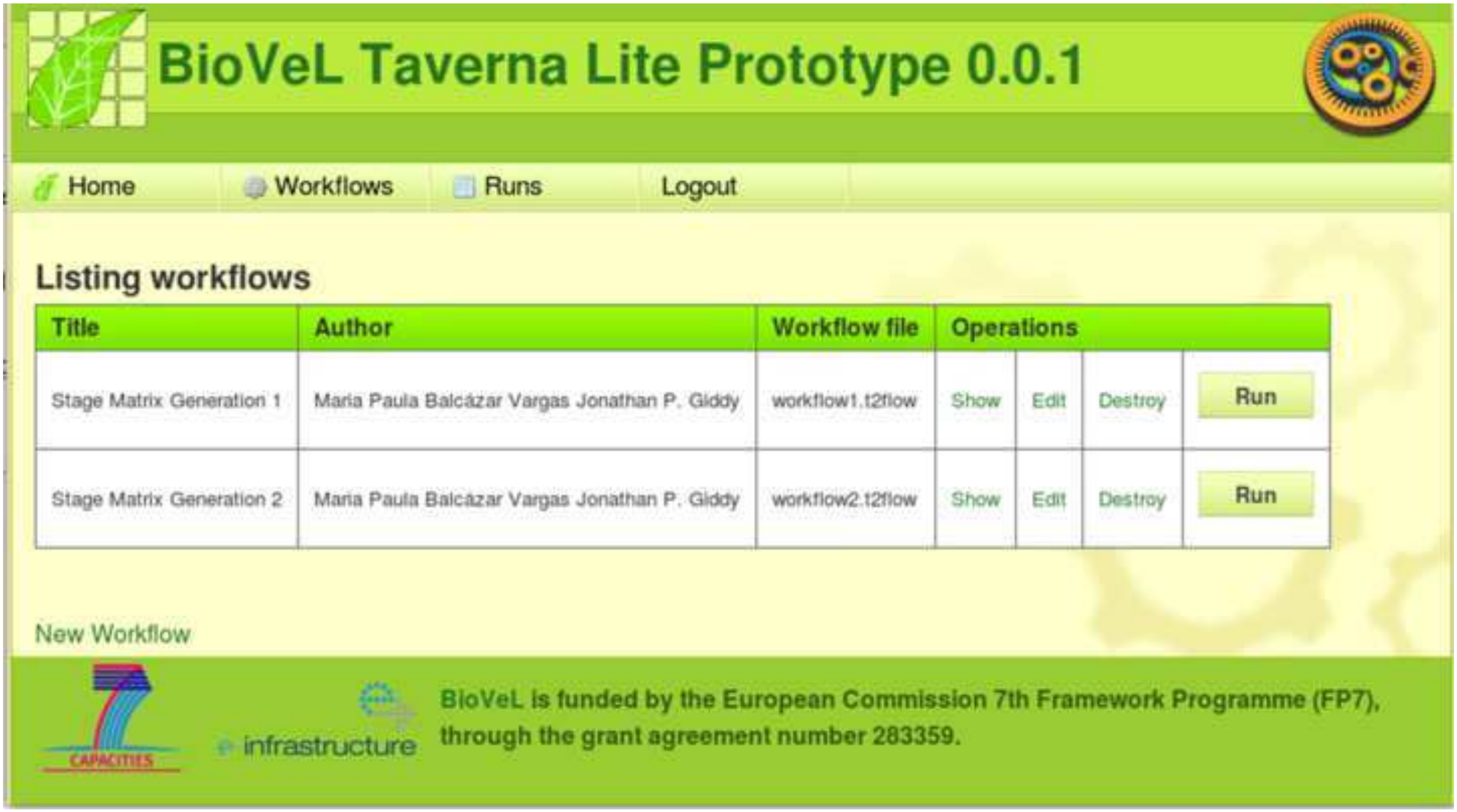

Figure 8 Stage 3 Exploration.

Click here to download high resolution image