USABILITY STUDY OF THE TAVERNA SCIENTIFIC WORKFLOW WORKBENCH

A dissertation submitted to the University of Manchester

for the degree of Master of Science

in the Faculty of Engineering and Physical Sciences

2012

Kymbat Yeltayeva

School of Computer Science

Table of Contents

Word count: 22697

Table of Figures

List of Tables

ABSTRACT

The Taverna Workbench provides functionality which allows the handling of large amounts of experimentation data, linking together various tools and services into a single research analysis and dealing with incompatible data formats. This project aims to understand the usability of Taverna so the user experience of the tool could be reviewed and improved.

The study examined and identified usability issues by observing two recruited groups of users of the Workbench: programmers and computational scientists. The main technique for collecting data was Remote Usability Testing used together with the Think-aloud protocol and Users Diaries. Obtained information was coded for further analysis using the open-coding technique of the Qualitative research and categories were formed within the Grounded Theory methodology.

The obtained results revealed a number of categories of the Taverna Workbench that warranted improvement, which were concentrated around Propagation, Visual Representation, and Sub workflow/Workflow piecing issues. Based on the findings, a list of suggestions to the Taverna development team was produced.

Study results suggested prioritisation using the MoSCoW prioritisation method such that Taverna developers have a map to the most important changes. Study findings showed that although most users find the user experience of the workbench generally satisfying they face difficulties in specific areas when interacting with the Taverna Workbench.

DECLARATION

No portion of the work referred to in the dissertation has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

INTELECTUAL PROPERTY STATEMENT

i. The author of this dissertation (including any appendices and/or schedules to this dissertation) owns certain copyright or related rights in it (the "Copyright") and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this dissertation, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has entered into. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the dissertation, for example graphs and tables ("Reproductions"), which may be described in this dissertation, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this dissertation, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://documents.manchester.ac.uk/display.aspx?DocID=487), in any relevant Dissertation restriction declarations deposited in the University Library, The University Library's regulations (see http://www.manchester.ac.uk/library/aboutus/regulations) and in The University's Guidance for the Presentation of Dissertations.

ACKNOWLEDGMENETS

First of all, I would like to gratefully and sincerely thank my supervisors, Prof. Carole Goble and Dr. Simon Harper for their great support and guidance throughout the project. The completion of the dissertation would not be possible without their wise advice and invaluable help.

I am deeply thankful to all the participants who contributed to the study, for their time and efforts dedicated to participation. I greatly appreciate their input to this work.

My special thanks go to the myGrid team for sharing their knowledge and for their friendliness and readiness to help.

Finally, I gratefully acknowledge the Bolashak International Scholarship and the Government of the Republic of Kazakhstan for providing the opportunity to study in the University of Manchester and giving the financial support.

CHAPTER 1. INTRODUCTION

A Scientific Workflow can be defined as a means for managing and sharing complex scientific analyses which is constructed by chaining together different services or codes [1]. Taverna is an open source Workflow Management System developed by the myGrid team which enables setting up, executing and monitoring scientific workflows.

More than 350 organizations around the world use Taverna for executing workflows and sharing them with others. The Taverna Scientific Workflow Workbench is widely used by scientists from different domains, such as Astronomy, Bioinformatics, Chemistry, data and text mining, Engineering, etc.. This project is to review and improve the User Experience of the Taverna workbench by running a systematic usability study of the tool.

User Experience is the field which studies the user's attitude to the particular (software) product and how users perceive various aspects of the tool such as the ease of use and efficiency. After investigating existing techniques User Testing was chosen as the main method in studying usability of the Taverna workbench. As Jakob Nielsen states [2]:

> "User testing with real users is the most fundamental usability method and is in some sense irreplaceable, since it provides direct information about how people use computers and what their exact problems are with the concrete interface being tested "

As opposed to the other popular methods of studying usability - questionnaires and focus groups - user testing involves actual observation of the users. The former implies listening to what people say, while in the latter case a researcher has an opportunity to directly observe the interaction and draw conclusions.

The users were recruited and continuously observed while they were working with the tool. The study participants are computational scientists from various disciplines with the difference in their Taverna experience. Two main groups of users were presented: programmers and computational scientists with 6-7 users in each group for qualitative

study. In qualitative studies the data is usually gathered by directly observing how people use technology to meet their needs. It helps understanding what people feel when they work with the system, as well as human behavior and the motives for that behavior. In these studies, a smaller number of participants is required in comparison to quantitative experiments [3]. A discussion of user study size in this project will be introduced later in this thesis.

1.1 Motivation

Taverna is a sophisticated system and scientific workflow construction is usually a complex and computationally intensive process. The Taverna workbench allows the accessing of multiple, distributed analysis tools and remote third-party services [4]. As a result of its broad functionality, the tool can be complicated and difficult to use. The Figure 1 shows an example of typical Taverna workflow [5]. An assessment and enhancement of the usability of the Taverna workbench was decided necessary.

Another aspect is the difference in research disciplines and programming background of the Taverna users. It is important that Taverna is intuitive and approachable for people with different programming experience from any domain. The usability study of the tool is also important for identifying the overall acceptance of the product.

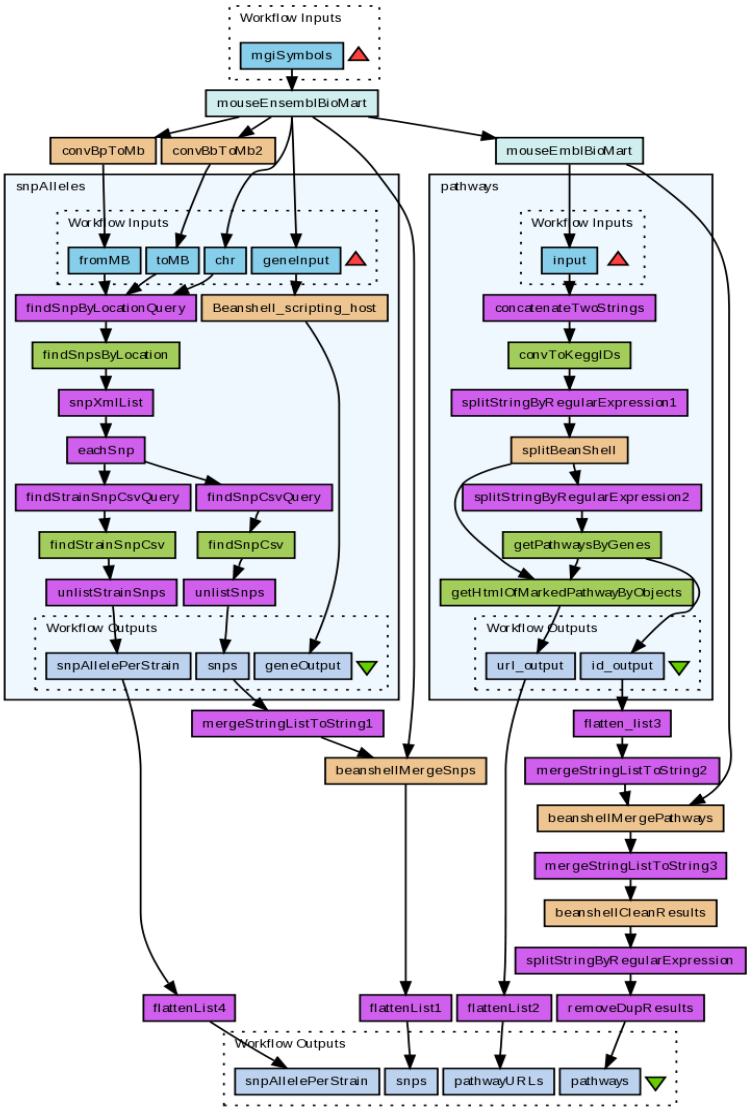


Figure 1. Example of a Taverna Scientific Workflow for mouse functional genomics from CASIMIR

Adapted from [5]

1.2 Aims and Objectives

The main aim of the Usability Study of the Taverna Scientific Workflow Workbench project can be stated as follows: to understand and measure the user experience of the Taverna Scientific Workflow Workbench by conducting a systematic usability study of the tool.

For reaching this aim the following objectives must be met:

- Develop the methodology for the study;
- Conduct the experiment;

- Produce the usability design;
- Report the observations to the Taverna team;
- Make recommendations to the development team;

In order to achieve formulated aims and objectives the following work on the dissertation has been done, which is also reflected in Figure 2 below. The work consisted of the six main stages:

1. Preliminary group: First, usability evaluation methods, user experience measurement tools and techniques were investigated for building the background for further work.
2. Methodology development: Then, a preliminary methodology for measuring the user experience of the tool was designed based on the findings from the previous stage. This methodology was used in a pilot experiment.
3. Pilot experiment: The pilot experiment was conducted in order to observe the effectiveness of suggested techniques within the methodology. The results were examined, and then the methodology was enhanced and modified based on the pilot study outcomes. Updated methodology was used in the main experiment of the study.
4. Participants' recruitment: The participants for the usability experiment were recruited and contacted, the environment was set up.
5. Main experiment: The main experiment was performed applying the developed methodology.
6. Analysis and Presentation: Open coding technique [6] was employed for developing initial concepts. Card sorting was used in order to allow categories to emerge. After processing the results, they were reported to the Taverna software development team. Based on the obtained information, suggestions were made to the Taverna development team.

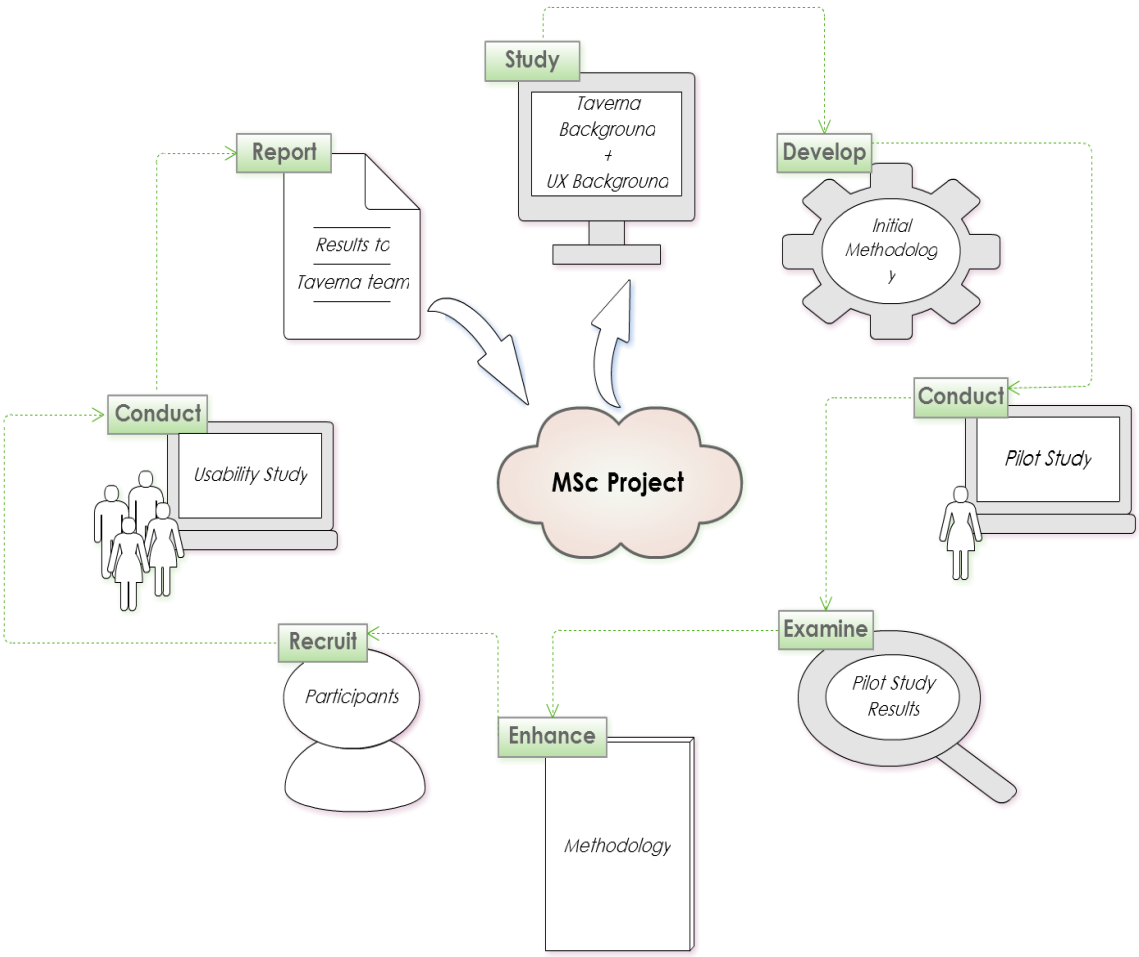


Figure 2. Overall process of work on the project

1.3 Scope and limitations

Having specified the aims and objectives of the project, the scope of the dissertation can be defined. First, the study relied on the Grounded Theory methodology where no hypotheses were suggested in advance. The theory was formed through the analysis of data obtained during the experiment. The details of the Grounded Theory are given later in the next chapter. The study was not a predefined, laboratory experiment but rather a field investigation, where the experiment process was uncontrolled. Users performed real tasks in a natural environment with usual settings. Next, the undertaken project employed remote techniques as most of the Taverna users are located in distant places. Finally, the length of the usability experiment was limited to 2 months period, during which video recordings were obtained every week from each user.

1.4 Thesis structure

The structure of the dissertation is the following:

Chapter 2 – Project Background: The chapter provides background information of the Taverna Scientific Workflows Workbench and User Experience. It discusses scientific workflows in general and proceeds to the detailed description of the Taverna Scientific Workflow Workbench. The notion of a scientific workflow system is discussed and an overview of the current scientific workflow management systems is provided. The relevant material on the User Experience background is also presented including methods description and comparison. The information given in this chapter is essential for further understanding of the project design and solutions.

Chapter 3 – Pilot experiment: The pilot study is described in this chapter. First, objectives of the pilot experiment are clarified and the environment settings are reported. The chapter gives the information regarding the initial methodology for the pilot study and illustrates its process. Finally, the results and improvements applied to the methodology are presented.

Chapter 4 – Experimental Work: This is a central chapter which discusses the usability experiment and it is organised according to the American Psychological Association (APA) experiment format. Following the format structure, it starts with the Participant recruitment process including limitations and user profile information. Next, it presents Materials employed in the study, such as software used in data collection and analysis. Steps taken to complete the experiment are described in the Procedure section. This section also covers initial considerations, task analysis and methodology description.

Chapter 5 – Evaluation and Results: This chapter discusses obtained results of the study.

Categories and codes are offered first, followed by their severity ratings discussion. Local and global findings are also presented. The chapter ends by presenting a “Word Cloud” and the Presentation to the Taverna team. The chapter analyses and demonstrates all the study outcomes.

Chapter 6 – Conclusion: The final chapter reflects on the developed methodology, methods employed and environment. The discussion on project achievements and possible future development and improvements is provided. The chapter is concluded by presenting obstacles overcome and identified risks.

CHAPTER 2. PROJECT BACKGROUND AND LITERATURE REVIEW

This chapter discusses the relevant background material to the project with the purpose of covering the environment where this project is situated, defining the specific terms and providing the reader with necessary information for further understanding. As the current project seeks to resolve the problem of measuring and improving the user experience of the Taverna Scientific Workflow Workbench, the information related to the Scientific Workflows Management Systems as well as the User Experience field will be given. The chapter also aims to justify the need for the project.

2.1 Scientific Workflows

Workflow as a notion emerged about three decades ago and it was defined in 1996 by the Workflow Management Coalition as an automated process where data is passed for further actions from step to step. The emphasis is made on the process, as a flow of action, from one phase to another, chaining required services for achieving a desired result [1]. At the beginning workflows were used in a business context, but later they found their application in science as well. Mainly this is due to the spread of *in silico* experimentations which make use of computers/computer simulations. Workflows which are used in these experimentations are called Scientific Workflows [7].

2.1.1 Scientific Workflows Overview

Scientific Workflows can be defined as a useful paradigm for describing, managing, and sharing complex scientific analyses [8]. Scientific and business workflows have similarities in terms of possibility to apply control flow modeling techniques used in Business Workflow Management Systems to Scientific Workflow Management Systems [9]. However, workflows in a scientific environment go beyond the initial notion of workflows in a business perspective. Scientific Workflows support not only the

management and transactions between resources within one domain, but also enable the automation of the data analysis through heterogeneous data resources [4].

There are several motivations for Scientific Workflows [9]:

- To build a collaborative workflow for complex e-science applications;
- To carry out a low-level expertise for using the underlying computing infrastructure such as Grid toolkits;
- To reuse, modify and share the analysis;

Scientific workflow is a composition of different remote local services in a linked components manner in order to produce results for further analysis. Each component performs a particular task which is a fragment of the overall work, that the workflow is composed to accomplish. The output of the previous component should fit to the input requirements imposed by the next node of the workflow. Often there might be the case of data formats incompatibility, when the input type of one workflow node is different from the output format of previous component which is going to be fed. Tasks within the workflow are different steps which present a particular computational process. Examples can be: executing a program, querying a database or invoking a service to use a remote resource. The output from one stage serves as an input to the next creating the flow of data [10]. This process of chaining workflow components is called *workflow composition*. The result is a graph-like structure which is illustrated in Figure 3.

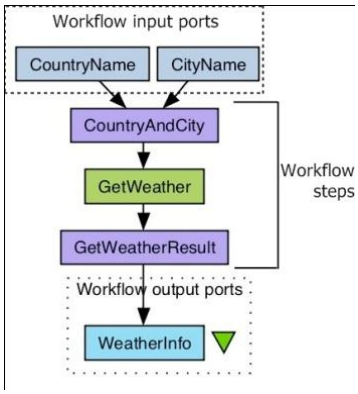


Figure 3. Example of a simple Taverna workflow
Adapted from [11]

Scientific Workflows help scientists by offering an abstract view, concealing at least some of the complexities and details of how the experiment process will be executed. Instead, Scientific Workflows allow a clear view of what the task is aiming to achieve. Scientific Workflows make available sufficient computational resources for researchers and allow access to necessary services and data. Scientists also have an opportunity to share and reuse workflows in a simple way. In addition, they can track the process of the workflow creation and execution. Scientific Workflows acquire more importance as science is becoming more computation-intensive. It is also difficult for researchers to handle the growing complexity of the experiments and Scientific Workflows come to help [12].

2.1.2 Scientific Workflows Management Systems Background

A Scientific Workflow Management System (Swfms) is a software package which enables the setting up and executing of scientific workflows by providing an environment for running of *in silico* experiments [13]. In most of these systems workflows are constructed and modified using a graphical interface. They are used by scientists for the assembly and management of complex distributed computations. Figure 1 presents an example of the Taverna Scientific Workflow performing such computation.

There are two main workflow system classes: data driven and control-driven. Data-driven workflow systems are concerned mainly with data itself, which transforms from stage to stage constituting the entire process. In contrast, a control-driven workflow system focuses on processes management and transfers control from component to component [14].

Workflows Systems support the graphical designing of the workflows. The user indicates the subsequent steps in the workflow and the system performs particular tasks

within those steps, such as getting the required data from a database, calling different web services or other software applications, and allocating tasks on a grid [14].

Scientific Workflow Management Systems try to [15]:

- Deal with the complexity of data analysis in a scientific domain;
- Provide an easy-to-use way of conducting in silico experiment;
- Hide at least some of the technical details of workflow execution allowing scientist to concentrate on the data analysis;
- Provide a graphical user interface so that users could compose web services into workflows;
- Enable scientists reusing and sharing workflows between them for example through web sites, such as myExperiment[16].MyExperiment is an environment for publishing and sharing Scientific Workflows and in silico experiments [17];
- Help to deal with data incompatibility;

The increasing popularity of Scientific Workflow Management Systems can be accounted for by the growing number of scientists relying on these systems for conducting complex, distributed computations.

2.1.3 Current Scientific Workflows Management Systems

There are various Scientific Workflow Management Systems based on dataflow languages, which provide a graphical interface for users for constructing applications as a visual directed graph by linking the components together. Amongst the most widely used examples of the current Scientific Workflow Management Systems are Taverna [1], Kepler [18], VisTrails [19], Triana [20] Pipeline Pilot [21] and KNIME [22].

Figure 4 gives the main features and characteristics of each of the abovementioned systems. We then give their more detailed description (Taverna's comprehensive description is given later).

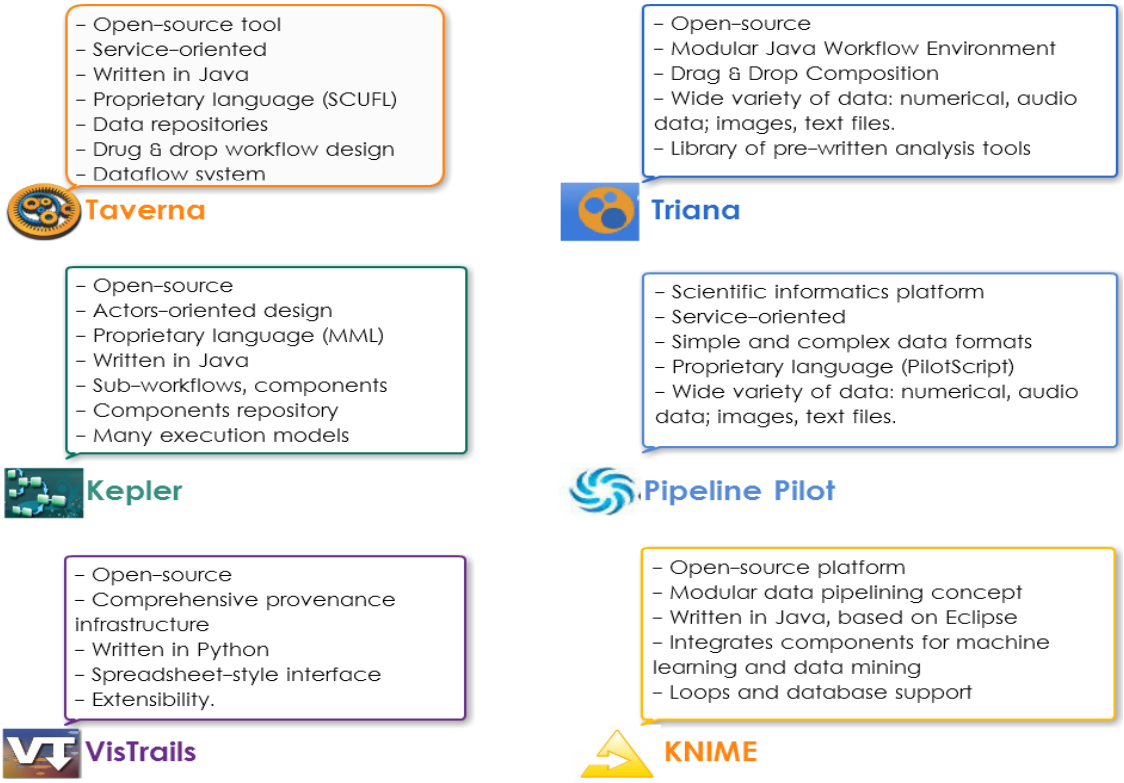


Figure 4. Main features of some of the Scientific Workflow Management Systems

Kepler [18] is an open-source Scientific Workflow System. Kepler includes a graphical user interface for building workflows in a desktop environment and a runtime engine for executing workflows separately from a command-line within the graphical user interface. A distributed computing option provides the ability to distribute workflow tasks between several of components in a computer cluster. Kepler makes an emphasis on actor-oriented design where actors are re-usable computational units, such as web services. Data is fed to the actors from inports and it is written to outports. Then actors can be combined by mapping from outports to inports [23]. Other features of Kepler are: workflows and components can be saved, reused, and shared with other researches with the means of the Kepler archive format (KAR). Kepler allows nested workflows. The software also includes a library with around 350 prepared for use processing elements, which can be searched, modified and linked in an easy way. They also can be executed from a desktop for carrying out an analysis, automating data management, and integrating applications efficiently [18, 24, 25, 26].

VisTrails [19] is a scientific workflow and provenance management system which delivers data exploration and visualization services. VisTrails is an open-source software package which main feature is a comprehensive provenance infrastructure with history information about the steps taken and data obtained during running an exploratory task. This information is given either as XML files or in a database so users can intuitively operate between workflow versions, to undo actions without losing results, match workflows and their results, and analyse the actions which produced a result. In VisTrails sequence operations and user interfaces are presented which make the design and management of workflow easier, providing the ability to create, enhance and query workflows by example [19].

Triana [20] is an open-source simulation system and problem-solving environment developed at Cardiff University. It is used by researchers for a variety of tasks, such as simulation, signal, text and image processing. Triana offers an intuitive visual interface along with data analysis tools for creating, modifying, managing and running workflows. Triana enables users to build workflows by dragging units or tools onto a working area and joining them together by connecting components using data and control links. Triana has a big library of pre-defined tools for data analysis and users can also easily add their own tools. Various workflow readers/writers can be integrated, for example, Web Services Flow Language (WSFL), Directed Acyclic Graph (DAG), Business Process Execution Language (BPEL), etc. [24]. Triana serves as a powerful toolkit for automating repetitive tasks, such as find-and-replace on all the text files in a specific directory, or continuously observing the data coming from long-lasting experiments. [20, 23, 24, 25].

Pipeline Pilot [21] is a commercial data pipelining framework and a platform which is used for integrating, accessing, handling and analysing large amounts of scientific data in domains such as chemistry, cheminformatics, bioinformatics, etc.. The tool provides an environment for managing service-oriented workflows throughout its life cycle. In order to

create service-oriented workflows two components are used: a custom manipulator component and a set of SOAP components [27]. Within the custom manipulator component the PilotScript language (a functional expression language) is used for specifying the operations performed on the service's input and output. In the SOAP component the Web service can be defined by indicating the path in the WSDL file. In the Pipeline Pilot command line, Web browser, or application can be used for enacting the workflow. The main benefits of Pipeline Pilot are its extensive library of nodes and the lightweight of the client graphical environment. Another advantage is the reliability of the tool. Lastly, Pipeline Pilot offers great capabilities for supporting service-oriented workflow management. The current version of Pipeline Pilot's client graphical environment works only with Microsoft Windows, imposing restrictions for Linux and Macintosh users [26].

KNIME (Konstanz Information Miner) [22] is an open-source and commercial analytics platform which supports data integration, processing, analysis, and exploration. It allows a data pipeline visual construction and interactive execution. KNIME is created for education, research and collaboration purposes. It supports easy integration of new algorithms and provides methods for managing data. One of the attractive features of KNIME is its built-in modular approach, which records and keeps the process of analyses in the order they were conducted, at the same time providing intermediate results availability. The main features of KNIME are its scalability through sophisticated data handling, simple extensibility and intuitive user interface. In KNIME workflows are presented as graphs with linked nodes, which call direct acyclic graph (DAG). New nodes and connections between them can be added using the WorkflowManager. The status of nodes can also be tracked and a pool of executable nodes can returned on demand [22, 28].

The Scientific Workflows area is a new developing field and the number of scientific workflow systems is growing every year. These systems aim to provide scientists

with necessary functionality for conducting compute and resource-intensive analyses. While these systems have common goals and characteristics, they differ in a set of requirements they impose and different languages and workflow execution engines implementation [25].

2.2 Taverna Scientific Workflow Workbench

Taverna is a Scientific Workflow Management System which is created to support the construction of workflows to perform different analyses and the automation of complex, service-based and data-intensive processes. It allows the employment and integration of the variety of different tools which are offered on the web [29]. Taverna is broadly used in diverse domains such as bioinformatics, arts, chemistry, medical research, astronomy, and the social sciences. Most Taverna users have programming experience as the process of work in Taverna requires at least some. The widest application the Taverna workbench found in the domain of the Life Sciences where it is exploited for experimental investigations.

Taverna Workflow Management System consists of the Taverna Workbench desktop application and the Taverna Server which serves for remote execution of workflows. Both of them are powered by the Taverna Execution Engine. It is also available as a Command Line Tool which allows a quick execution of workflows. The current usability study is conducted on the Taverna Workbench, therefore in the rest of the paper the term “Taverna” refers to the Taverna Workbench which provides the main user interface. Taverna Scientific Workflow Workbench allows for the creating, visualization, editing and running of workflows as a desktop application on a computer. Taverna Workbench has a graphical workflow designer where users can drag and drop workflow components. The main features of Taverna are its free availability, domain independence and a wide range of services offered. The important Taverna features include the ability to

immediately consume arbitrary third party services, the support of collection of provenance and the viewing of intermediate results. It also has a plugin platform including external tools. The set of available services is not limited and new services can be rapidly imported into the Taverna Workbench [30]. Taverna supports finding workflows created by others and share yours through myExperiment [16, 17] website. The workflows discovered through myExperiment can be downloaded, edited and run within the Taverna Workbench.

The Graphical user interface of the Taverna workbench is used for workflows construction, execution and results browsing which are generated from workflow runs. There are three perspectives in the Taverna workbench which serve for accomplishing particular tasks in the different stages of workflow composition [30]:

- The Design Perspective is the main perspective of the workbench which offers a means for workflows building;
- The Result Perspective provides functionality for monitoring workflow runs and viewing intermediate and final workflow results;
- The MyExperiment Perspective is a way to access and query the myExperiment website[16] from within the Taverna Workbench;

All the Taverna menus, toolbars and panels are organised into abovementioned perspectives.

Let us describe the Design Perspective illustrated in Figure 5 - the main working view of Taverna which provides functionality for building workflows. It consists of three main areas: Workflow Explorer, Service Panel and Workflow diagram [30].

- The Workflow Explorer is located at the bottom left of the screen. It offers a hierarchical view of the current workflow units, such as services, workflow inputs and outputs, data connections and coordination links and annotations associated with them.

- Service Panel at the top left provides the functionality for managing the tools for building workflows. These tools are displayed as a hierarchy and they can be searched by regular expression. The user can also add services to the existing list of services offered in Taverna.

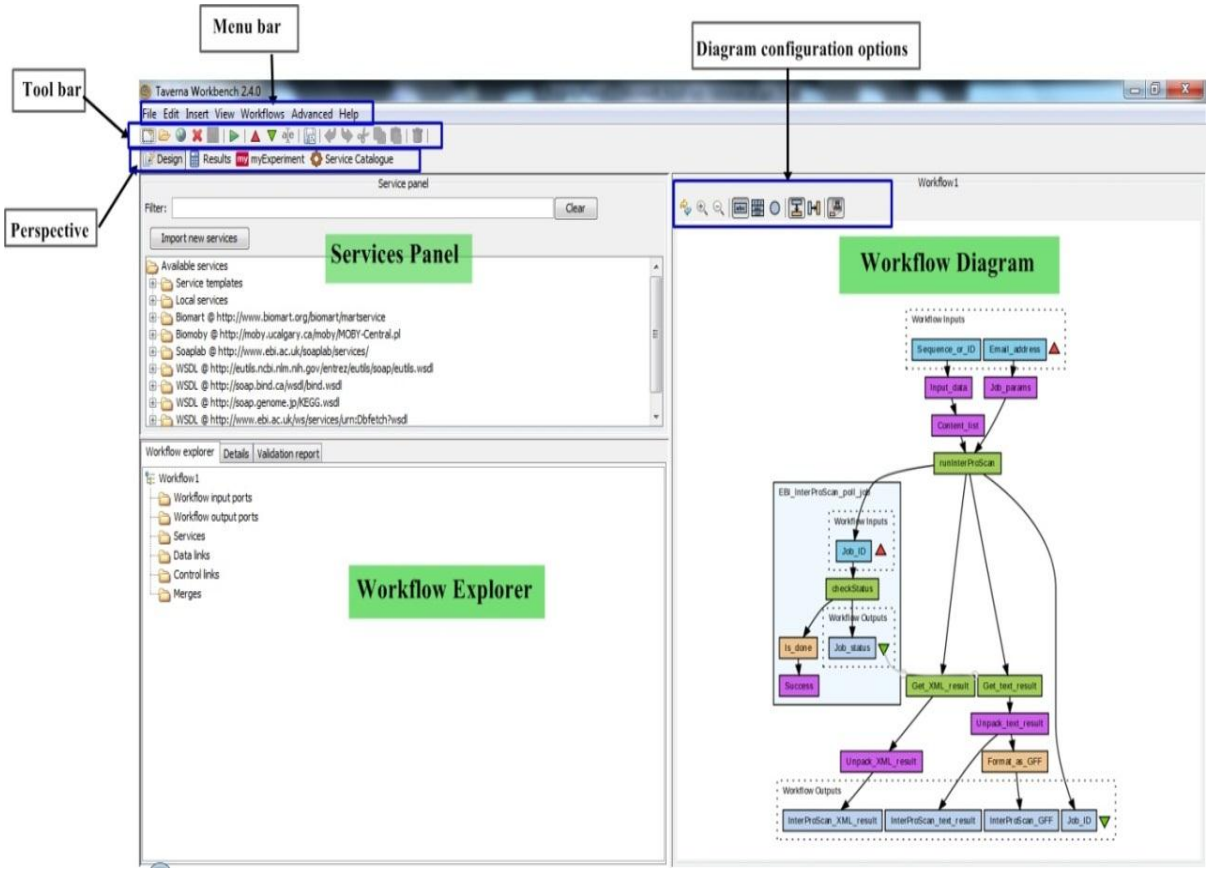


Figure 5. Taverna Workbench - Design Perspective

- The Workflow Diagram, which occupies the right hand-side of the displayed area, provides a graphical view of the current workflow. The diagram can be used to create, edit and modify workflows. Inputs, outputs and processors are presented as boxes of different colours and data and control links are presented as arrows between them.

In order to perform analysis several analytical tools and databases usually need to be used in a sequential order. Connecting the tasks together is typically accomplished either by copy-pasting manually between web pages or by writing a complex scripts. While the first one is simply cumbersome and inconvenient, the second requires good

programming skills. In Taverna the Workflow construction is accomplished through a graphical user interface, by combining different services and into automatic workflows. It seems like a simple and natural procedure to a programmer, but to the scientific end-user "visual programming" methods offered in workflow systems can be unusual and complicated. Particular difficulties can arise when workflow construction passes over into actual programming such as repeating iterations over workflow parts and defining parallel workflows. Another problem might be that the systems require users' knowledge about the necessary workflow components for performing their experiment, as well as the data location which is requested by these components. In addition, the systems also assume that a researcher knows in advance which experiment they are describing [23].

The strengths of the Taverna workbench are its capability to combine a significant range of autonomous services and reproduce scientific analyses and processes [24]. The weakness is that the software can be complicated and difficult to use due to its impressive amount of the functionality.

2.2.1 Process of work in Taverna

The Drag-and-drop interface of the Taverna workbench allows construction of the workflows by chaining services together. The required services are dragged from the Service Panel into the Workflow Diagram. Then these services are connected by indicating ports and drawing arrows between them.

The process of work in Taverna in the full lifecycle of a scientific workflow can be described as follows:

- Determine general workflow intention;
- Discover relevant data and services;
- Build the workflow using available tools and services in Taverna;

- In case of reusing workflow, download a workflow from myExperiment using corresponding tab in Taverna and then apply modifications;
- Execute workflow, invoking used services;
- Collect the results and record the provenance;
- Analyse and share the results using myExperiment.

Taverna offers good tool-suit support in the whole scientific workflows lifecycle and functional programming model that eases data flow modeling [31].

2.2.2 Taverna services.

Taverna allows accessing a great number of web-services in various domains. All services can be accessed from the Service Panel in the Taverna Workbench. Taverna can invoke any Web service with a WSDL (Web Service Description Language) interface, if the URL address of this service is provided. WSDL is an XML format which has the machine-readable description of the functions provided by the service. Other types of Web services offered in Taverna are BioMoby (collection of biological Web services), BioMart (allows querying a BioMart database) and SoapLab (wraps command-line and legacy programs as Web services) services.

Besides, Taverna offers local services, which are also listed in the Service Panel in the Taverna Workbench, such as Beanshell and Rshell scripts. A Beanshell service in Taverna is based on the Beanshell Java scripting language and it enables data manipulation, parsing and formatting. Rshell is a service that allows incorporating the R statistical package into Taverna workflows.

2.2.3 Taverna Users

The Taverna user audience is broad and as study results suggested, mostly Taverna is used by computational scientists or programmers rather than by people with no

programming experience. Taverna is an expert's tool and requires some prior programming knowledge for using it.

Taverna users are scientists from different domains, from Biology to Astrophysics, who use Taverna for supporting their scientific experiments. The challenge is to make Taverna adjustable for specific domains, so that unnecessary functionality for other fields does not disturb and confuse users, on the other hand providing all the necessary facilities for each particular discipline.

Taverna users are originated from different countries. They rarely have connections to each other so they do not have an opportunity to contact and communicate any problems or uncertainties.

Users employ Taverna in average several times per week. They rarely use it for workflow composition from the scratch, but usually reuse others' workflows modifying them. Building a workflow is not an easy task and it can be compared to writing a computer program.

2.3 User Experience Background

> "User experience is not about the inner workings of a product or service. User experience is about how it works on the outside, where a person comes into contact with it. When someone asks you what it is like to use a product or service, they are asking about the user experience. Is it hard to do simple things? Is it easy to figure out? How does it feel to interact with the product?"
>
> Jesse James Garrett "The elements of user experience" [32]

The "User Experience" (UX) is a concept which was first used in 1995 by User Experience Architect Donald Norman [33]. The term "User Experience" is difficult to define because a common agreed understanding of UX is not reached yet. User Experience can be described as "dynamic, context-dependent, and subjective. It is also seen as something individual (instead of social), that emerges from interacting with a product, system, service or an object" [34]. UX is closely related to the term "Usability". Both are central terms in the Human-Computer Interaction discipline. Let us examine the difference

and relationship between these terms. According to J. Nielsen, Usability considers five basic components [2]:

- Learnability;
- Efficiency;
- Memorability;
- Error tolerance and prevention;
- Satisfaction.

Usability can be presented as the user's ability to complete a task successfully using the tool, while User Experience goes beyond that and takes into account the entire process of the user's interaction with the product, including the user's feelings which result from this interaction. The User Experience measurements are important, but they are based on Usability dimensions [35].

Next, Usability is considered to be a prerequisite for User Experience [36]. User Experience aims to design not only usable software, but pleasurable software as well. Figures 6 and 7 illustrate this relationship between the two terms.

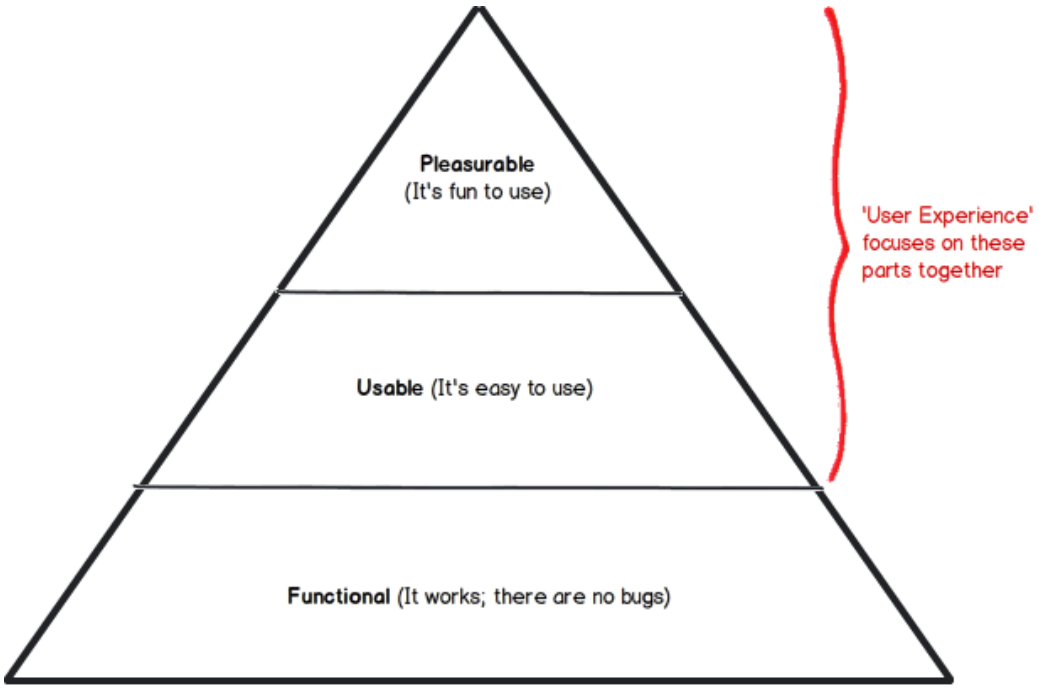


Figure 6. User Experience in the user's hierarchy of
Adapted from [37]

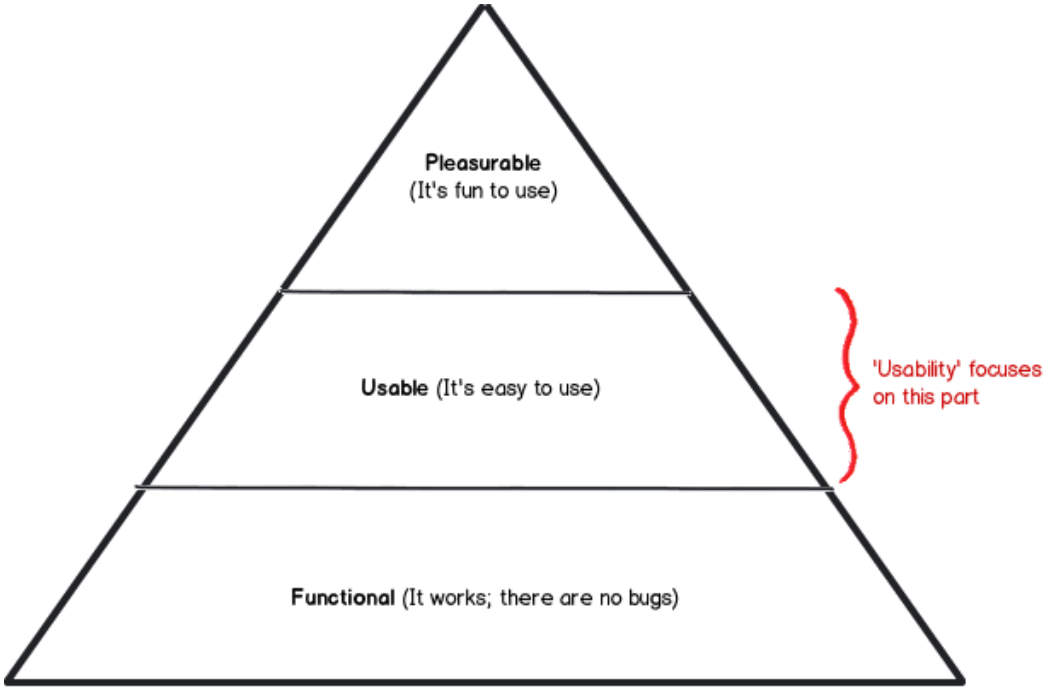


Figure 7. Usability in the user's hierarchy of needs
Adapted from [37]

It can be assumed that the user has three hierarchical categories of needs [37]:

1. Functional – the most basic need: the software must work. This is a prerequisite to usability and UX;
2. Usable: the software should be easy to use. It is a prerequisite to UX;
3. Pleasurable: the software should be enjoyable to use.

A difference between Usability and User Experience can also be made in terms of methods they apply. The goal of the former is to enhance human performance while the latter aims to improve user satisfaction with achieving both pragmatic and hedonic goals. Sometimes the term “User Experience” is used to refer to both approaches [38].

The reason of the growing popularity of the User Experience field in both academia and industry can be the fact that the limitations of the traditional Usability Framework have been understood. The Usability Framework concentrates mainly on performance of a user in the process of human-computer interactions, while User Experience takes into account all the aspects of how people use the system [34].

2.3.1 Grounded Theory methodology and Data Coding

The Grounded Theory is a systematic methodology, which is applied to the Qualitative studies. This methodology allows discovery of a theory during the analysis of data. The important notion of the Grounded theory is an "emergence" of concepts. The researcher does not build in advance or otherwise affect the hypothesis, but observe its emergence during the study. The researcher analyses the data with an open mind, focusing on the characteristics of the data collected [39].

Codes are the most meaningful data extracts, its key points. The coding is a process of dividing extensive data sets into analyzable data pieces by forming the concepts derived from the data [6]. The coding process is divided into two main stages: open coding and selective coding. Open coding is the initial stage of identifying and gathering important concepts in the data. The gathered data is analysed line by line or word by word, and each data extract is constantly compared with the already existing codes in order to identify its characteristics. Selective data coding is the next stage, where a group of categories is associated into one core category. This process delimits the experiment, which is done by going over previously produced codes and coding them again, and it helps to build a theory [40, 41]

Sorting can be applied after the open-coding and selective data coding processes, grouping the codes. Sorting produces new ideas and categories. Sorting is the key process, during which the theory is emerging.

2.3.2 Usability Evaluation methods and Techniques Comparison

Usability evaluation aims to assess the functionality of the tool, to identify the effect it has on users, and to detect any application problems [42]. There are numerous methods for usability evaluation which are divided into four main types such as testing,

elicitation, inspection, and inquiry. A brief description of each type and related methods is presented below.

- *Usability Testing* is the activity which involves observing users interacting with a product, performing particular tasks. Usability testing allows us to see what people actually do, not what we guess they would do or what they assume they would do if they were using a product. The knowledge obtained from the usability testing about the users' experience covers all the sides of design and development [3,43].

  The main benefit of user testing is that it deals with real behaviours of users' representatives, which means that feedback is obtained directly from the target audience. Usability testing focuses on the detailed analysis of the process of users' interaction with the product for accomplishing tasks [44].
- *Usability Elicitation* is a type of usability evaluation method where representatives of real users are observed. It involves users performing a set of tasks interacting with the system while their behaviours are observed and information related to the way participants accomplish the tasks is collected. This method is viewed as one of the most effective methods since the exact information on users' problems can be obtained with the actual interface being tested [2]. Commonly used usability elicitation methods are the Think aloud protocol and Remote Usability Testing [45].
- *Usability Inspection* represents a set of usability evaluation methods for finding usability problems and examining usability-related aspects of the interface [45]. In contrast with usability testing, in Usability inspection the user interface is assessed by the inspector (researcher). Commonly deployed usability inspection techniques are Cognitive Walkthroughs, Heuristic Evaluation and Pluralistic Walkthrough
- In *Usability Inquiry* information related to users' preferences, requirements and understanding of the tool is collected through verbal communications or asking

them to response to given questions in a written form. Commonly used usability inquiry methods are Focus Groups, Interviews and Questionnaires.

Let us examine each of the abovementioned methods highlighting the main issues related to them.

- ✓ *The Think aloud Protocol* is a method where an observed participant uses the product while continuously thinking out. It gives to the researcher an understanding of how the user views the software, their feelings and real thoughts. Moreover, the information about which particular sections of the tool result in the most problems is obtained as this technique demonstrates the users' view regarding each interface item [3]. In the Think-aloud Protocol, users explain their actions while working with the system. The protocol helps to identify why users act in a particular way, especially when the users' behaviour is unexpected. However, it is more obtrusive in comparison to observation and thus can change the process of performing the task [46].
- ✓ *Remote usability* testing is a method which is used when the participants are located in a distance from the usability evaluator. In this method the network acts as a link between evaluators and users, where evaluation is performed with users connected via this bridge and working in their natural work settings [47]. Most of the time audio/video recording is used for conducting usability testing. The recordings are systematically analysed in order to detect usability related issues experienced by the participant [43].
- ✓ *Heuristic evaluation* is an inspection technique which is conducted by having several usability evaluators assessing an interface design. They check whether the interface conforms to usability design requirements [48].
- ✓ In *Cognitive walkthrough* the system's interface is evaluated by a group of inspectors. The tool is assessed in terms of ease of understanding and learning,

particularly by an exploration. The reason for this is that it was noticed that usually users prefer learning how to use the tool by exploration [49].

- ✓ *Pluralistic walkthrough* is a usability method where developers, users and human factors engineers gather to pass step by step through a scenario, considering and assessing the product usability [50].
- ✓ *The Focus groups* method is an informal technique for evaluating user needs and feelings. In a focus group, about six to nine users discuss new concepts and identify issues related to the software usability for approximately two hours [47].
- ✓ *Interview* is a usability inquiry method which concentrates not on the user interface itself but only on the users' views about it. It is a verbal method where the information related to the usability of the product is obtained by directly asking users which features they particularly like or dislike [48].
- ✓ *Questionnaires* usability inquiry method refers to indirect techniques as it does not study the user interface but obtains users' opinions about it. Questionnaire consists of a series of questions which are designed with the purpose of learning the way users use the tool and what is their attitude [2].

A comparison of the abovementioned techniques in terms of applicable stages, advantages and disadvantages as well as the description of each method are given in the Table 1 below.

Evaluation Method	Evaluation Method Type	Applicable Stages	Description	Advantages	Disadvantages
Think aloud protocol	Elicitation	Design, coding, testing and release of application	In this method participants report their thoughts on the system while performing set of tasks	- Less expensive - Results are close to what is experienced by users	- The Environment is not natural to the user
Remote Usability testing	Testing	Design, coding, testing and release of application	The researcher does not directly observe the users while they interact the product, but activity may be recorded for subsequent viewing	- Efficiency, effectiveness and satisfaction - the three usability issues - are covered	- Additional Software is necessary to observe the participants in a distance
Focus groups	Inquiry	Testing and release of application	A moderator guides a discussion with a group of users of the software being evaluated	- If done before prototypes are developed, can save money - Produces a lot of useful ideas from the users themselves - Can improve customer relations	- The environment is not natural to the user and may provide inaccurate results. - The data collected tends to have low validity due to the unstructured nature of the discussion
Interviews	Inquiry	Design, coding, testing and release of application	The test participants are interviewed to learn about their experience and expectations	- Good at obtaining detailed information - Few participants are needed - Can improve customer relations	- Cannot be conducted remotely - Does not address the usability issue of efficiency
Cognitive walkthrough	Inspection	Design, coding, testing and release of application	A team of evaluators walks through the system discussing usability issues through the use of a paper or a working prototype	- Good at refining requirements - Does not require a fully functional prototype	- Does not address user satisfaction or efficiency - The designer may not behave as the average user when using the application
Pluralistic walkthrough	Inspection	Design	A team of users, usability engineers and product developers review the usability of the paper prototype of the application	- Usability issues are resolved faster - Greater number of usability problems can be found at one time	- Does not address the usability issue of efficiency

Table 1. Usability Evaluation Techniques Comparison

Adapted from [51].

2.4 Related Work

Considerable previous research has been conducted in the area of Scientific Workflow Management Systems and Scientific Workflows in general. The scientific community expects the diversity of complex issues to be resolved by workflow management systems. Various solutions were considered and suggested for meeting these expectations.

For example, in the “Scientific Workflow: A Survey and Research Directions” paper [25] Barker and van Hemert examine problems of usability, sustainability and tooling. This work investigates existing workflow systems from both business and scientific domains and draws conclusions regarding future workflow research directions and possible areas of improvements.

V. Curcin and M. Ghanem in their work “Scientific workflow systems - can one size fit all?” [24] give a comprehensive overview and comparison of current leading workflow systems such as Discovery Net, Taverna, Triana, Kepler, Yawl and BPEL. The comparison is made in terms of their control handling and data constructs and attempts to determine a suitable system for a particular task.

McPhillips, Bowers, Zinn and Ludäscher in “Scientific workflow design for mere mortals” [52] review current Scientific Workflows Systems, but they make an emphasis on users – scientists, who have understandably limited programming background. Authors present a set of requirements for scientific workflow systems which would allow ordinary researchers to build the workflows they need easier to support their analyses.

However, with all abovementioned studies in the scientific workflows field, there is a lack of investigation in the particular area of usability of these systems. Only few of this kind of researches have been done.

Gordon and Sensen conducted a pilot usability study of the Taverna workbench in 2007. They describe and discuss the study outcomes in “A Pilot Study into the Usability of a Scientific Workflow Construction Tool” [53]. Study participants represented two groups of Taverna users: programmers and non-programmers, who performed a predefined task in Taverna. User observation and questionnaires were used for assessing the usability of the tool. The difference between the pilot study and the current usability study is that the former concentrated on the difference between the problems programmers and non-

programmers encounter. The latter focuses on the identification of user experience of the tool, that is the users' feeling about the Taverna.

Downey [54] performed a group usability testing for measuring the Kepler workflow system usability. The "Group Usability Testing: Evolution in Usability Techniques" study was conducted in two rounds and in each round a Group usability testing technique was used. As in the case with the Pilot study of Gordon and Sensen, the task was pre-defined. The study was concentrated on introducing and comparing a "group usability testing" method with other usability methods, which is different from the focus of the current study.

"Taverna: lessons in creating a workflow environment for the life sciences" [7] by Oinn, Greenwood et.al is the Taverna assessment from a technical point of view, with anecdotal user observations. The authors discuss the workflows' role in the scientific experimentations environment.

The discussed studies have similarities to Usability Study of the Taverna workbench in terms of their intentions, but they differ in their focus and methods applied.

2.5 Chapter Summary

Scientific Workflows aim help solve the problem of the scientific applications complexity. The environment for running these workflows is provided by the Scientific Workflow Management Systems, one of which is the Taverna Workflow System. The Taverna Workbench is a desktop application that provides a means for exploiting the range of features that the system offers. Users can face the problems during the process of their work with the tool, as it sometimes involves actions which can be difficult either because some Taverna users have no programming background or due to the impressive functionality of the tool. The User Experience aims to identify and address these problems using various techniques and methods, which were described in this chapter. The

participants recruited for the usability experiment are located remotely. For this reason the Remote User Testing technique was applied for the study. User diaries and Think aloud methods were also used in the methodology as they fit in imposed limitations and requirements.

CHAPTER 3. PILOT EXPERIMENT

The chapter discusses the process and results of the preliminary study which was piloted before administering a full scale study. It describes the initial run of an experiment with the purpose of testing the developed methodology and enhancing the study design. This chapter is written in APA (American Psychological Association) format. First, it introduces the participants' recruitment process and materials used. Then the chapter discusses the experiment implementation and methodology. Finally, it presents the experiments findings, followed by the analysis of methodology and suggested improvements.

3.1 Aims and Objectives

The main goal of the pilot experiment was to verify that the methodology is feasible. It also aimed to identify the weaknesses of the developed methodology and redesign it according to findings before running the actual study. In order to meet these aims the following objectives had to be met:

1. Design the initial methodology based on investigated user experience methods and techniques;
2. Recruit representative users for the pilot experiment;
3. Meet users and explain the details;
4. Continuously during three weeks:
    a. Collect the data from the users using suggested techniques;
    b. Analyse the data;
    c. Obtain the results;
    d. Conversation with purpose after each analysed recording;
5. Classify the issues;
6. Refine the methodology based upon results.

One of the benefits of conducting a pilot experiment is that a researcher has an advance warning about methodology main weaknesses and can verify whether suggested techniques are suitable, so the likelihood of the project's failure is decreased.

3.2 Participants and Materials

For the pilot experiment two representative users of the Taverna workbench were recruited and observed. Both participants involved in the pilot study were local users. It helped at the early stages of investigation better understand users and to establish the experiment process. The Table 2 below provides some basic information about the participants which has an impact on the study outcomes.

Participants	Age	Sex	Discipline	PL	Background	Taverna experience	Tool/services experience
Participant 1	40	F	Bio-informatics	Java, Perl, Matlab, R, mysql, php, Javascript, C, C++	Computational scientist	3 times per week (~1 year)	Quite a bit of experience using tools related to discipline, and accessing the main sources of data (ensembl, genbank)
Participant 2	36	F	Helio-physics	Java, C, C++	Programmer	every day for 6 hours (~1 year)	expert in the services assembling

Table 2. Pilot Experiment participants' background information

The pilot experiment participants were recruited by personal intervention by Prof Carole Goble, and the task was explained during one of the Taverna meetings.

The Materials used for the pilot experiment were:

1. Taverna Workbench Software – In the pilot experiment study participants used the Taverna Workbench version 2 for completing real tasks. Figure 5 presented in the Chapter 2 illustrates the Taverna Workbench;

2. Recording Software - For Video Recording, participants employed Camtasia Studio Version 7 screen recording software which has 30-days free trial [55]. It is published by TechSmith and it is used for screen capturing. Camtasia Studio provides flexible screen recording options. Camtasia was chosen because it is not difficult to learn how to use this software and it meets all the requirements for conducting the experiment [55]. The screenshot of the Camtasia Screen Recorder is given in the Figure 8 below;

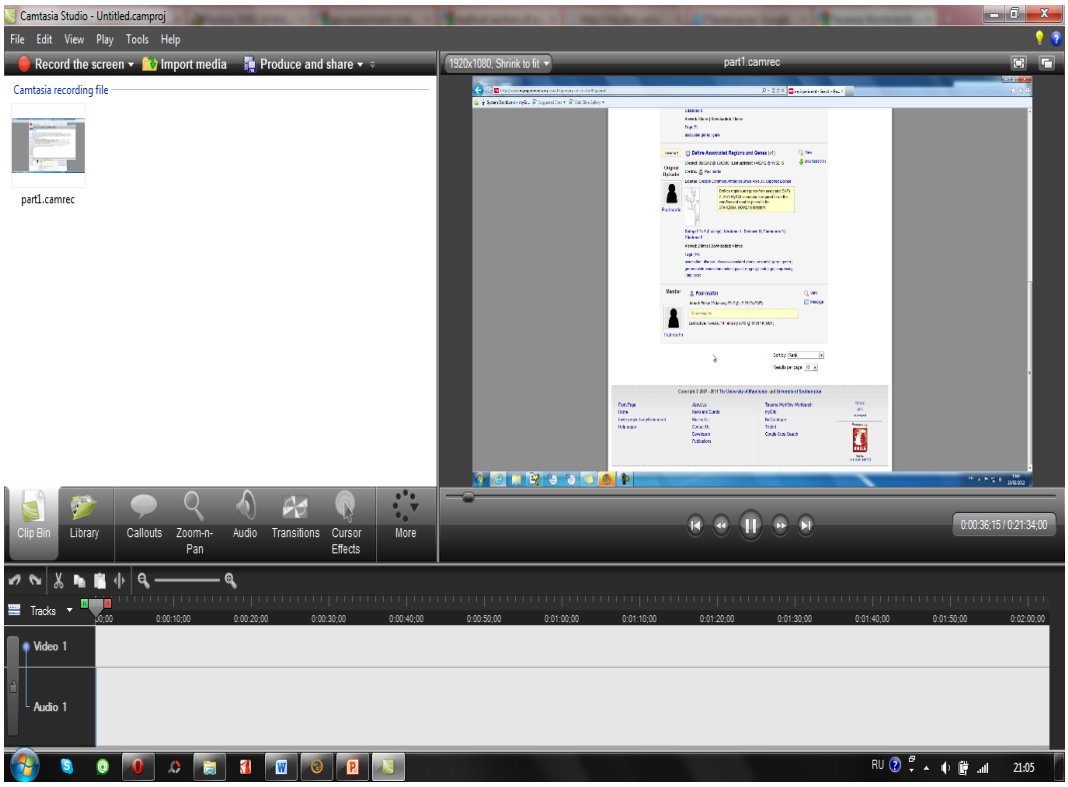


Figure 8. Camtasia Screen Recording Software [55].

3. Qualitative Data Analysis, Research and Coding Software. – For data analysis and data coding during the pilot experiment, AQUAD 6 software was used [56]. AQUAD assists in qualitative research and supports content analysis of open data. It was created in 1987 at the University of Tübingen in Germany. In Figure 9 the screenshot of the AQUAD software is demonstrated.

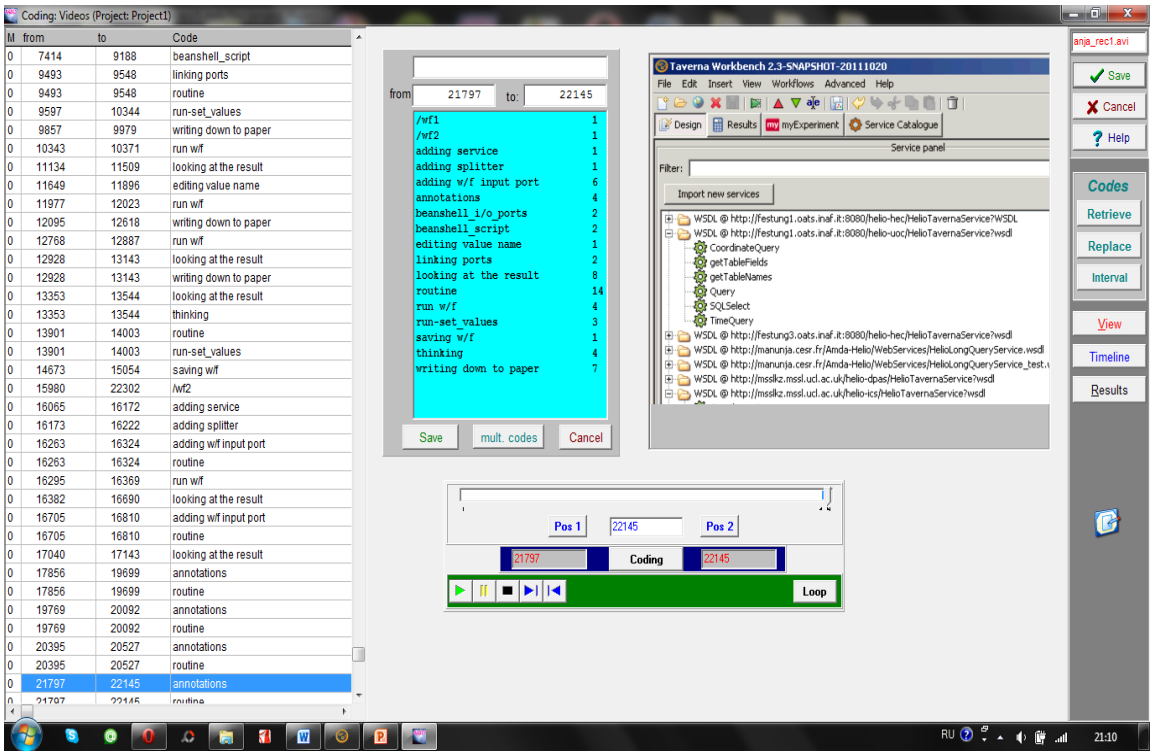


Figure 9. AQUAD 6 qualitative data analysis software [56]

3.3 Research Design

The Research Design section provides the details of the study setup and the description of the pilot experiment process and methodology. This section provides the necessary information for other researchers to conduct their own experiment using the same techniques and to possibly obtain the same results.

3.3.1 Experiment set up

As the project involved human research, ethical approval for the project was obtained. The ethical approval's main purpose was to confirm that the study met the requirements of general ethical values and standards. The ethical approval for the Usability study of the Taverna Workbench project can be found in Appendix A.

The scenario was "Open-ended", where no task was specified. The participants were not asked to perform any pre-defined task, but conducted their usual experiments in

Taverna and recorded it. This allowed the usability researcher to focus on naturally occurring problems. The study was conducted individually with each participant.

The experiment task and details were first explained in person and later study participants were regularly contacted via emails, asking to make contact if any questions/issues arise.

The duration of the pilot experiment was three weeks. One more week was devoted to analysis of the findings and modification of the methodology.

3.3.2 Pilot experiment methodology

The undertaken experiment had the form of the Field study, where the researcher carried out the investigation in natural settings. As it was mentioned before, the participants were asked to perform their usual activities in Taverna, in their usual environment (in their offices/home).

Field studies provide the usability researcher with the opportunity to observe participants in their natural habitat to learn their normal interaction with the system. As opposed to the laboratory testing, in these studies participants use a product in their own environments, with their own equipment and files, bookmarks, and other data. The drawback of the field study can be the fact that the usability researcher has less control over the investigation. But the benefit is that the product is assessed in the actual context in which it is used.

After investigating existing techniques User Testing was chosen as the main technique for the initial methodology. There were three main approaches that had been suggested for the pilot usability study, and were applied in turn:

1. *Remote usability testing with Think aloud protocol*. The benefit of this approach was that the researched had an additional source of information from the Think aloud protocol. The process was the following:

a. Users record their work with Taverna using Camtasia screen recorder.
b. They also Thought Aloud while working, commenting on their actions
c. Afterwards the video was analysed by the researcher to identify the user experience issues.
d. "Conversation with Purpose" was conducted after analysing each video recording. "Conversation with Purpose" is an interview with the user which purpose is to verify the issues revealed after the analysis of the recording.

2. *Remote usability testing only*. In this method Remote usability testing was used without the Think aloud Protocol. The procedure of this method was the same as described above except that users did not make any comments on their work while recording. Instead, the user is given a template to fill in any problems they encounter during the interaction. The benefit of this technique was the natural behavior of users, because they are more likely to forget about recording and work as usual.
3. *Usability testing.* This method was beneficial in a way that it gave to the researcher space to log notes or observe. However the method was intrusive, which caused users to feel uncomfortable and the process of the user working with the system was also affected.
    a. The user's interaction with the tool was observed (the researcher was sitting next to him/her)
    b. Notes were made about the behaviour of the users, problems he/she had or any other observed issues
    c. The user was interviewed afterwards, discussing the assumed problems.

The method which was assumed to be used as an additional source of information at the beginning is Archival analysis. Archival analysis is an observational method, where the researcher examines the collected documents or archives. For the initial methodology the following techniques have been suggested:

1. Examine training material of the Taverna Workbench, looking at the parts which are actually trained, understand why these parts are included and what the problems might be.
2. Examine the Taverna Issue tracker (JIRA) and email archives looking for the usability issues reports.

3.3.3. Procedure

The pilot experiment participants were observed individually. In order to identify the most suitable methodology for the main experiment, the pilot experiment with the first user was performed in three rounds, applying three techniques described above in turn (one technique in each round): Remote usability testing with the Think aloud protocol, Remote usability testing, Usability testing.

During the first meeting the participant was given the instructions. The experiment details were explained, such as which software to use for recording, how long the video recording should be. In the first round, Remote usability testing with Think aloud protocol technique was tested. The participant was asked to perform their usual task (building and enacting a workflow in Taverna), record the interaction using Camtasia recording software and comment while working. The process of the video recording analysis started with converting the recording from MP4 to AVI format, as AQUAD works only with the latter format. After that, the researcher transformed the speech from the video/audio material into a written form, and this procedure is called transcription. Next, the data was coded by labeling segments of transcription and list of codes for this participant was produced. The screenshot of AQUAD software run during the experiment is provided in Figure 9. The list of codes was analysed for repetitions and relationship between the codes. After qualitative coding and analysis of the video, the conversation with the participants was held, discussing the issues identified during the analysis.

In the second round Remote usability testing without Think Aloud protocol was tested. The user was asked to record the work again but with no audio commentary. The assumed benefit of this technique was more favourable environment for observing natural behaviour of the participant. Instead, the template to the participants was given, where they could write any problem encountered, indicating the time when the problem occurred. Both the video and template were coded and analysed for repetitions. The conversation with purpose was hold at the end. The drawback of this method was the lack of information as no voice accompaniment was available.

In the last round, *Usability testing* sitting next to the user was tried. The researcher came to the participant's lab, and observed the process of performing the task. This technique made available more sources of information, such as user's emotions, gestures, and the whole working process. The researcher was sitting behind the user in order not to disturb or affect the participant's work. During the observation, notes were kept on the user's actions the problems he/she faced and the way of dealing with them. The user was interviewed afterwards, discussing the task and identified problems. This method had one significant drawback. The user was conscious about the researcher sitting behind and this influenced the user's behaviour and the natural process of work.

After testing and analysis the suggested techniques with the first participant, the Remote usability testing with Think aloud protocol method was used for the further work with the second user. The video recordings were transcribed, coded and analysed as it was described previously. During the work with the second participant the methodology was established for the full-scale usability experiment.

The difficulties experienced during the course of the pilot experiment included the problem of creating the natural environment where user would feel at ease. Sometimes users did not realise that it was the Taverna software are under scrutiny, not them. Next, as the Taverna is the special software for the particular disciplines, such as Biology,

Bioinformatics, Astronomy, the observer did not always understand all the details in the recording, as she did not know all aspects of the user's particular discipline.

3.4. Results discussion

The outcome of the pilot experiment was the list of identified issues for each of the two participants. The pilot experiment results are informative in terms of methodology shortcomings, but they can be unreliable and therefore they are not provided in this dissertation.

The analysis of the pilot study results was performed by comparing three rounds of the experiment completed with the first participant. It was identified that the methodology used in the first round resulted in the most objective outcomes. It also offered an additional source of information in the form of the Think Aloud Protocol. The participant was likely to forget about the recording and work as usual, which produced more realistic results. This methodology was verified by applying it to work with the second participant.

3.5. Improvements

After analysing the results of the pilot experiment the following refinements to the methodology have been suggested:

- *Use only remote usability testing with Think aloud protocol.* It was identified that Usability testing conducted sitting next to the user influenced the results and lead to unnatural behaviour of the user. The Think aloud protocol was proved to be useful as an additional source of information.
- Apply the *Grounded Theory method and open coding* which would allow categories to emerge. In the Grounded Theory methodology, from the source data codes are produced, next the concepts are formed and from them in turn categories are emerging. Finally, a theory is developed [57,58]. This method supports the subjectivity of the results.

- *Conversation with purpose is excluded,* as participants were located remotely and the researcher did not have an opportunity to meet them every week.
- *Exclude archival analysis,* because the researcher does not have control over how data was collected and previous issues may be outdated.
- Using *the User's Diary* technique as an additional source of information instead. Diary study is a method where users are asked to keep a diary as they are using a product. Using this method different information can be tracked. For example, which mistakes users make, what they learn, and what they find inconvenient or appealing in the tool (or anything which can be interesting to researchers). Afterwards, the diaries are coded and analysed in order to find usage patterns and common issues. The main benefit of this technique is that diaries can reveal information which would be difficult to identify otherwise. Diaries are also one of the geographically distributed qualitative research methods, which allows performing research in remote locations [59].
- *Use another qualitative data analysis and coding tool.* AQUAD data coding software is outdated; it has been suggested to use more convenient and modern ATLASTI Qualitative Data Analysis & Research Software, which will be described in the next chapter.

3.6. Chapter Summary

The pilot experiment of the project allowed testing the study design and methodology, identifying any complexities or inconsistencies at early stages. During the pilot study, suggested techniques and methods were tested, which led to methodology adjustment and improvement. The pilot experiment provided a comprehensive view and objective analysis of the developed basis for the study. Apart from that, the effectiveness

of the software involved in the study was checked. Generally, the results of the pilot experiment showed that suggested methods and overall design of the study are feasible.

CHAPTER 4. EXPERIMENTAL WORK

This chapter describes the implementation of the usability experiment, offering a detailed overview of the performed research. It includes the methods and procedures used in the experiment. The chapter is structured in APA format, starting with the Participants section, where people involved in the study are described in details. It is followed by the Materials part, which provides information regarding equipment used in the study, such as software. Finally, the Procedure section is presented, where the entire process of conducting the experiment is described step-by-step.

4.1 Participants

The section provides relevant information about the study participants. First, limitations of the recruitment process will be discussed, followed by recruitment process itself and user profiles.

4.1.1 Limitations

One of the limitations of this experiment included the users' locations. As it was mentioned earlier, the Taverna workbench is used all over the world, so most of its users are located remotely. This affected the experiment in several ways: first, remote usability techniques had to be applied. Next, there were no opportunities to establish the relationship with users in person. All the communication was carried via emails.

The next limitation was the exceptional use of the system, as the users rarely use the Taverna workbench for actually building workflows, but modifying existing workflows and running them. The reason for this is that the process of creating a workflow is a difficult task, similar to writing a program.

Some participants did not have the opportunity to add the audio commentaries to the video recording, as they were working in the shared environments and did not want to disturb other people.

Finally, the study had a time limit, and participants not always were available at the given period of time.

4.1.2 Recruitment process

Initially three types of users were identified for participating in the study:

- Novices. People in this category have no or little programming experience and very little expertise in using the tool.
- Intermediate users. These have some programming experience and can build satisfactory workflows.
- Experts. These users are programmers and have been using the tool for a long period of time. They build complex workflows using the advanced capabilities of Taverna Scientific Workflow Workbench

However, the representatives of novice users in Taverna are rarely found. The reason behind this is that Taverna workbench is a sophisticated tool which requires at least some programming experience.

For recruiting the users the Taverna Technical manager was consulted, who provided initial information about the users and their emails for contacting them. In the experiment two types of users were engaged: computational scientists and programmers. Participants were asked to refer themselves to one of these groups.

4.1.3 The size of the study groups

The agreed opinion about the sample size which should be used in the usability experiments is not reached yet. Nielson [60], Virzi [61] and Lewis [62] suggested using a small number of participants in a usability testing. As they claim, 5-8 representative users are a sufficient number for identifying about 80% of the usability issues. The idea of using fewer participants for user testing for finding the majority of the problems has been widely supported. Hwang and Salvendy [63] in 2010 are also concluded that the number of usability study participants should be around 10 for discovering the majority of usability problems.

However, some of the recent researches [64] disagree with these statements, re-estimating the sample size of the usability study participants. The author believes, that previous researches in this are did not take into account the fundamental mathematical properties of the problem, and therefore he believes that the sample size number is underestimated. Author claims that an extended statistical model will assist in defining the undiscovered issues number. The number of participants in the experiment will be increased gradually until most of the problems are discovered.

For the usability experiment in the current project we followed the opinion reflected in the most reports on the usability evaluation effectiveness while suggesting using a larger sample size for future work on the topic.

4.1.4 Participants profiles

The average Taverna user is a scientist with experience in a particular domain, such as chemistry, biology, astronomy, etc. As the current experiment findings suggested, most of the Taverna users have computing/programming knowledge to different extend.

In the usability experiment 13 users participated from various domains, which were divided into two main groups depending on their background: programmers and computational scientist, with 5 and 8 users in each group respectively. The age of participants varied from 21 to 58, 5 of them were females and 7 males. All the participants were familiar with both the interface and the task domain.

Tables 3 and 4 below provide information about each participant, including basic facts and information related to the Taverna Workbench use respectively.

User	Gender	Age	Background	Discipline
Participant 1	F	36	programmer	Heliophysics
Participant 2	M	33	programmer	Taxonomic Data Processing
Participant 3	M	31	computational scientist	Digitisation,Cultural Heritage
Participant 4	F	24	computational scientist	Mass spectrometry
Participant 5	M	58	programmer	Biodiversity science
Participant 6	F	40	programmer/computational scientist	Bioinformatics
Participant 7	M	29	programmer	Semantics and Fuzzy Logic
Participant 8	F	35	programmer	Bioinformatics
Participant 9	F	34	computational scientist	Bioinformatics
Participant 10	M	39	computational scientist	Astrophysics
Participant 11	M	27	computational scientist	Biotechnologist
Participant 12	F	32	Computational scientist	Astronomy, e-science
Participant 13	M	21	computational scientist	Bioinformatics

Table 3. Basic information about the main experiment participants

User	Programming Language	Taverna experience	Tools/services experience
Participant 1	Java, C, C++	every day for 6 hours	expert in the services assembling
Participant 2	Java, C, C++,Perl,Python	for 1 year	in-house services, expert
Participant 3	Java, Scala, Python	2-4 hours per week (~3years)	expert in the services assembling
Participant 4	R, Java, awk	2-3 days per week (~6 months)	in-house services, expert
Participant 5	Fortran, LISP & Flavors, Visual Basic, MySQL	once in 2 weeks (~1 year)	in-house services, expert
Participant 6	Java, perl, Matlab, R, mysql, php, Javascript, C, C++	3 times per week (~1 year)	expert in the services assembling
Participant 7	Java, C++	for 3 months	in-house services, expert
Participant 8	Perl, Java	At least once a week (8 years)	expert
Participant 9	Java	every day (~1 year)	in-house services, expert
Participant 10	Perl, Python, PHP, IDL	once a month (~1 year)	Beanshell, AstroTaverna plugins, JDBC Database Connector Plugin
Participant 11	Matlab, C	once a week (~3months)	novice, KEGG database
Participant 12	C, C++, Java,php, python	2 times per week (~1 year)	Python script&Virtual Observatory, intermediate
Participant 13	Java, Python, R	every day (~1 month)	R and Rserve

Table 4. Information related to Participants' Taverna Workbench use

4.2 Materials

The materials used in the experiment are described in this section. It includes recording software and data analysis and research software. Also, ethical approval details obtained prior the experiments are provided.

4.2.1 Ethical approval

As the project involves human research, the ethical approval for the project was requested and approved by the School Ethics Committee. The approval number is CS23. The ethical approval main purpose was to confirm that the study met the requirements of general ethical values and standards. The manual paper version of the application form for approval of a research project is provided in Appendix A.

4.2.2 Recording Software

Software used for the video recordings in the experiment was the Camtasia screen recorder [55]. It has a free trial which was used for the pilot usability study. The Camtasia license for performing the actual usability study has been purchased.

4.2.3 Qualitative Data Analysis and Coding Software

As it was described in the previous chapter, previously for data coding Aquad qualitative data analysis software [56] was used. After testing it during the pilot experiment, some of its drawbacks were identified. First, the use of this software was complicated by the accepted format of the video. Next, the tools provided in the software were insufficient and interface outdated.

Instead, ATLAS.ti software program was employed [65]. It is used in qualitative research for exploring complex data phenomena. The use of this software is quite simple, but at the same time it offers a variety of tools and functions. It has the free trial version

which is limited only in the size of projects that it lets you save, but it is perfect for smaller projects for an unlimited period of time. It provides tools for managing, extracting and comparing meaningful pieces from data.

The process of work in ATLAS.ti starts with creating a project per participant. To this project corresponding data sources, in our case video recordings and Users' Diaries are added. In ATLAS.ti the data sources are called Primary Documents. After that, quotations are formed from the recordings and Users' Diaries. Quotation is an important segment of a video recording which is created by the researcher. Based on these quotations the researcher produces data codes. The separate list of codes is created for each participant from all the videos of this participant. Memos and comments can be added at any stage of the process.

Figure 10 illustrates screenshot showing the main working area of the ATLAS.ti software, including the Primary Document window (1), Quotation manager (2), Code manager (3) and Timeline (4).

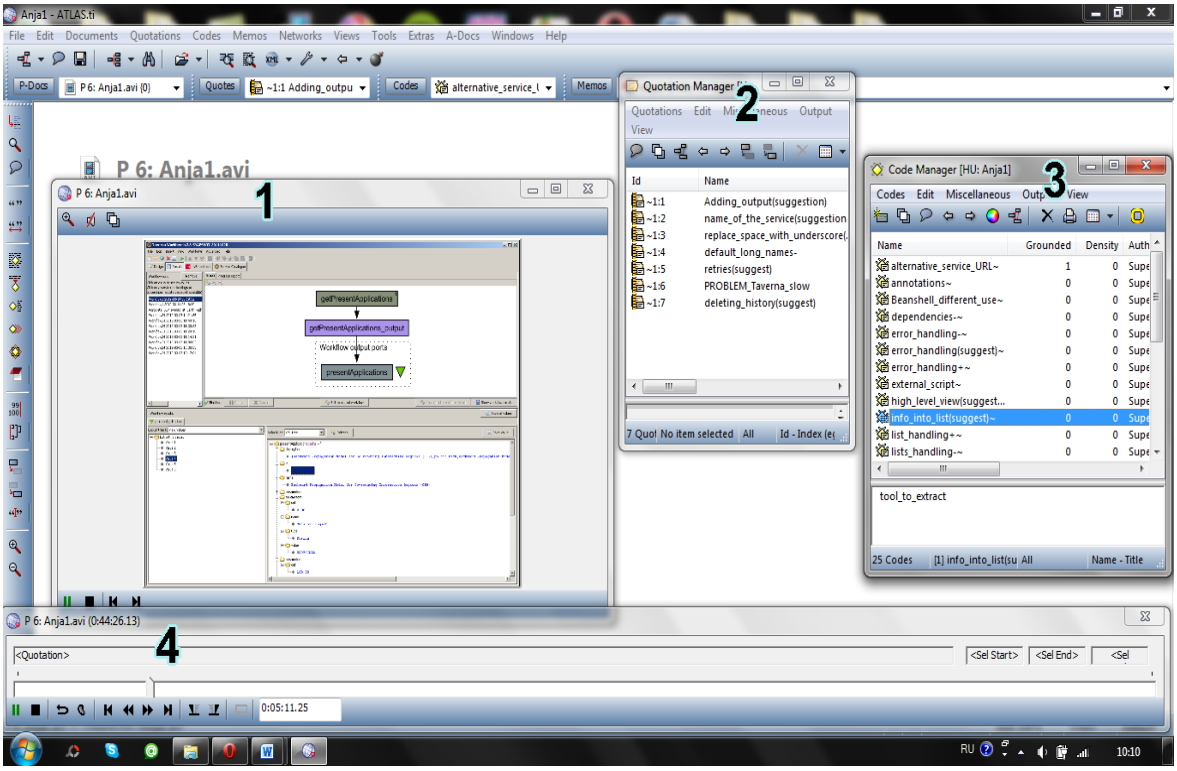


Figure 10. Screenshot showing the main workspace and windows of the ATLAS.ti software, including Primary Documents window (1), Quotation manager (2), Code Manager (3) and Timeline (4).

4.3 Procedure

According to the APA format, the next part of the Experimental work section discusses procedures adopted in the experiment. The methodology description, the data collection process, the steps order as well as the Initial settings are given.

4.3.1 Initial Settings

Before launching the usability study the following arrangements were made:

- Camtasia Studio license was purchased.
- After establishing the contact with users, it was arranged that users are sending the video recordings to the researcher every week, attaching them to the emails.
- The data set from the pilot experiment is kept separate from the usability study data.
- It was agreed that remote usability study was running for two months.

4.3.2 Task Analysis

The users were asked to perform the task in Taverna (version 2.3) and record it with audio commentaries, reporting their thoughts and feelings. The requested video length of the video should be around 30-45 minutes, recording the usual interaction with the Taverna Workbench. There was a diary entry that was just users' notes, of any issues, problems or their wishes, regarding the tool.

As in the pilot experiment, the observer did not specify a definite task to be accomplished and participants were asked to select and perform their own task. This allowed usability researcher to observe naturally emerging problems. This type of tasks is called 'Open-ended', watching participants using the product as they would use it in the real world, to understand their natural behaviour.

4.3.3 Experimental Design

In this usability research Grounded theory methodology was used, which was proposed after investigating the pilot experiment results.

The User Testing, which was conducted within the suggested methodology, which adopted qualitative approach. This approach was discussed in the previous chapters.

The undertaken experiment took the form of *formative usability evaluation*, which concentrates on finding and fixing problems to contribute to further improvement of the system, as opposed to *summative evaluation*, where the focus is on verifying that the product met its usability requirements. Formative evaluation can help tool designers understand better people who are using the system in real situations and on the other hand stimulate user interest and satisfaction with the final product [2,66].

4.3.4 Methodology

The methodology for the usability study of the Taverna Scientific Workflow Workbench was developed after careful examining and analysing usability methods and user experience techniques. It was modified based on the findings from the Pilot Usability Study described in the previous chapter. The methods used in the methodology are considered to be efficient and to fit to the requirements and limitations imposed, such as remote locations of the users (remote usability testing). These methods are: Usability testing, Think-aloud protocol, Users diaries. The first two techniques were applied as described in the pilot experiment. Users' diaries were also coded line-by-line and reviewed for repetitions together with codes from video recordings. An example of a User Diary can be found in the APPENDIX B.

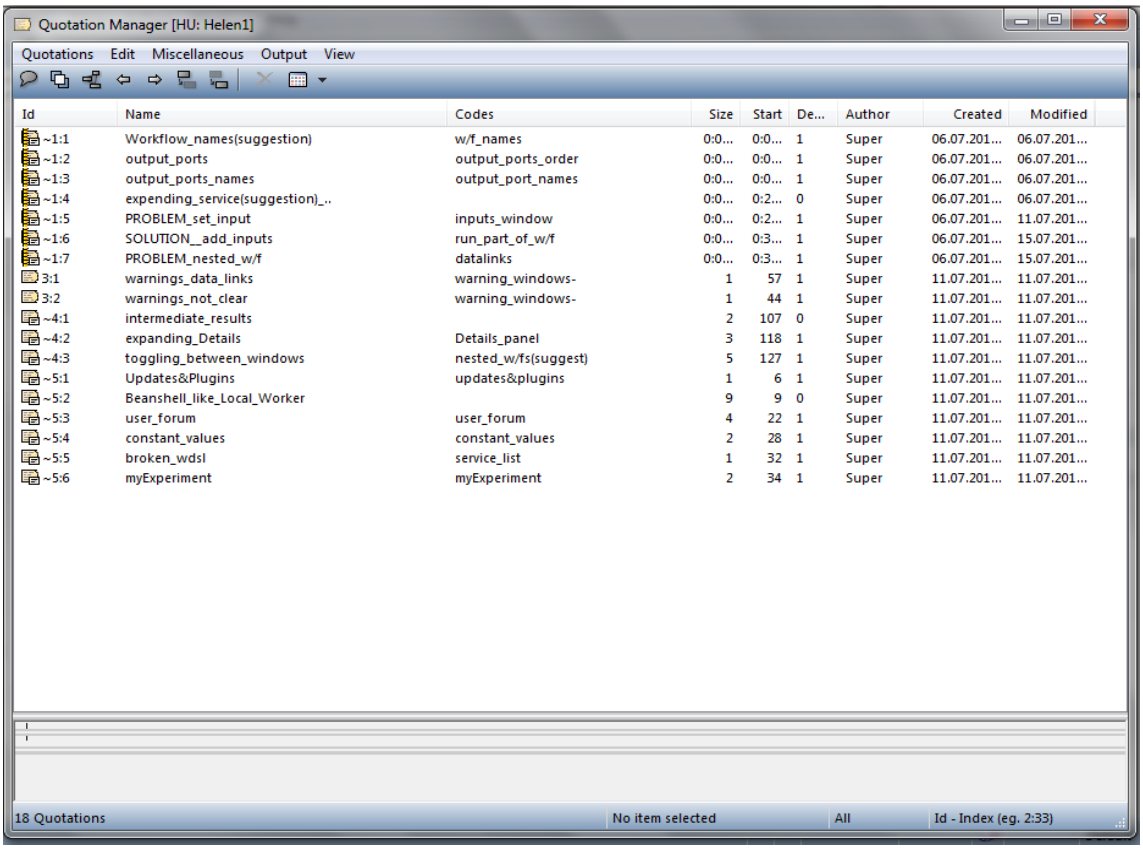
4.3.5 Video recordings

In total 23 recordings were obtained from the experiment participants, the approximate length of each video was 30-45 minutes. The video recordings were two types: with audio commentary and without. As it was mentioned in “Limitations” earlier in this chapter, participants not always could record the audio, as they were working in offices with other people, so they were not following Think Aloud protocol. Recordings without audio commentaries had less data, as the main source of information – audio – was not available. Each video recording was viewed by the researcher several times: first, the video was reviewed for having the overall picture of the recording, then it was pieced into quotations, after that it was open-coded and finally, selective coding was applied. The whole process of analysing one video recording took me 8-10 hours.

4.3.6 Conducting the study

The process of conducting the study after obtaining the required data is described as follows:

1. The analysis of data started by creating a separate project in ATLAS.ti software for each participant, where video recordings and Users’ Diaries of a corresponding participant were added.
2. The video recordings were reviewed using ATLAS.ti software.
3. Next, the list of quotations was produced. It was done by going over the video and User’s Diaries and recording the meaningful parts. An example of a quotation can be: “warning_data_links ”, which is produced from the piece of Users’ Diaries. Figure 11 represents the screenshot of this stage with the list of quotations for one of the participants.



Quotation Manager [HU: Helen1]

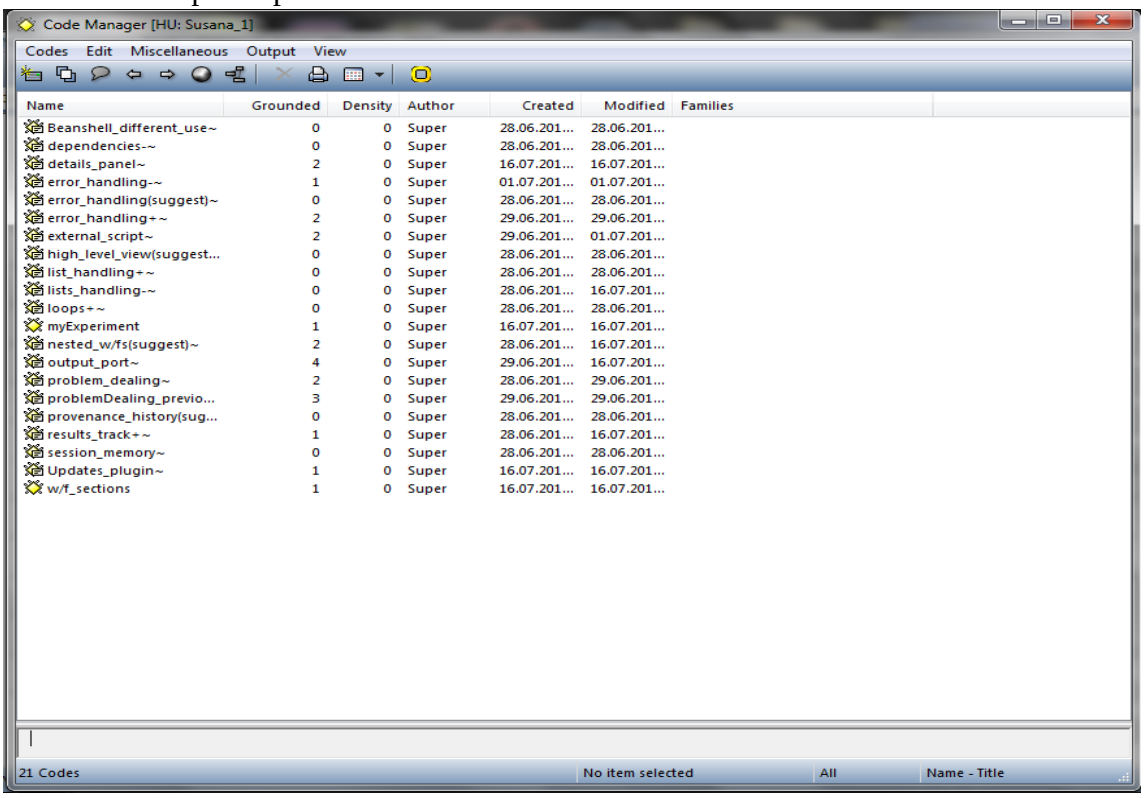
Quotations Edit Miscellaneous Output View

Id	Name	Codes	Size	Start	De...	Author	Created	Modified
~1:1	Workflow_names(suggestion)	w/f_names	0:0...	0:0...	1	Super	06.07.201...	06.07.201...
~1:2	output_ports	output_ports_order	0:0...	0:0...	1	Super	06.07.201...	06.07.201...
~1:3	output_ports_names	output_port_names	0:0...	0:0...	1	Super	06.07.201...	06.07.201...
~1:4	expending_service(suggestion)_..		0:0...	0:2...	0	Super	06.07.201...	06.07.201...
~1:5	PROBLEM_set_input	inputs_window	0:0...	0:2...	1	Super	06.07.201...	11.07.201...
~1:6	SOLUTION__add_inputs	run_part_of_w/f	0:0...	0:3...	1	Super	06.07.201...	15.07.201...
~1:7	PROBLEM_nested_w/f	datalinks	0:0...	0:3...	1	Super	06.07.201...	15.07.201...
3:1	warnings_data_links	warning_windows-	1	57	1	Super	11.07.201...	11.07.201...
3:2	warnings_not_clear	warning_windows-	1	44	1	Super	11.07.201...	11.07.201...
~4:1	intermediate_results		2	107	0	Super	11.07.201...	11.07.201...
~4:2	expanding_Details	Details_panel	3	118	1	Super	11.07.201...	11.07.201...
~4:3	toggling_between_windows	nested_w/fs(suggest)	5	127	1	Super	11.07.201...	11.07.201...
~5:1	Updates&Plugins	updates&plugins	1	6	1	Super	11.07.201...	11.07.201...
~5:2	Beanshell_like_Local_Worker		9	9	0	Super	11.07.201...	11.07.201...
~5:3	user_forum	user_forum	4	22	1	Super	11.07.201...	11.07.201...
~5:4	constant_values	constant_values	2	28	1	Super	11.07.201...	11.07.201...
~5:5	broken_wdsl	service_list	1	32	1	Super	11.07.201...	11.07.201...
~5:6	myExperiment	myExperiment	2	34	1	Super	11.07.201...	11.07.201...

18 Quotations | No item selected | All | Id - Index (eg. 2:33)

Figure 11. Screenshot of the Quotation Manager in the ATLAS.ti software

4. After producing the list of quotations, the data was open-coded for developing initial concepts. The data codes were produced by reviewing the quotations and extracting the key points. Figure 12 illustrates the screenshot of ATLAS.ti software with list of codes for one of the participants with number of occurrences for each code:



Code Manager [HU: Susana_1]

Codes Edit Miscellaneous Output View

Name	Grounded	Density	Author	Created	Modified	Families
Beanshell_different_use~	0	0	Super	28.06.201...	28.06.201...	
dependencies-~	0	0	Super	28.06.201...	28.06.201...	
details_panel~	2	0	Super	16.07.201...	16.07.201...	
error_handling-~	1	0	Super	01.07.201...	01.07.201...	
error_handling(suggest)~	0	0	Super	28.06.201...	28.06.201...	
error_handling+~	2	0	Super	29.06.201...	29.06.201...	
external_script~	2	0	Super	29.06.201...	01.07.201...	
high_level_view(suggest...	0	0	Super	28.06.201...	28.06.201...	
list_handling+~	0	0	Super	28.06.201...	28.06.201...	
lists_handling-~	0	0	Super	28.06.201...	16.07.201...	
loops+~	0	0	Super	28.06.201...	28.06.201...	
myExperiment	1	0	Super	16.07.201...	16.07.201...	
nested_w/fs(suggest)~	2	0	Super	28.06.201...	16.07.201...	
output_port~	4	0	Super	29.06.201...	16.07.201...	
problem_dealing~	2	0	Super	28.06.201...	29.06.201...	
problemDealing_previo...	3	0	Super	29.06.201...	29.06.201...	
provenance_history(sug...	0	0	Super	28.06.201...	28.06.201...	
results_track+~	1	0	Super	28.06.201...	16.07.201...	
session_memory~	0	0	Super	28.06.201...	28.06.201...	
Updates_plugin~	1	0	Super	16.07.201...	16.07.201...	
w/f_sections	1	0	Super	16.07.201...	16.07.201...	

21 Codes | No item selected | All | Name - Title

Figure 12. Screenshot of the Code Manager in the ATLAS.ti software

5. Consequently, the codes were combined into a single list of codes. From this list of codes the List of identified issues was produced, by adding corresponding information to the data codes from the video transcripts and User's Diaries. The list is which is available in the APPENDIX D. This list was analysed for repetitions. Based on the most repeating issues, Local findings were produced. Local findings are individual issues which generally have little impact but can have serious consequences, such as users not being able to complete a particular task. Local findings may seem to be easy-to-fix unimportant issues. However, sometimes they turn out to be global findings, pointing at design or implementation problems, which need to be addressed throughout the product [43].
6. Selective data coding was performed next by going over existing codes and coding them again. As a result, initial categories started emerging. The outcome of this stage was Preliminary groups.
7. Severity ratings were assigned to each group. Severity rating is an impact, which has a preliminary group, taking into account the frequency of mentioning and the number of users which mentioned this particular group.
8. Finally, categories started emerging and they were sorted manually using Card sorting. Card sorting is a technique for organizing data and dividing it into categories, grouping related concepts [36]. First, each problem from the list of identified issued presented above was written on a separate card. Next, the main meaningful words in each quotation were highlighted. The figure 13 below illustrates the card sorting at the beginning of the activity.

Figure 13. Card sorting process

After reviewing and analysing every card, they were grouped into categories, according to their common properties. Next, each group of cards was given a label/name. The result of this process is illustrated in the Figure 14 below.

Figure 14. Sorted cards grouped into categories

During the process of card sorting cards were continuously rearranged, moved from one group to another until the sorting was completed. Finally, the global findings/areas of concern emerged. A global finding is a significant finding which is induced from local findings and reflects problems of design or implementation [67].

Generally, if we compare two groups of users of users presented, programmers performed better than Computational scientists in terms of the time taken for completing similar tasks, although their experience in using Taverna tool was comparatively similar.

4.4 Chapter Summary

This Chapter presented the usability experiment which was conducted as the main part of the project. The whole process started from the participants' recruitment, setting up the environment and a step-by-step description of the study. The methodology, techniques and methods applied were discussed in detail, which made it possible for future researcher to conduct similar studies and compare results. The chapter gives to reader sufficient information for entering into the results discussion, which will take place in the next Chapter.

CHAPTER 5. RESULTS DISCUSSION

The implementation of the usability study was successfully completed and its outcomes are presented in this chapter. It provides two types of findings: local and global findings, as well as the process of generating them. The overall analysis of the obtained data, categories and severity rating are also given.

5.1 Main experiment Outcomes

The experiment results are presented in the following way: first, the data codes as a result of the open-coding process are provided. Then Preliminary groups, formed from these codes, are given along with the severity rating of each problem group. Two types of findings are presented and discussed next: local findings with corresponding recommendations and global findings, which are produced after the card sorting process. Finally, participants' positive comments extracted from the video recordings and world cloud as a visual representation of the experiment results are given.

5.1.1 Data codes

The Table 5 lists all the codes for each user and presents the total number of each code as well the sum of all codes created. These codes are explained in the APPENDIX C. The detailed description of the codes can be found in the List of Identified issues in the APPENDIX D. The different color indicates the different Preliminary group, which is the result of selective coding. This process will be described and discussed later in this chapter.

Participants **Code name**	***P1***	***P2***	***P3***	***P4***	***P5***	***P6***	***P7***	***P8***	***TOTALS:***
alternative_service_URL	1								1
annotations					1	3		1	5
Beanshell_different_use		1	1						2

Participants Code name	*P1*	*P2*	*P3*	*P4*	*P5*	*P6*	*P7*	*P8*	*TOTALS:*
constant_values			1						1
details_panel			1					1	2
error_handling-								1	1
error_handling(suggest)		1							1
error_handling+								1	1
external_script		1					1	1	3
high_level_view(suggest)		1							1
inputs_window			1						1
list_handling+		1							1
lists_handling-		1			2				3
loops+		1							1
memory_allocation						1			1
myExperiment			1						1
nested_w/fs(suggest)		2	1	2		1		1	7
nested_w/fs+						1			1
output_port_names			1			2			3
output_ports	2		1		2	1		3	9
output_ports_order	1		1						2
problemDealing		1					1	1	3
provenance_history	1	1	1						3
python_shell				1				1	1
results_track		1	1					1	3
retries	1								1
run_part_of_w/f			1						1
SAMP_functionality							1		1
script				1					1
service_list			2						2
service_names	3					1			4
session_memory		1							1
Updates_&_Plugins			1					1	2
user_forum			1						1

Participants Code name	*P1*	*P2*	*P3*	*P4*	*P5*	*P6*	*P7*	*P8*	*TOTALS:*
w/f_names			1						1
w/f_sections								1	1
warning window+					1				1
warning_windows-			2						2
XML_splitter	1								1
TOTALS:	10	13	19	3	6	10	4	14	78

Table 5. Codes and total number of their occurrences

5.1.2 Preliminary Groups and Severity ratings

Initially, all the codes were organised into 10 main groups, according to issue characteristics. This preliminary code grouping helped in organizing concepts and forming new ideas.

Table 6 below represents these Categories and also specifies their count, how many users mentioned the issue from this category and lists all related codes for a particular category. The color of each group corresponds to the codes' colors presented above.

№	Groups	Count	№ of users	Codes
1	Details Panel	7	4	Annotations, details panel
2	Script	7	5	Beanshell different use, python shell, external script, script
3	Alert box	2	1	Warning windows +, warning windows -
4	Error handling	4	2	Error handling+, error handling-, error handling(suggestions), problem dealing, retries
5	List handling	4	2	List handling+, list handling-
6	Nested workflows	10	5	Nested workflows+, nested workflows(suggestions), run part of the workflow, workflow sections
7	Output ports	14	5	Output port names, output ports, output ports order

№	Groups	Count	№ of users	Codes
8	Results tab	6	4	Provenance history, results track
9	Services	6	3	Service list, service names, XML splitter
10	Miscellaneous	11	3	Constant values, high level view, (suggest), inputs window, loops+, memory allocation, myExperiment, SAMP functionality, session memory, updates & plugins, user forum, workflow names

Table 6. Categories and codes

Based on information from this table, the severity rating of each category was identified, using the count of codes in each category and the number of users who mentioned this problem. The following levels of the severity were formed: High, Medium and Low. Tables 7 and 8 below define the levels of severity and provide group assignment to each level, according given definition.

Level	Number of codes	Number of users
High	Higher than 7	More than 4 users mentioned
Medium	Between 5 and 7	3- 4 users mentioned
Low	Less than 5	1-2 users mentioned

Table 7. Definition of the Levels of severity.

High	Medium	Low
Script	Details Panel	Alertbox
Nested workflows	Results tab	Error handling
Output ports	Services	List Handling
	Miscellaneous	

Table 8. Groups and severity ratings.

5.1.3 List of Identified Issues

In the Appendix D all the issues identified in the usability experiment are provided. This list was produced from the data codes, by appending to each code the corresponding information, extracted from video transcript and Users' Diaries. The list of identified issues is presented in the APPENDIX D. The original wording extracted from video commentaries is retained.

5.1.4 Global Findings

Card sorting was the next stage, during which gathered data was analysed and organised. The process of the card sorting was described in the previous chapter. As a result of this stage, common patterns, categories and relationships between them were revealed [35]. It led to identifying Global findings of the study. As a result of the card sorting, the following global findings/areas of concern emerged, reflected in Table 9:

Global Findings/Areas of concern	**Descriptions**
Sub workflows/workflow piecing	Participants would like to work with workflow pieces, being able to copy workflow parts, encapsulate part of the workflow into a component, run only part of the workflow, etc.
Visual representation/Navigation	Many issues mentioned by users were related to the visual representation. Users complained that it is difficult and cumbersome to navigate the nested workflow
Defaults and Automates	Most Taverna users agreed that some of the Taverna workbench defaults need to be changed, such as passing single value instead of empty list (where possible) to the nested workflow, services/output port names, inability to use spaces when naming, inability to open several submenus, etc.
Accessibility	Participants complained that it was difficult access required information, for example workflows' constant values, information about the depth of lists created, information about the workflow (title & description) when uploading to myExperiment
Propagation	Participants mentioned information propagation problems, e.g. nested workflows annotations, beanshell information.
Feedback/ Users' support	For example, Taverna users asked for guidelines to name variables, users forum

Global Findings/Areas of concern	Descriptions
Resetting and repetition task	Participants had to repeat some tasks, for example saving nested workflows one-by-one, deleting provenance history on-by-one, to be able to remember certain variable for the entire Taverna session
Annotations	One of the most repeating problems participants faced were related to different concerns regarding annotations
Control over functions/Capability	Users would like to have more capabilities and control, for example to be able to set the default number of retries for all services, rename the services in the service list, to be able to save users' Beanshell to local services, to be able to change memory allocation to Taverna, etc.
Clarity of functions	Sometimes participants did not understand what was going on the screen ("the input window is not disappearing, it is not clear if that means the workflow is running or not"), or did not figure out the purpose of a particular function, what is does (e.g. buttons in the "Updates and Plugins").
Convenience	People mentioned inconvenience issue regarding different matters. For example, writing Python code in a small window, inspecting module output (have to add output ports to check the result),output port order, string constant does not take the name of the service content

Table 9. Global findings and their description

5.1.5 Local findings and recommendations

After analysing all the identified issues listed above, the following list of most repeating issues, which are called local findings, was produced and recommended, which is provided in the Table 10 below:

№	Local finding
1.	Enable button to delete all the provenance history both within Taverna and using keyboard
2.	Replace' Space' with 'Underscore' when naming services/ports
3.	Shorten default Output port names
4.	Allow ordering output ports
5.	Display the name of the workflow at the top of the window when hovering over the pathname. If the pathname is too long, then there is no place where the name of the workflow could be seen
6.	Display a constant value of the workflow more easily accessible, on the diagram. e.g. on hover.
7.	In the Details panel (when selecting an element in the workflow diagram pane) allow expanding several submenus simultaneously (i.e. "Description" or "List Handling" or "Predicted Behaviour")

№	Local finding
8.	Enable enlarging the property tool window, when writing Python code (in “Tools”)
9.	Enable annotations from the nested workflows propagate to the output workflows annotations
10.	Add more fields to annotations: to be able to specify not only authors names, but contributors as well
11.	Allow saving all the nested workflows up in the chain rather than saving each separately
12.	Support expanding nested workflows from the context menu
13.	When the workflow gets large it gets difficult to navigate and add components. The window is too small to get a good overview
14.	Enabling “switching off” parts of the workflow
15.	Allow copying and pasting the entire workflow sections

Table 10. Local findings

5.1.6 Positive impression

Although the study’s goal was to seek areas of difficulties, some of the positive comments users made while working with the Taverna workbench:

1. “List handling in Taverna is intuitive”.
2. “The really good thing, which is nice about this panel displaying the results, is that you can actually go to each component and check the outputs, and it is really useful, because it makes it much easier to debug”.
3. “MyExperiment is lovely and useful”.
4. “Looking at the intermediate results is easy and intuitive, though obviously it can be cumbersome in a large workflow”.
5. “It is convenient that you can just replace nested components and the input and outputs will still be coupled in Taverna”.
6. “Loops in Taverna are nice”.
7. “Taverna has impressive functionality”.
8. “Personally, I like warning windows, because user can find out more about the warnings if he thinks they are important, but it would be too much information each time otherwise”.
9. “Error handling in Taverna is easy”.

5.1.7 Word cloud

Finally, a word cloud is provided below as a visual representation of the identified issues, which was created using the http://www.wordle.net/ online service. The word cloud illustrate in Figure 15 was produced from the list of all identified issues, which is available in the Appendix D (original wording was retained).

Figure 15. Word cloud produced from the list of the identified issues

The size of the words reflects the number of times a specific word was mentioned, so the biggest words are the most repeated. Some of them are “nested” (workflows), “output” (ports), “service”, “components”, “convenient”, “script”, “see” (visual representation), “able” (ability/capability), etc. This word cloud representing the repeated issues is consistent with Global Findings identified during the card sorting.

5.2 Presentation to the Taverna team

The presentation of Usability study results was organised for the members of the Taverna developers’ team during one of the Taverna meetings in August 2012. The presentation demonstrated to the myGrid team the work conducted, discussed study outcomes and suggested recommendations.

The presentation was created using PowerPoint and described the process of work during the usability experiment, showing how study results were produced. All the findings were represented and suggestions given.

5.3 Results Interpretation and Discussion

The results of the Initial grouping showed that Script, Nested workflows and Output ports are the areas of high ratings of severity, which were most mentioned by users. It can be interpreted as follows:

- Script – Group of participants, who used Python scripts within Taverna, indicated that they would like to have a Python shell within Taverna. Many problems were related to script issue. However, this additional functionality may only complicate the use of the system for those users who do not implement Python scripts. So, it can be concluded that Taverna should provide functionality for different types of users, supporting users' preferences, according to their discipline.
- Nested workflows – Participants' comments included inability to run part of the workflow, expand the nested workflow, and copy a component from the nested workflow. As the experiment results suggest users would like to have the ability to use a piece of workflow. Taverna should provide a components approach, allowing users to operate on workflow fragments.
- Output ports – most complains under this code were related to the output ports order and difficulty to see all the output ports when more are added. This issue is closely related to the Visual Representation and Navigation of workflows.

The global findings, which were obtained after the card sorting process, revealed that most problems in Taverna were concentrated around 10 general areas. They can be seen as indicators of Taverna profound problems, which result in the overall complexity of

the tool. The word cloud, which was produced from the initial list of all identified problems, showed that the most repeated words are related to the problem areas.

5.4 Chapter Summary

The Chapter analysed the results of the usability study. There were two main types of findings: local and global findings, which were presented and discussed. Local findings are small issues which can be easily fixed, while global findings are indicators of bigger problems in the Taverna workbench. The recommendations regarding local findings were made and global findings were described, giving directions for improvements.

CHAPTER 6. CONCLUSION AND FUTURE WORK

This dissertation described the process of setting, conducting and analysing the results of the Usability study of the Taverna workbench. The project presented in this work met its aims and objectives set at the early stages. This final chapter reports project achievements, discusses the obstacles overcome, provides reflection on the developed methodology and suggests further work on the topic.

6.1 Project achievements

The main aim of the Usability Study of the Taverna Scientific Workflow Workbench project - understanding and measuring the user experience of the Taverna Scientific Workflow Workbench by conducting a systematic usability study of the tool - was met.

The following achievements were made towards the usability study

- The study methodology was developed;
- The usability study of the Taverna workbench was conducted;
- The results presented to the Taverna team;
- Recommendations to the development team were made.

One of the other important project achievements was establishing the relationship with users which showed the willingness of the Taverna team to meet their needs and made them feel that they are heard.

6.2 Reflection on the methodology

Based on the observations made in this study and challenges encountered the analysis of the produced methodology cam be made. First, let us discuss some of the advantages of the developed approach. The methodology revealed the main local and global usability issues of the Taverna workbench due to combining several usability evaluation techniques.

Study participants were representative users of the Taverna workbench and they conducted tasks in their natural environment. These settings helped gaining more comprehensive and realistic results. It was also attempted to make the tool evaluation as objective as possible, with the observer not interfering with the process, allowing issues emerge instead of building a hypothesis. By using the Grounded Theory method the process of identifying usability problems had an open nature, where participants contributed to the study. The users and the researcher collaborated to develop the theory and obtain the results. The methodology had a low cost, as it used techniques that did not require expensive facilities and tools.

On the other hand, the approach had some limitations. First, the developed methodology did not follow the iterative design due to the time limitation exposed. As a result, there was no opportunity to compare and verify the study outcomes. Next, a clear users' grouping was not possible as users were from diverse domains, with different backgrounds, different ages and with a difference in Taverna Workbench experience .The conducted experiment had reduced control over the participants and testing environment as the remote user testing technique was used. With remote user testing it was also difficult to build rapport and trust between evaluator and participants. Finally, the participants' facial expressions and non-verbal clues were not available as an additional source of information as users were located remotely.

6.3 Future work

In view of the limited time available for this project and resource constraints there are several recommendations for future work.

- Summative evaluation: The future experiment can differ in term of its goal: the current study had the form of formative assessment concentrating on the areas of improvement. The proposal for future work is performing summative

evaluation with documenting the User Experience of a product at the end of the development cycle.

- Iterative design: An experiment can be conducted in several sessions, where after each session identified problems will be taken into account and changes will be applied. The experiment can then be repeated and results compared.
- Experiment setup: the usability experiment can be conducted as a controlled laboratory study, where participants are recruited and grouped according to their background, experience, age, etc. All the participants can perform the same task, and this would allow comparing the groups' performance, obtaining the indicators of expertise, observing the learnability effect. The indicators of expertise would help to identify the level of the user proficiency: e.g. expert, intermediate, beginner. Examples of such indicators could be the number of failures of workflow run, how many tries user makes until the workflow finishes successfully , how many times user is looking for something on the web, how many times user checks previous workflow to complete current work or how long does it take to complete the work .
- Usability evaluation methods: different techniques applied to different groups of users and then compare and contrast the obtained results, for example, applying group user testing and individual user testing.
- Incentives: incentives can be offered to stimulate the participation of users and reward for their time investment.
- Greater number of participants: conduct the experiment in the field with much larger group of participants.

The listed recommendations are a minimal set, proposed for enhancing presented methodology and are by no means exhaustive.

6.4 Obstacles overcome and identified risks

There were several obstacles and risks to the project which were identified and overcome:

1. Participants are not recruited for the study or they are not representative users. Usability study participants were recruited in advance and Taverna team contribution was requested. In order to ensure that they are real representative users of Taverna Workbench one of the members of Taverna software development, who deals with users issues within Taverna team, was consulted.
2. Remote location of the users. Remote usability testing technique was proposed to solve this problem, which makes use of particular software to record users' actions.
3. Infeasible methodology techniques. In order to verify the proposed methods and techniques and refine the initial methodology Pilot usability study was conducted.
4. Researcher affects the process and results of the study. During the pilot study possible effects on the study process and results from the researcher were identified and addressed.
5. Unrealistic results. For yielding more reliable study outcomes several techniques were combined, where results of the one method were compared to the results of the other. For example, two datasets were built, one of them from usability testing and the other from user's diaries. The obtained results of both methods were compared.
6. Participants feel that it is not the interface, but their work being evaluated. The observer made sure that participants were informed that the study is not about their performance, but the performance of the Taverna workbench.

The Usability Study of the Taverna workbench project was challenging, but very interesting. It allowed gaining extensive knowledge in the usability field, experience in working with people from different countries and from the Taverna team.

There is a lack of investigation in the area of usability of complex scientific tools. The usability experiments of these tools are essential, as they put the user at the center of the development process, taking into account his needs and wishes. Tool developers are experts in their field and things which seem obvious to them might be difficult for the end-users. Conducting a usability study helps take a step back and understand users better.

We believe that this project was beneficial both for Taverna developers and its users. It influenced the direction of the Taverna Workbench and the Taverna team is tackling identified issues as a direct result of this work.

LIST OF REFERENCES

[1] Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M., Li, P., Oinn. T.. Taverna: a tool for building and running workflows of services. Nucleic Acids Research. 2006; 34: 729-732.

[2] Nielsen, J. Usability engineering. Boston, MA: Academic Press; 1993.

[3] Denzin, Norman K. & Lincoln, Yvonna S. (Eds.). (2005). The Sage Handbook of Qualitative Research (3rd ed.). Thousand Oaks, CA: Sage. ISBN 0-7619-2757-3

[4] Wolstencroft, K., Fisher, P., De Roure, D., Goble, C. Scientific Workflows. In: Research In a Connected World. In Voss A., Vander Meer E., Fergusson D (Eds.):2009. Retrieved from the Connexions Web site: http://cnx.org/content/m32861/1.3/

[5] CASIMIR consortium, Mouse functional genomics. Scientific Workflow. [Accessed: 05/09/2012 - http://www.myexperiment.org/workflows/126.html]

[6] Lockyer, Sharon. "Coding Qualitative Data." In The Sage Encyclopedia of Social Science Research Methods, Edited by Michael S. Lewis-Beck, Alan Bryman, and Timothy Futing Liao, v. 1, 137-138. Thousand Oaks, Calif.: Sage, 2004.

[7] Oinn T., Greenwood M., Addis M., Alpdemir M., Ferris J., Glover K., Goble C., Goderis A., Hull D., Marvin d., Li p., Lord Ph., Pocock M. R., Senger M., Stevens R., Wipat A., Wroe Ch.. Taverna: lessons in creating a workflow environment for the life sciences. Concurrency and Computation: Practice and Engineering. 2005; 18:1067–1100

[8] Singh M.P., Vouk M.A., "Scientific workflows: scientific computing meets transactional workflows," Proceedings of the NSF Workshop on Workflow and Process Automation in Information Systems: State-of-the-Art and Future Directions, Univ. Georgia, Athens, GA, USA; 1996, pp.SUPL28-34.

[9] Chen J., W.M.P. van der Aalst. On Scientific Workflow. TCSC Newsletter, IEEE Technical Committee on Scalable Computing; 2007: 9(1).

[10] Goble, C., De Roure, D. The impact of workflow tools on data-centric research. In Data Intensive Computing: The Fourth Paradigm of Scientific Discovery. In Hey T., Tansley S., Tolle K.(Eds); 2009:pp. 137-145.

[11] Tanoh, F.Get Weather Information. Scientific Workflow. [Accessed: 05/09/2012 – http://www.myexperiment.org/workflows/146].

[12] Gil, Y. From Data to Knowledge to Discoveries: Scientific Workflows and Artificial Intelligence. In Scientific Programming. 2009; 17 (3): pp. 231-246.

[13] Wassink, I., van der Vet, P. E., Wolstencroft, K., Neerincx, P. B. T., Roos, M., Rauwerda, H., Breit, T. M. Analysing scientific workflows: why workflows not only connect web services. In: IEEE Congress on Services 2009; 06-10 July 2009, Los Angeles, CA, USA.

[14]. Gil, Y., Deelman, E., Ellisman, M. H., Fahringer, T., Fox, G., Gannon, D., Goble, C. A., Livny, M., Moreau, L., Myers, J. Examining the Challenges of Scientific Workflows. IEEE Computer; 2007. 40(12):24-32.

[15] Wassink I., P.E van der Vet, Wolstencroft K., Neerincx P.B.T., Roos M., Rauwerda H., Breit. Analysing Scientific Workflows: Why Workflows Not Only Connect Web Services. In IEEE Congress on Services; 2009.

[16] myExperiment website. Home page. [Accessed: 05/09/2012 http://www.myexperiment.org/]

[17] De Roure, D., Goble, C. and Stevens, R. (2009) The Design and Realisation of the myExperiment Virtual Research Environment for Social Sharing of Workflows. Future Generation Computer Systems 25, pp. 561-567

[18] The Kepler Project. Home page. [Accessed: 05/09/2012 - https://kepler-project.org]

[19] VisTrails Workflow System. Project home page. [Accessed: 05/09/2012 http://www.vistrails.org]

[20] Triana problem Solving Environment. Home page. [Accessed: 05/09/2012 http://www.trianacode.org/]

[21] Yang, X.Y., Bruin, R.P., Dove, M.T. Developing an End-to-End Scientific Workflow. A Case Study Using a Comprehensive Workflow Platform in e- Science. In Computing in Science & Engineering. 2010; 12: 52-61.

[22] Professional Open-Source Software. Home page [Accessed: 05/09/2012 http://www.knime.org/]

[23] McIver R., Jones A., White R. Workflow Systems for Biodiversity Researchers: Existing Problems and Potential Solutions. In Proceeding of Biodiversity Informatics: challenges in modelling and managing biodiversity knowledge; 2008.

[24] V. Curcin, M. Ghanem. Scientific workflow systems - can one size fit all? In Proceedings of the 4th Cairo International Biomedical Engineering Conference, CIBEC 2008. IEEE; 18-20 December 2008. pp.1-9

[25] Barker, A., van Hemert, J. Scientific workflow: A survey and research directions. Parallel Processing and Applied Mathematics (PPAM 2008). 4967. Springer-Verlag; 2008. p. 746-753.

[26] GridNexus: A Grid Services Scientific Workflow System Jeffrey L. Brown, Clayton S. Ferner, Thomas C. Hudson, Ann E. Stapleton, Ronald J. Vettera, Tristan Carland, Andrew Martin, Jerry Martin, Allen Rawls, William J. Shipman, and Michael Wood University of North Carolina Wilmington, USA

[27] Wolter, R. (2001) Extreme XML: Simply SOAP. Microsoft Corporation online library.
[Accessed: 18/Aug/2012 – http://msdn.microsoft.com/en-us/library/ms950803.aspx].

[28] Berthold M., Cebron N., Dill F., Di Fatta G., Gabriel T., Georg. F, Meinl T., Ohl P., Sieb Ch., Wiswedel B. KNIME: The Konstanz Information Miner. In SIGKDD Explorations. 2009; 11 (1): 26-31

[29] P. Missier, S. Soiland-Reyes, S. Owen, W. Tan, A. Nenadic, I. Dunlop, A. Williams, T. Oinn, and C. Goble, "Taverna, reloaded," Procs. SSDBM 2010, M. Gertz, T. Hey, and B. Ludaescher, Heidelberg, Germany. 2010.

[30] Sroka J., Hidders J., Missier P., Goble C.. A formal semantics for the Taverna 2 workflow model. J. Comput. Syst. Sci.2010; 76(6): 490-508

[31] Building Scientific Workflow with Taverna and BPEL: a Comparative Study in caGrid Wei Tan1 , Paolo Missier2, Ravi Madduri3 and Ian Foster1

[32] Garrett J. The Elements of User Experience: User-Centered Design for the Web and Beyond, New York: New Riders; 2010

[33] Norman D., Miller F., Henderson A. What You See, Some of What's in the Future, And How We Go About Doing It: HI at Apple Computer. Proceedings of CHI. Denver, Colorado, USA. 1995; p.155

[34] Law E., Roto V., Hassenzahl M., Vermeeren A., Kort J. Understanding , Scoping and Defining User eXperience : A Survey Approach. CHI '09 Proceedings of the 27th international conference on Human factors in computing systems. 2009 .Boston, MA, USA. 79(3): 719-728

[35] Wilson Ch. User Experience Re-Mastered: Your Guide to Getting the Right Design. Burlington, MA, USA: Elsevier; 2010

[36] Kurosu M.(Ed.). Human Centered Design. First International Conference, HCD 2009. Held as Part of HCI International 2009. San Diego, CA, USA, July 19-24, 2009; Proceedings. Lecture Notes in Computer Science, 5619, Springer: 2009.

[37] Michael Heraghty, The Difference Between UX and Usability, 2011. On-line [Accessed: 05/09/2012 - http://www.userjourneys.com/blog/difference-ux-usability/]

[38] Bevan N. What is the difference between the purpose of usability and user experience evaluation methods? In INTERACT 2009. UXEM'09 Workshop. Uppsala, Sweden; 2009.

[39] Patricia Yancey Martin & Barry A. Turner, "Grounded Theory and Organizational Research," The Journal of Applied Behavioral Science, vol. 22, no. 2 (1986), 141.

[40] Kelle, U. (2005). "Emergence" vs. "Forcing" of Empirical Data? A Crucial Problem of "Grounded Theory" Reconsidered. Forum Qualitative Sozialforschung / Forum: Qualitative Social Research [On-line Journal], 6(2), Art. 27, paragraphs 49 & 50

[41] Thornberg, R., & Charmaz, K. (2012). Grounded theory. In S. D. Lapan, M. Quartaroli, & F. Reimer (Eds.), Qualitative research: An introduction to methods and designs (pp. 41-67). San Francisco, CA: John Wiley/Jossey-Bass.

[42] Dix A., Finlay, J., Abowd, G., Beale, R.. Human-Computer Interaction. Prentice Hall International, UK, Hemel Hampstead: 1998.

[43] Brinck T., Gergle D., Wood S. User Needs Analysis. In Chauncey Wilson (Ed). User Experience Re-Mastered. Your Guide to Getting the Right Design., Burlington, MA, USA :Elsevier; 2010. pp. 61

[44] M. Matera, F. Rizzo, G. Toffetti Carughi. Web Usability: Principles and Evaluation Methods. In E. Mendes, N. Mosley (eds.), Web Engineering. Springer Verlag; 2006, pp. 143-180.

[45] Matera M., Rizzo F., Toffetti Carughi G.. Web Usability: Principles and Evaluation Methods. In: E. Mendes, N. Mosley, (Eds.) Web Engineering. Springer; 2006. Pp. 143-180,

[46] Brinck T., Gergle D., Wood S. User Needs Analysis. In Chauncey Wilson (Ed). User Experience Re-Mastered. Your Guide to Getting the Right Design., Burlington, MA, USA :Elsevier; 2010. pp. 61

[47] Castillo, J. C., Hartson, H. R. and Hix, D. Remote usability evaluation: Can users report their own critical incidents? Proceedings of CHI 1998, ACM Press ;1998. 253-254.

[48] Scholtz J. Usability Evaluation. National Institute of Standards and Technology. 2004.

[49] Wharton, C., Rieman, J., Lewis, C., and Polson, P. The Cognitive Walkthrough Method: A Practitioner's Guide. In Nielsen, J. and Mack, R. (eds.), Usability inspection methods, John Wiley & Sons, Inc., New York; 1994. pp. 105-140.

[50] Nielsen, J. Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.). Usability Inspection Methods. New York, NY: John Wiley and Sons. 1994.

[51] Pauline. G. Usability Evaluation: Methods and Techniques: Version 2. University of Texas. 2002

[52] McPhillips T., Bowers Sh., Zinn D., Ludäscher B. "Scientific workflow design for mere mortals" in Future Generation Computer Systems, Vol 25 Issue 5, 2009. Pp. 541-551

[53] Gordon P., Christoph W. Sensen. A Pilot Study into the Usability of a Scientific Workflow Construction Tool

[54] Downey L., Group Usability Testing: Evolution in Usability Techniques. In Journal of Usability Studies. Vol. 2, Issue 3, May 2007, pp. 133-144

[55] TechSmith website. Camtasia Studio. On-line [Accessed: 05/09/2012 - http://www.techsmith.com/camtasia.html]

[56] AQUAD website. Homepage. On-line [Accessed: 05/09/2012 - http://www.aquad.de/en/]

[57] Kelle, U. (2005). "Emergence" vs. "Forcing" of Empirical Data? A Crucial Problem of "Grounded Theory" Reconsidered. Forum Qualitative Sozialforschung / Forum: Qualitative Social Research [On-line Journal], 6(2), Art. 27, paragraphs 49 & 50

[58] Faggiolani, Chiara, "Perceived Identity: applying Grounded Theory in Libraries," JLIS.It, vol. 2, no. 1 (2011). doi:10.4403/jlis.it-4592.

[59] Kuniavsky M. Ongoing Relationships. In Kuniavsky M., Observing the User Experience. A Practitioner's Guide to User Research, Burlington, MA, USA: Elsevier; 2003. pp. 369-370

[60] Virzi, Robert A. “Streamlining the Design Process: Running Fewer Subjects.” In D., Woods, and E., Roth, (Eds.). Proceedings of the Human Factors Society, Santa Monica, USA; 1990. pp 291-94.

[61] Lewis, J. R. Sample Sizes for Usability Studies: Additional Considerations. Human Factors. 1994; 36: 368-378.

[62] Nielsen, J. “Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier in Bias”. In Randolph, G. , Mayhew, Deborah J. (Eds.). Cost-Justifying Usability. Burlington, MA: Academic Press; 1994. pp 245-272.

[63] Hwang, W. and Salvendy, G. Number of people required for usability evaluation: The 10±2 rule. Commun. ACM 53, 5 (May 2010), 130–133.

[64] Schmettow, M. “Sample Size in Usability Studies”. In Communications of the ACM Magazine, Vol. 55, Issue 4, April 2012, pp. 64-70 , ACM New York, NY, USA.

[65] ATLAS.ti: The Qualitative Data Analysis & Research Software website. On-line [Accessed: 05/09/2012 http://www.atlasti.com/index.html]

[66] Janice (Ginny) Redish, Rolf Molich, Randolph G. Bias, Joe Dumas, Robert Bailey, Jared M. Spool. Usability in Practice: Formative Usability Evaluations — Evolution and Revolution.

[67] Barnum C. Establishing the essentials. In Barnum C. Usability Testing Essentials: Ready, Set...Test! Burlington, MA, USA: Elsevier; 2011. pp.9-20

APPENDIX A. ETHICAL APPROVAL APPLICATION FORM

MANCHESTER 1824

The University of Manchester

COMMITTEE ON THE ETHICS OF RESEARCH ON HUMAN BEINGS

Application form for approval of a research project

This form should be completed by the Chief Investigator(s), after reading the guidance notes.

Project Details:

Title: Usability Study of the Taverna Scientific Workflow Workbench

Abstract: A scientific workflow represents a multi-step experimental process, protocol, or methodology. They are used to encode and run repetitively executed scientific data and analytical pipelines. Workflows are constructed from chaining together private, in house or public, third party services.

The Taverna workbench and execution engine (http://www.taverna.org.uk), developed by the myGrid project (http://www.mygrid.org.uk), enables researchers to construct and execute workflows that link together distributed analysis tools and data resources. It is an open source workflow management system that has achieved wide adoption in the scientific community, including Biology, BioDiversity, HelioPhysics, Astronomy, and Image processing of ancient documents. Workflows are typically designed using a graphical user interfaces and look like node and link graphs. myExperiment (http://www.myexperiment.org), a public repository and web collaboration space also developed by the myGrid team, holds over 2000 workflows.

Study Details:

The study type is: Postgraduate usability evaluation

Study Title: Usability Study of the Taverna Scientific Workflow Workbench

Abstract: The Taverna Workbench is a sophisticated tool, and workflows are often complex things composed using complex and non-harmonised steps. This project is to run a systematic usability study of the workbench, with access to its users, and make usability recommendations to the development team.

Applicants: *Kymbat Yeltayeva.

1: Proposed start date of the study

26.03.2012

2: Anticipated completion date for the study

07.09.2012

3: What is the principal research question/objective?

The Taverna Workbench is a sophisticated tool, and workflows are often complex things composed using complex and non-harmonised steps. This project is to run a systematic usability study of the workbench, with access to its users, and make usability recommendations to the development team.

4: What is the scientific justification for the research? What is the background? Why is this

an area of importance? Has any similar research been done already?

The scientific workflows management systems are designed to facilitate researchers' needs by providing great capabilities of the tool. However, often usability aspect in this case is overlooked.

It is important to make the tool maximally convenient and easy to use for any type of users. The aim of the project is to run a usability study of the Taverna Scientific Workflow Workbench, identify the problems associated with using the software and make usability recommendations to the Taverna software development team.

5: Give a full explanation of the purpose, design and methodology of the planned research.

It should be clear exactly what will happen to the research participant, how many times and in what order.

The evaluation is to help determine the usability of this postgraduate project. As such the participants will engage in a 15 minute training period in which the functionality under evaluation will be shown. After this a 30 minute directed evaluation will be undertaken using the 'Think Aloud Methodology'. The evaluation itself will comprise a maximum of

25 directed activities at which time the evaluator will make written notes relating to the comments and suggestions of the participant. These notes will be formally transcribed after the evaluation taking due care to anonymize the participant information as well as any comments or notes which could lead to the participants identification being deduced by third-parties. The Think Aloud methodology is a well understood evaluation process evolving mainly from design based approaches. In this case it will produce qualitative data and will occur as part of an observational process (and is therefore not a direct measurement of participant performance, as would be normal in more formal laboratory settings). 'Think Aloud' requires the evaluation activities to be completed, however it is not the direct measurement of those activities. Instead, it is the associated verbalisations of the participants as they progress through the activities describing how they are feeling, what they think, and what they think they need to do. In this case, we wish to understand explicitly the activities and thoughts of the user, as they are performing the evaluation activities specific to this evaluation. The main risk with 'Think Aloud' is that it is very easy to implicitly influence the participant into providing outcomes that are positive regardless of the true nature of the interface or interaction. Indeed, the very act of verbalising their thoughts and feelings means that participants often change the way they interact with the system.

6: Describe the methods that will be used to analyse the data collected in the study.

The evaluator will analyse the data. This will take the form of drawing conclusions regarding usability from common themes and user experiences reoccurring throughout the formal transcripts. Understanding common positive and negative aspects of the user experience will enable future work to be suggested and/or changes to be made to the artifact currently under evaluation.

7: How many participants will be recruited?

13

8: Provide details of the participants.

Male and female Taverna scientific workflow workbench users who are between the ages of 21 and 58.

9: Will the participants be from any of the following groups? (Tick as appropriate)

None of the above

10: Will you have direct contact with participants?

Yes

11: How will you identify and select participants?

Networks and recommendations

12: Please enter the text used for recruitment.

I would like to ask you to participate in the usability study of the Taverna Scientific Workflow Workbench as a Taverna user. The study is a part of MSc project. Users are expected to record their interaction with the tool while they are working on Taverna using screen recording software and describe how they are feeling and what they think they need to do.

Anticipated duration of participation for each participant is one hour or less. Please, note that the main goal of the study is workbench testing, not the users' ability to work with the tool.

13: Will participants receive an incentive for taking part?

No

14: What is the potential for adverse effects, risks or hazards for research participants, including potential for pain, discomfort, distress or inconvenience?

It is not anticipated that there will be any physical discomfort associated with the study, but it is possible that some participants may find performing the evaluation difficult, and therefore stressful. Before the evaluation starts, participants will have time to practice using the new software, and getting used to the commands, and they will also be able to ask questions at any point during the evaluation. Participants will be free to take a break or withdraw from the evaluation at any point.

15: Will individual or group interviews/questionnaires discuss any topics or issues that might be sensitive, embarrassing or upsetting, or is it possible that criminal or other disclosures requiring action could take place during the study (e.g. during interviews/group discussions)?

No

16: How long do you anticipate the total duration of participation for each participant?

One hour or less

17: What is the potential for adverse effects, risks or hazards, pain, discomfort, distress, or inconvenience to the researchers themselves?

It is not anticipated that there will be any risks to the experimenter associated with the study.

18: How will risks or inconvenience to the participant/researcher be minimised?

It is not anticipated that there will be any risks to the experimenter associated with the study.

19: Will a signed record of consent be obtained?

Yes

20: How long after they receive the information sheet will participants have to decide whether to take part in the research?

More than 24 hours

21: Will you be using any of the following forms of data recording?

Video recording

22: Where will the experiment take place?

University of Manchester premises

23: Will the research be carried out wholly within the UK?

Yes

24: Please confirm that data will be:

Obtained and used only in the way(s) for which consent has been given

Fairly and lawfully processed

Processed for limited purposes

Adequate relevant and not excessive

Accurate

Not kept longer than necessary

Processed in accordance with the participant's rights

Secure

Not transferred to settings without adequate protection.

25: What measures have been put in place to ensure confidentiality of personal data? Give details of whether any encryption or other anonymisation procedures have been used and at what stage.

All data from participants will be stored under a subject number. This number will not be linked

with the participant's name, providing anonymity.

26: Where will the data analysis take place?

A private study area

27: Will the data be stored in a secure place (e.g., a locked drawer, accessible only to the researcher, or secure, password protected electronic files.) at all times?

Yes

28: Who will control the data generated during the study and act as its custodian?

The researcher

The supervisor

29: Who will have access to the data generated by the study?

The researcher

The supervisor

30: Will the data be kept for 10 years?

Yes

31: Will any adverse events be reported to the University Research Ethics Committee?

Yes

32: Does this research pose any conflicts of interest?

No

33: How will the results of the study be reported and disseminated?

Dissertation/thesis

Signature of Applicant(s)

Name

Date

Signature

Signature by or on behalf of the Head of School

The Committee expects each School to have a pre-screening process for all applications for an ethical opinion on research projects. The purpose of this pre-screening is to ensure that projects are scientifically sound, have been assessed to see if they need ethics approval and, if so, go to the relevant ethics committee. It is not to undertake ethical review itself, which must be undertaken by a formal research ethics committee.

The form must therefore be counter-signed by or on behalf of the Head of School to signify that this pre-screening process has been undertaken

I approve the submission of this application

Name
Date
Signed by or on behalf of the Head of School

APPENDIX B. EXAMPLE OF A USER'S DIARY

NOTE ABOUT THE VIDEO

In this video, first, I check the result of the workflow built in the previous session, since at the time I built the workflow one of the service involved, sesame service, did not work.

DESCRIPTION OF THE WORKFLOW

This workflow reads a file with a list of name of galaxies. Then, it uses the name of the galaxy to build the proper URL to query to HyperLEDA service. This is a web service, and the output is a html file which needs to be parsed in order to extract the value of the property that is searching for.

This workflow has been used in bigger ones as a nested workflow.

PROBLEMS AND REMARKS

- It is not very intuitive the fact that when you give a file as an input, it is the content of the file which is transmitted, and not just the path file. The first attempts to run this workflow got errors, because I though that, after pass the file, I needed to read it to pass the content of the file to the next module of the workflow.
- I usually use the external tool module to include python scripts, mostly for parsing strings or for calculating properties. It is very difficult to write code in the small box inside of the property tool window, so I have to implement the script using my favourite editor (with highlighting tools, etc). I always test the script out of Taverna, running it from console and checking if the results are correct, and then, when I am sure it is correct, I copy and paste the script code to the Taverna tool.
- When the workflow get errors, I would like to see the intermediate output of the modules, so I need to edit the workflow and add output port for all the module outputs that I want to check. It could be very useful if in the result panel, we can inspect the output of the modules without having to add output ports to the workflow

APPENDIX C. DATA CODES

Code name	Description
alternative_service_URL	To have the option to give an alternative service URL address as a fall-over service in case the first web service is down (the given number of retries have all failed).
annotations	Various issues regarding annotations
Beanshell_different_use	How beanshell is used (e.g. build dialogs & GUI components in separate JAR and paste the code to the Beanshell in Taverna)
constant_values	The constant value should be displayed somewhere more easily accessible, on the diagram.
details_panel	Several problems related to the details panel, for example it is only possible to expand one submenu at a time in the Details pane.
error_handling-	Problems participants faced which are associated with error handling
error_handling(suggest)	Suggestions made by users about the error handling
error_handling+	Positive comments regarding error handling
external_script	Python script is added from an external file.
high_level_view(suggest)	A suggestion made by one of the study participants that for scientists, non-technical users easier environment only for running workflows needs to be created.
inputs_window	The problems mentioned about the inputs window, for example if input is not specified w/f still runs but hangs later
list_handling+	Positive comments about list handling
lists_handling-	The problems faced with the list handling issue
loops+	Positive comments about loops in Taverna
memory_allocation	Participant's comment about the memory allocation
myExperiment	Issues related to myExperiment
nested_w/fs(suggest)	Suggestions regarding nested workflows
nested_w/fs+	Positive commentaries about the nested workflows
output_port_names	Different issues about output port names
output_ports	Problems faced with output ports
output_ports_order	Order of the output ports
problemDealing	Different ways of dealing with the problems by participants

Code name	Description
provenance_history	Provenance history related issues
python_shell	The issues mentioned about the python shell
results_track	Problems related to the Resluts tracking
retries	Setting the default number of retries for all the services in the workflow
run_part_of_w/f	The ability to run only part of the workflow
SAMP_functionality	One of the participants asked to have SAMP functionality (button in the result view) integrated in Taverna
script	Various problems related to the script issue
service_list	Issues regarding the service list
service_names	The suggestion made by one of the participants that the string constant (default name) should take the name of the service content
session_memory	The ability to remember certain variables in the entire Taverna session
Updates_&_Plugins	Difficulties related to Updates and Plugins dialog from Advanced Menu in Taverna
user_forum	One of the participants proposed the idea of creating users forum for all users to communicate
w/f_names	Problems regarding workflow names
w/f_sections	Issues related to the work with the workflow sections
warning window+	Positive comments about warning windows
warning_windows-	Some problems faced by participants related to Taverna warning windows.
XML_splitter	A participant mentioned that he would want to click on a particular parameter and add XML splitter rather than adding XML splitter to the whole service.

APPENDIX D. LIST OF IDENTIFIED ISSUES.

1. Annotations

a) “Annotations about services can still only be seen in the BioCatalogue plugin and not on the services themselves. This information is most important when users are linking services together, so it should be visible when users try and do this”.

b) “Annotations from the nested workflows are not propagating to the output workflows annotations”.

c) “Add more fields to annotations: to be able to specify not only authors names, but contributors as well”.

d) “It would be nice to have people's myExperiment IDs automatically inserted in annotations”.

e) “When users upload a workflow to myExperiment, they need to provide some information like the title and the description of the workflow. Usually, this information is in the details of the workflow, and users have to close the myExperiment panel to go to the design panel in order to copy this information and then paste in the myExperiment panel. It would be useful if this information were extracted from the details of the workflow”.

2. Details panel

a) “When selecting an element in the workflow diagram pane, and choosing "Details" (so that the details pane appears for that element), it is only possible to expand one "type" of thing at a time - i.e. "Description" OR "List Handling" OR "Predicted Behaviour" etc. But sometimes users want to look at several of these things at once, or even expand them ALL, it would be nice If user can open and see all submenu simultaneously”.

b) “The description of the services in the “details” panel is not enough. Whenever users add a service to the workflow, they have to check the output of this service by running the whole workflow, in order to know what the output is”.

3. Beanshell different use

a) “Users build dialogs and GUI components in a separate JAR and deploy them in the Taverna Workbench”.

b) “It would be nice to be able to save users’ Beanshells to local services”.

4. External script

a) “Beanshell idea is good, but when building slightly complex components, it is faster to separate them in a JAR (it also allows to build GUI)”.

b) “For now Python script is added from an external file”.

c) “It is difficult to write the code in a small window, as a result code is implemented in another editor and then copy and paste to Tool in Taverna (also in Taverna there is no highlights when editing the code)”.

5. Python shell

a) “Script output file: If users use a tool service template that receives a script as input and if the script implies the creation of some files, they are created in temp folders instead of the workflow folder”.

6. Warning windows

a) “When there is a message “Workflow has warnings but still can be run” it is not clear what is the warning, more information would be helpful”.

b) “It would be good to warn the user with an "alertbox" when datalinks are removed automatically”.

7. Error handling (problems)

a) “The error report does not give sufficient information to figure out the problem”.

8. Error handling (suggestions)

a) “It would be nice when if one iteration fails to extract errors, but still allow w/f to run”.

9. Problem dealing

a) “It would be nice to be able to set the default number of retries for all subservices”.

12. List handling

a) “In Taverna empty lists are created and passed to the nested workflows when single value can be passed instead”.

b) “When users have set iteration strategies or just link things together, the information about the depth of any lists created is available, but much hidden. It would be nice if you could see this on the diagram somehow”.

c) “Always get a list of list, but want single value (need to apply Flatten List to get single value)”.

13. Nested workflows

a) “To be able to save all the workflows up in the chain rather than saving each separately”.

b) “To be able to expand the nested workflow from the context menu (sometimes you want to see nested workflow in context of the bigger workflow)”.

c) “In the design pane, to look at the beanshell input ports, depths and script, users need to open the nested (or each of the nested nested nested) workflow(s), while in the Result Pane, it is possible to select, say, a Beanshell from within a nested (or nested nested

nested) workflow, and look at its input and output result values. it does mean it can take quite a bit of dodging about to get to a particular beanshell description or service description to compare what its designed to do, with what result it gives".

d) "When the workflow gets large it gets difficult to navigate and add components. The window is too small to get a good overview. Maybe workflows with X number of nested workflows or web services just become too complex and the user should be asked to consider stop adding components?".

e) "It would be nice to be able to copy easily a component from a nested workflow to the main workflow".

f) "It would be nice to be able to change the name of nested workflows in a bigger workflow, in a submenu".

g) "When more nested workflows are added, it is becoming difficult to see thing on the screen, need to collapse ports".

h) "Encapsulate part of a workflow as nested workflow does not exist".

15. Run part of the workflow

a) "User has to delete parts of the workflow which he does not want to run now, it would be nice if user can "switch off" parts of the workflow".

16. Workflow sections

a) "It would be convenient if we could copy and paste the entire w/f sections".

17. Output port names

a) "Output port names are too long by default".

b) "It would be nice if space is automatically replaced with underscore when naming ports.

c) "It would be nice to have guidelines on how to name the variables".

18. Output port

a) "Sometimes it becomes difficult to see the workflows, when too many output ports are added".

b) "If there is only 1 output port, you still have to change the view to see all ports before you can link things to it in the diagram".

c) "We cannot inspect the output of the modules without having to add output ports to the workflow".

d) "Additional output port needs to be created in order to see the result (and then deleted later)".

e) "We cannot read anything, need to hide ports".

f) “I would like to have better export options for the output, and to be able to organize my output data in a more interactive and useful way”.

19. Output ports order

a) “Output ports order is not convenient”.

b) “Alphabetical order of the output ports of Biomart is awkward”.

20. Provenance history

a) “To have button to delete provenance history together/using keyboard, not only within Taverna”.

b) “Deleting provenance history one by one is not convenient”.

21. Results track

a) “It is nice in Taverna that we can go to each component and check the output”.

b) “When Results appear, “Values” are not active; user needs to click on “Value” it to be active at the beginning”.

c) “It is useful to check the results of previous runs of the workflow, but sometimes they are not stored. Maybe there is an option in Taverna to keep the results of the workflows, but it is not clear where this option is”.

22. Service list

a) “If you import a service, and then later, its wsdls breaks or is unavailable, Taverna tries to import it on startup and complains slightly. If, after many times failing and you do not use that service anyway, you might want to remove it from the list. But how? It is not on the available services list, because it failed”.

b) “Local services’ description is available only in user’s manual; it would be nice to have the description inside Taverna without going to User’s Manual web-page”.

23. Service names

a) “It would be more convenient if the string constant (default name) take the name of the service content”.

b) “When adding W/F input/output's names replace space with underscore”.

c) “Rename the services in the service list. At the moment users always see the URL to the service, but only a small part of the URL is significant in finding the service”.

d) “It is annoying that cannot use spaces in names”.

24. XML splitter

a) “Click on a particular parameter and add XML splitter rather than add XML splitter to the whole service”.

25. Constant values

a) “When there is a constant value in the workflow, it would be handy if the value itself were displayed somewhere, more easily accessible, on the diagram. e.g. on hover”.

26. High level view (suggest)

a) “For scientists, non-technical users: easier environment only for running workflows”.

1. Inputs window

a) “The inputs window is not disappearing. It is unclear if that means the workflow is running or not (usually it disappears when things work good)”.

b) “If input is not specified w/f still runs but hangs later”.

2. Memory allocation

a) “It would be nice to be able to change memory allocation to Taverna in TAVERNA itself, without editing Taverna shell script”.

3. myExperiment

a) “myExperiment is lovely and useful. Would benefit from being more easily searchable - e.g. when you search for a workflow which you know the name of it does not always appear anywhere in the top of the list, even though you wrote title exactly”.

4. SAMP functionality

a) “It would be great to have SAMP functionality (button in the result view) integrated in Taverna to send a table to TOPCAT/Aladin”.

5. Session memory

a) “To remember certain variables in the entire Taverna session”.

6. Updates & Plugins

a) “On the Updates and Plugins dialog, available from the Advanced menu, it is unclear what the Find Updates button does. Does it find updates for the selected plugin (this would seem most natural) or for all plugins in the list (this would seem most useful). If the latter is a case, then maybe the button should be renamed. If the former, then updating the plugins one-by-one would be tedious, so would be good to have a Find Updates to all plugins button”.

7. Users forum

a) “It would be nice to have user forum for all users to communicate”.

8. Workflow names

a) “Because w/f has the long pathname, the name is not displayed. it would be handy to hover over the pathname and the name of the w/f would be displayed. At the moment there is no place where w/f name is displayed”.