Project Number:	**215219**
Project Acronym:	**SOA4All**
Project Title:	**Service Oriented Architectures for All**
Instrument:	**Integrated Project**
Thematic Priority:	**Information and Communication Technologies**

D2.5.3. Final Summative Evaluation of SOA4All Studio

Activity:	1- Fundamental and Integration Activities
Work Package:	2- Service Deployment and Use
Due Date:	30/04/2011
Submission Date:	29/04/2011
Start Date of Project:	01/03/2008
Duration of Project:	38 Months
Organization Responsible of Deliverable:	UNIMAN
Revision:	1.1
Authors:	Abdallah Namoun (UNIMAN) Ali Owrak (UNIMAN) Usman Wajid (UNIMAN) Martin Carpenter (UNIMAN)
Reviewers:	Yosu Gorroñogoitia (ATOS) Martin Junghans (KIT)

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)

Dissemination Level		
PU	Public	**X**

Version History

Version	Date	Comments, Changes, Status	Authors, contributors, reviewers
0.1	18/4/2011	ToC and Structure outlined	Abdallah Namoun
0.2	19/4/2011	ToC updated and initial ideas	Abdallah Namoun
0.3	20/4/2011	User Study One Procedure and Results	Abdallah Namoun
0.4	23/4/2011	User Study Two Procedure and Results	Abdallah Namoun
0.5	24/4/2011	User feedback toward the SOA4All Tools	Usman Wajid Martin Carpenter
0.6	27/4/2011	Inclusion of screening questionnaire and workbooks	Abdallah Namoun
0.7	27/4/2011	Inclusion of screenshots of SOA4All tools	Ali Owrak Abdallah Namoun
0.8	27/4/2011	Inclusion of participants background	Abdallah Namoun
0.9	27/4/2011	Conclusion and executive summary	Abdallah Namoun Ali Owrak
1.0	28/4/2011	Addressing reviewer comments & minor fixes	Abdallah Namoun
1.1	29/4/2011	Final corrections	Nikolay Mehandjiev

Table of Contents

LIST OF FIGURES **4**

LIST OF TABLES **5**

GLOSSARY OF ACRONYMS **6**

EXECUTIVE SUMMARY **7**

1. INTRODUCTION **8**

1.1 INTRODUCTORY EXPLANATION OF THE DELIVERABLE 8
1.2 PURPOSE AND SCOPE 8
1.3 STRUCTURE OF THE DOCUMENT 8
1.4 METHODOLOGY 9

2. USABILITY EVALUATION **10**

2.1 USER TESTING ONE 15
2.1.1 Participants' Background 15
2.1.2 Evaluation Procedure and Test Scenario 17
2.1.3 Analysis method 19
2.1.4 Usability Results from the First User Testing 20
2.2 USER TESTING TWO 29
2.2.1 Participants' Background 29
2.2.2 Evaluation Procedure 29
2.2.3 Analysis method 31
2.2.4 Usability Results from the Second User Testing 31

3. CONCLUSION **35**

4. REFERENCES **37**

ANNEX A – EVALUATION WORKBOOK 1 **38**

ANNEX B – EVALUATION WORKBOOK 2 **50**

ANNEX C – SCREENING QUESTIONNAIRE **58**

List of Figures

Figure 1. WSMO-Lite Editor 12
Figure 2. Consumption Platform 12
Figure 3. Discovery Module 13
Figure 4. Process Editor 13
Figure 5. Monitoring Module 14
Figure 6. User Assisted Composition Tool 14
Figure 7. Programmers and Non-programmers' Development Experience 17
Figure 8. Programmers and Non-programmers' Interest and Likelihood in Uptaking Software Development 17
Figure 9. Usability Rating of Consumption Platform 21
Figure 10. Usability Rating of Consumption Platform by Programmers and Non-programmers 22
Figure 11. Usability Rating of User Assisted Composition Tool 23
Figure 12. Usability Rating of UAC Tool by Programmers and Non-programmers....................23
Figure 13. Usability Rating of SOA4All Process Editor 24
Figure 14. Usability Rating of SOA4All Process Editor by Programmers and Non-programmers 25
Figure 15. Usability Rating of Monitoring Module....................26
Figure 16. Usability Rating of Monitoring Platform by Programmers and Non-programmers 27
Figure 17. Usability Rating of SOA4All Studio 28
Figure 18. Usability Rating of SOA4All Studio by Programmers and Non-programmers....................28
Figure 19. Usability Rating of Consumption Platform....................32
Figure 20. Usability Rating of WSMO-Lite Editor 33
Figure 21. Usability Rating of Discovery Module....................33
Figure 22. Usability Rating of SOA4All Studio and its Tools 35

List of Tables

Table 1. Summary of the WP2 Summative Usability Studies 11

Table 2. Average completion time for service consumption, assisted modelling, manual modelling, and service monitoring..20

Table 3. Average usability scores of the Process Editor 24

Table 4. Average completion time for service consumption, annotation, and discovery 31

Glossary of Acronyms

Acronym	Definition
D	*Deliverable*
UAC	*User Assisted Composition Tool (a view on Process Editor)*
SPICES	*Semantic Platform for the Interaction and Consumption of Enriched Services*
WSDL	*Web Services Description Language*
PE	*Process Editor*
WSMO-Lite	*Web Services Modelling Ontology Lite*
SPSS	*Statistical Package for Social Sciences*
T-test	*Statistical hypothesis test*
P	*Significance value*
STD	*Standard Deviation*
M	*Mean*

Executive Summary

Deliverable D2.5.3 is a continuation of the usability evaluation efforts carried out in earlier studies and focus groups whose results were detailed in D2.5.1 [3] and D2.5.2 [4]. This deliverable summarises the procedure and main findings of the final summative evaluation of the SOA4All Studio assessing user acceptance and ease of use of the various tools and features within the Studio. Unlike previous formative tests which aimed to inspect user interfaces and identify usability problems, the studies undertaken as part of this final summative test focused on collecting users' attitudes and opinions towards the concepts implemented within the Studio and rating of different features. Two separate usability studies, involving a total of 34 users, were conducted to evaluate service consumption, discovery, annotation, assisted composition and full views of process modelling, and monitoring using the SOA4All Studio.

The summative tests outlined here involved representative end users fulfilling the user-centred design [12] and assessed how well the SOA4All Studio satisfies the usability objectives of the SOA4All project. The aim of the first study was to determine the applicability of the SOA4All Studio to different user groups, and involved the participation of 12 programmers and 12 non-programmers. Following a narrative presentation of the SOA4All Studio the evaluation focused on assessing user development experience when interacting with the consumption, process modelling and monitoring platforms. The first evaluation results revealed that both user groups considered interaction with the Studio intuitive, easy and promising. Moreover, participants felt that following further interaction and training they would be comfortable in using the Studio on a regular basis. The second study, involving 7 programmers and 3 non programmers, aimed to determine the usability of the consumption, WSMO-Lite and Discovery Module. The sequence of the user centred tasks was purposely devised to demonstrate the relevance of service annotation. The study demonstrated that although participants found the consumption module easy to use, they appreciated more the comprehensive search facility offered through the Discovery Module. Furthermore, participants recognised the relevance and importance of annotating services using the WSMO-Lite editor, thus demonstrating the value and feasibility of semantics for the future Internet of Services.

1. Introduction

1.1 Introductory explanation of the deliverable

This deliverable primarily reports on and concludes the final evaluation efforts conducted within Work Package 2. The evaluation studies undertaken focus on the usability aspects of the SOA4All Studio within the context of the WP2. The general aim of these evaluation exercises is to ascertain user interaction with the SOA4All Studio and its constituent tools before finally releasing for public consumption (and / or further business and technical development).

The deliverable details the methodology we adopted to evaluate the final outcome of the SOA4All project and the results gained from these usability evaluations. The evaluation study began with a screening questionnaire which aimed at categorising participants into technical and non-technical groups. Two summative usability tests are described within this deliverable. The first test focused on the evaluation of consumption, modelling and monitoring tools. 24 participants (12 technical and 12 non-technical) were invited to a training presentation outlining the concepts and features of the SOA4All Studio. Following initial training, users undertook interactive participation with the Studio carrying out tasks devised to test the system. The second test focused on the evaluation of 10 participants (7 of which are technical) with the aim of determining the usability of the consumption, annotation and discovery tools through a number of key tasks. Following an initial training presentation, participants were invited to interact with the tools outlined through a number of user centred tasks purposely devised to demonstrate the relevance of service annotation and the importance of service discovery.

Furthermore, this deliverable sums up the results of user rating of various usability dimensions of the SOA4All tools and summarises their comments and feedback.

1.2 Purpose and Scope

The purpose of this document is to detail the results of the final usability evaluation we conducted within the context of WP2. These evaluations aimed to gauge user acceptance of the SOA4All Studio and validate its suitability to the representative end users.

This document

- recaps the objectives of the SOA4All Studio and its constituent tools
- describes the evaluation procedures, data collection and analysis methods and rationale behind them
- presents the usability results of two user studies covering differing phases of service development covering annotation, discovery, consumption, assisted composition, modelling, and monitoring; the results are presented in the form of average usability scores and user comments about the designated SOA4All tools

1.3 Structure of the document

The document is organised into 3 major sections. Section 2.1 describes the evaluation procedure of the first user study and presents its usability results whereas Section 2.2 describes the evaluation procedure of the second user study and presents its usability results. Both sections focus on how well the objectives of the SOA4All Studio were satisfied in addition to determining the user acceptance of the Studios design features. Finally, Section 3 concludes the deliverable with an overall comparison of user satisfaction toward the SOA4All tools.

1.4 Methodology

The approaches behind our usability evaluation studies are aligned with the overall evaluation strategy of SOA4All described in the Description of Work and also in the D2.5.1 deliverable [3]. As it is the objective of this deliverable to report on final usability results of the SOA4All project, we concentrated on measuring user experience and satisfaction when developing software services using the developed tools. We undertook summative tests rather than formative tests as they are often conducted at the end of the software development lifecycle to gauge users' acceptance and satisfaction with the SOA4All tools and predict future adoption and use by representative users. Further details of the evaluation procedure and methods used are given in the respective sections below.

2. Usability Evaluation

Usability evaluation is carried out to test the ease of use with which users perform different tasks using a particular interactive system and how satisfied end users are with the different features and options of the system [10]. Usability tests are of two types: formative and summative [10]. Formative tests are usually conducted in the early stages of the design of a system and they aim to inspect usability problems and provide relevant design feedback and recommendations to the developers. Summative tests, however, are conducted at the very end of software development process and employ a ready-to-use system to validate the main concepts and features developed in the SOA4All project. The latter tests focus on collecting user performance data (e.g. time users spent completing tasks and correctness percentage of their solutions), satisfaction rating (e.g. subjective rating of usability dimensions such as ease of use, ease of learning, ease of memorability, *etc*…), and user acceptance and attitudes (e.g. comments and verbal feedback) toward the underlying concepts within the differing tools of the Studio.

The current deliverable reports on usability results from two major user studies including a total of 34 participants as follows:

- ***User testing one***: this evaluation lasted for 1.5 hours, included 24 participants and assessed Consumption Platform, Process Editor, and Monitoring Module of the SOA4All Studio and the User Assisted Composition Tool (or ‘UAC’ for short). This User Assisted Composition Tool is a specialised view onto the Process Editor, which employs a wizard-like interface to assist binding of services to template tasks. This was developed by the University of Manchester to specifically target non-programmers’ expected levels of skills and time constraints when undertaking service composition. It allows them to compose services simply by customising goal-based templates and without the need to model services or write programming code [2]. Both technical and non-technical participants interacted with these tools to implement a process model which enables students to register in a UK University. The set of development tasks comprised: finding, executing, and adding services to list of favourites using the Consumption Platform; modelling the ‘University Registration’ process by (1) selecting services, removing and arranging activities in the student registration template as necessary using the UAC and (2) specifying conditions and parallel activities and defining dataflow between existing services using the Process Editor; and observing and analysing the behaviour of existing services using the Monitoring Module.
- ***User testing two***: this evaluation lasted for 1 hours, included 10 participants and assessed Consumption Platform, Discovery Module, and WSMO-Lite Editor. The tasks of this user study comprised: finding services using simple text matching and adding them to list of favourites, annotating a SchoolContact service with semantic descriptions, finding the annotated services using their parameters (e.g. input and output) and particular ontologies.

In both user testing, we employed the think-aloud protocol [1] to gather user opinion and questionnaires to capture their satisfaction in the form of usability scores towards the designated tools. We have also recorded users’ interaction behaviour and their verbal reports using SnagIt screen-capturing program [7] for follow up analysis. The objectives of the WP2 evaluations are as follows:

- collect user acceptance of the general SOA4All design features in the form of opinion data and usability scores; and
- evaluate how well the objectives of the SOA4All Tools (thus the Studio and the UAC) have been satisfied.

Table 1. Summary of the WP2 Summative Usability Studies

	User study one	**User study two**
Number of participants	24	10
Type of participants	12 technical users (i.e. Programmers) and 12 non-technical users (i.e. Non-programmers)	7 technical users (i.e. Programmers) and 3 non technical users (i.e. Non-programmers)
SOA4All tools evaluated	Consumption Platform Process Editor Monitoring Module User Assisted Composition Tool	Consumption Platform WSMO-Lite Editor Discovery Module
Length of user study	1 hour and a half	1 hour
Length of training session	1 hour	1 hour

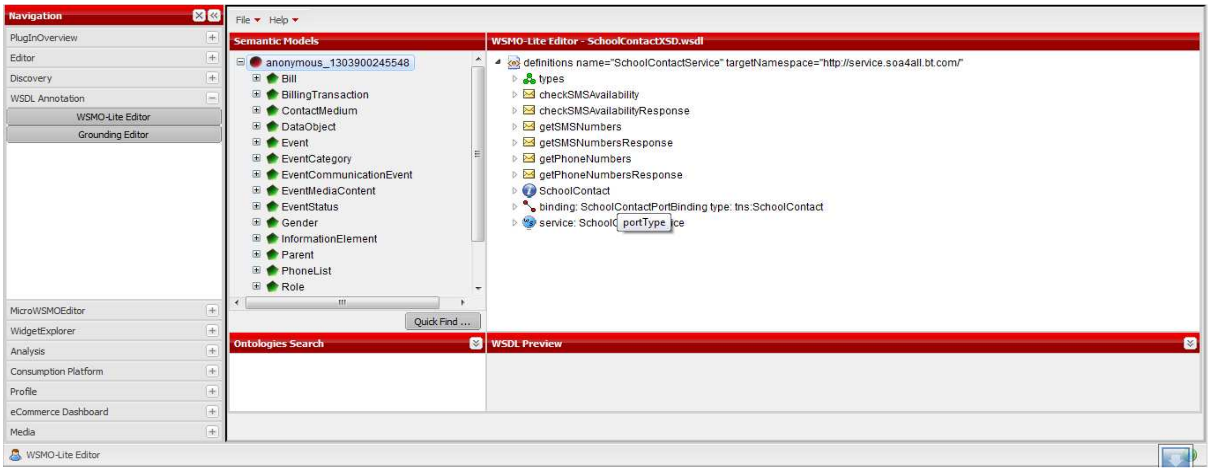
The first user study concentrated on evaluating the usability of the Consumption Platform, Process Editor, the User Assisted Composition Tool and Monitoring Module of the SOA4All Studio. However, the second user study concentrated on evaluating the usability of the Consumption Platform, WSMO-Lite Editor, and Discovery Module.

To evaluate the coverage of the SOA4All objectives, we start by reaffirming the objectives of the SOA4All Studio and its constituent tools: WSMO-Lite Editor, Consumption Platform, Discovery Module, Process Editor, Monitoring Module, and User Assisted Composition Tool. The Studio was developed with the principal aim of empowering end users, including non-programmers, to build interactive service-based applications that fulfil their personal or professional needs without the necessity to write mundane programming code. In regard to the tools of SOA4All Studio the goals for each tool are listed as follows:

WSMO-Lite Editor: aims to enable end users to:

- Objective 1: Add semantic descriptions, in the form of ontological concepts, to WSDL-based descriptions of web services [8] to allow for dynamic invocation and discovery of the resulting semantic web services
- Objective 2: Explore differing service elements, such as data types and operations and their parameters
- Objective 3: Publish semantic service annotations in an online repository (i.e. iServe [9]) which can be later retrieved and used in the subsequent steps in the service development lifecycle (i.e. Process modelling)

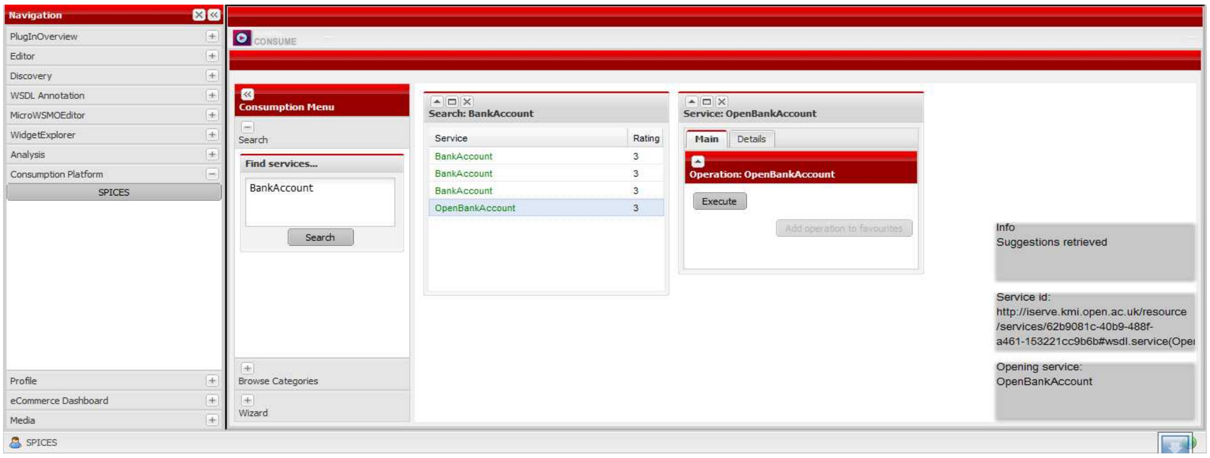
Figure 1. WSMO-Lite Editor



Consumption Platform aims to enable end users to:

- Objective 1: Retrieve and find target services shared through SOA4All using a simple and easy to use service search engine
- Objective 2: Explore the different operations and their corresponding details, and view user comments and ratings of the available services
- Objective 4: Execute and inspect the behaviour of the available services
- Objective 3: Add the desired services to the list of favourites for the follow-up stages

Figure 2. Consumption Platform



Service	Rating
BankAccount	3
BankAccount	3
BankAccount	3
OpenBankAccount	3

Info
Suggestions retrieved

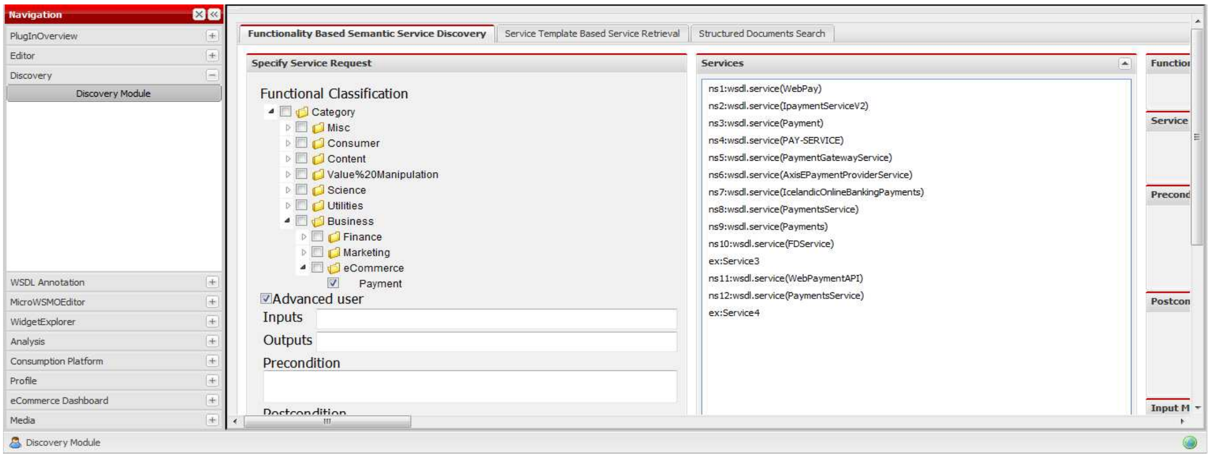
Service id:
http://iserve.kmi.open.ac.uk/resource/services/62b9081c-40b9-488f-a461-153221cc9b6b#wsdl.service(Ope

Opening service:
OpenBankAccount

Discovery Module aims to enable end users to:

- Objective 1: Retrieve and find appropriate/desired services shared through SOA4All using an ontology-based search engine; users specify the parameters (input and output), preconditions and post-conditions to find annotated services. They can also select classes from the classification, as well as specify non-functional requests
- Objective 2: Add the annotated services to the list of favourites for the proceeding stages, like the creation of service compositions within the Process Editor

Figure 3. Discovery Module

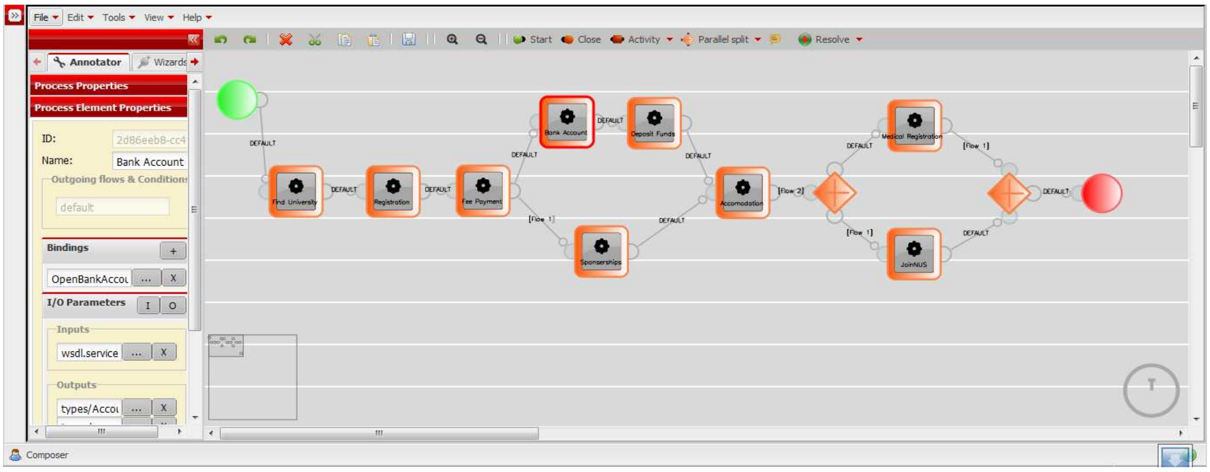


ns1:wsdl.service(WebPay)
ns2:wsdl.service(IpaymentServiceV2)
ns3:wsdl.service(Payment)
ns4:wsdl.service(PAY-SERVICE)
ns5:wsdl.service(PaymentGatewayService)
ns6:wsdl.service(AxisEPaymentProviderService)
ns7:wsdl.service(IcelandicOnlineBankingPayments)
ns8:wsdl.service(PaymentsService)
ns9:wsdl.service(Payments)
ns10:wsdl.service(FDService)
ex:Service3
ns11:wsdl.service(WebPaymentAPI)
ns12:wsdl.service(PaymentsService)
ex:Service4

Process Editor aims to enable end users to:

- Objective 1: Model a process visually without the need to write programming code or deal with low level technical details in pursuit of a their needs/ goals; visual notations in the form of activities, splits, merges, start and end notations were made available to help users design their processes
- Objective 2: Use the desired services from their list of favourites and bind them to the appropriate activities of the process model
- Objective 3: Specify dataflow between different services via a dataflow editor

Figure 4. Process Editor

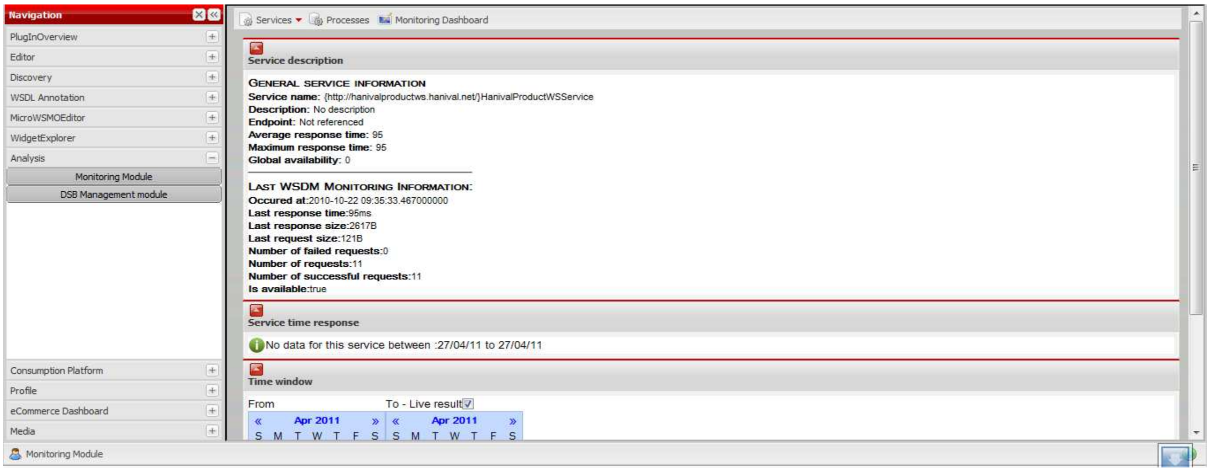


ID: 2d86eeb8-cc4
Name: Bank Account
default
OpenBankAccou
wsdl.service
types/Accou

Monitoring Module aims to enable end users to:

- Objective 1: Browse their services and processes and watch/analyse their behaviour over time; this platform acts as a summary provider for the performance (e.g. average response time and number of successful request) of the deployed services

Figure 5. Monitoring Module



GENERAL SERVICE INFORMATION
Service name: {http://hanivalproductws.hanival.net/}HanivalProductWSService
Description: No description
Endpoint: Not referenced
Average response time: 95
Maximum response time: 95
Global availability: 0

LAST WSDM MONITORING INFORMATION:
Occured at:2010-10-22 09:35:33.467000000
Last response time:95ms
Last response size:2617B
Last request size:121B
Number of failed requests:0
Number of requests:11
Number of successful requests:11
Is available:true

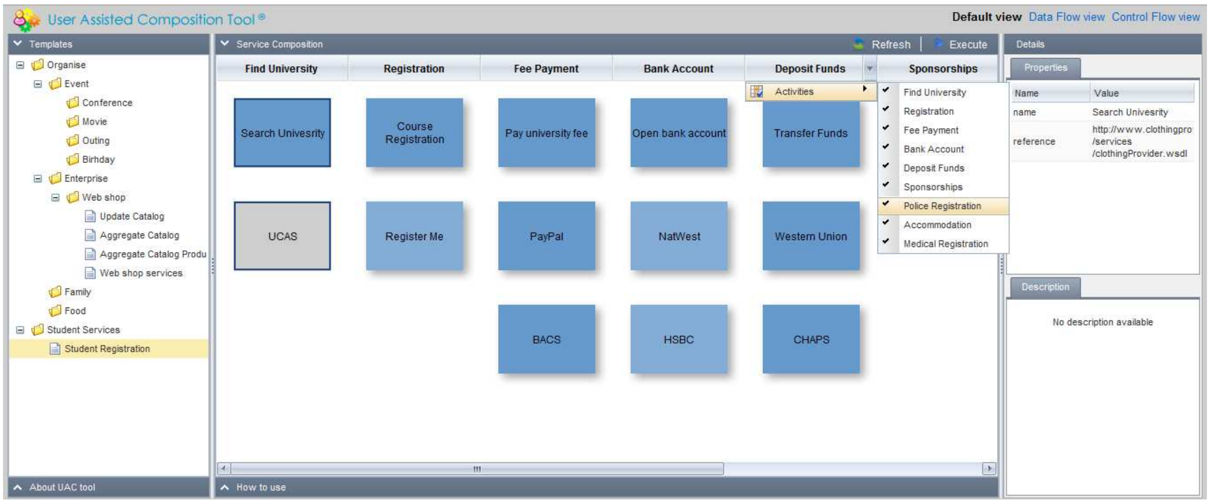
No data for this service between :27/04/11 to 27/04/11

From
To - Live result ☑

User Assisted Composition Tool is a specialised view onto the Process Editor, a wizard-like interface which hides service connections and allows users to bind services to template tasks in a simple and time-efficient manner. This was developed at the University of Manchester after earlier formative evaluation studies revealed certain limitations of non-programmers in terms of both available time for learning and doing service composition, but also in terms of their mental models of inter-service dependencies. It aims to to specifically empower non-programmers to compose services simply by customising goal-based templates and without the need to model processes or write programming code. It aims to enable end users to:

- Objective 1: Compose services by instantiating and customising existing service templates (left menu in Figure 6). The instantiation of existing templates involves selecting appropriate services to bind to process tasks, and customising was limited to removing non-core tasks (activities), represented as columns in Figure 6, from a template according to user needs without visually modelling the process.
- Objective 2: Facilitate the modelling process and make it more accessible for non-programmers; the goal here is to achieve a composition without the need to learn or deal with the boring low-level technical details

Figure 6. User Assisted Composition Tool



✓ Find University
✓ Registration
✓ Fee Payment
✓ Bank Account
✓ Deposit Funds
✓ Sponsorships
✓ Police Registration
✓ Accommodation
✓ Medical Registration

Name	Value
name	Search Univesrity
reference	http://www.clothingpro /services /clothingProvider.wsdl

No description available

2.1 User Testing One

2.1.1 Participants' Background

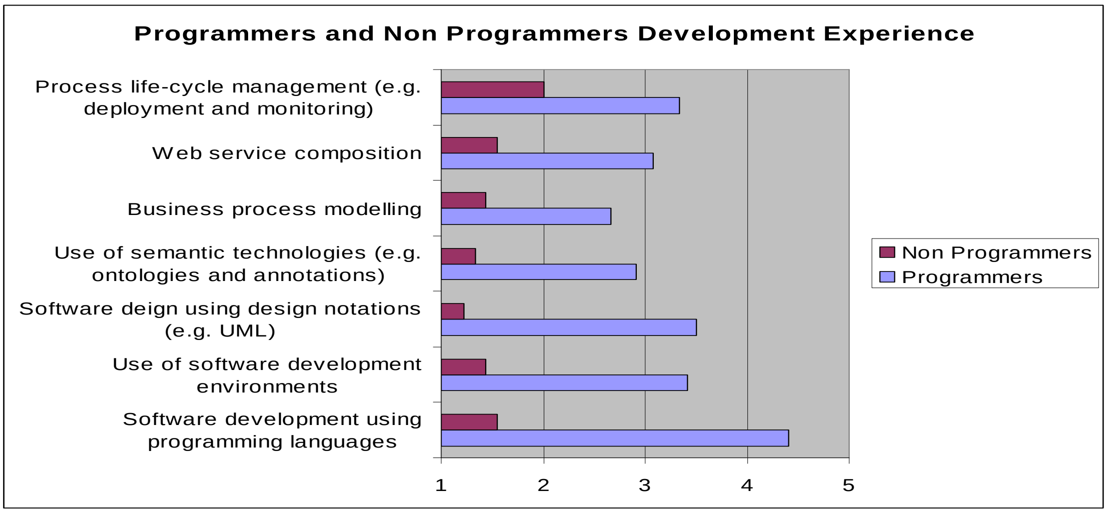
A total of 24 users took part in the first evaluation of SOA4All tools. Of those, 12 participants coming from Computer Science disciplines had extensive programming and development experience using a variety of programming languages such as Java, C, and C++ and software development environments such as Visual Studio .NET, NetBeans and Eclipse, whereas the remaining 12 participants had no previous programming / development experience and come from other Non-Computer Science disciplines such as: Business and Management, International Human Resource Management, Marketing, and Managerial Psychology. A carefully-administered screening questionnaire was distributed to the University staff and student mailing lists prior to selecting and inviting suitable candidates for the purpose of our evaluation. The questionnaire (in Annex C) focused on measuring participants' programming/development experience, modelling experience, background knowledge of modelling tools and semantic technologies, and general demographic information. Since SOA4All project endeavours to cover a wide range of users with different skills and characteristics, we ensured that our participating users have different profiles and thus we used their technical background and programming skills to divide them into two main groups: programmers and non-programmers. This division has the advantage of:

1. Testing the difference between user perceptions of the usability of the SOA4All tools
2. Ascertaining whether the SOA4All tools are suitable for different types of end users, particularly programmers and non-programmers

Participants scored their software development experience on 5-point Likert scale [11] where 1 signifies '*Extremely poor'*, 5 signifies '*Excellent'*, and 3 signifies '*Average'*. We collected and averaged Programmers and Non-programmers' scores to different software development questions as shown in Figure 7. These scores were then entered into SPSS, a statistical analysis software [5]. We have chosen and run several Independent Samples T-tests [6], which are statistical tests to compare the means of two independent samples, as our test samples (i.e. Programmers Vs Non-programmers) were separate and contained different individual users. If the results of a T-test are significant we say that the means of our two samples are significantly different. Results showed that Programmers' development skills (in Figure 7) differed significantly from Non-programmers' in all measures, (T-tests, P<0.01). For instance, Programmers have a stronger experience in developing software using programming languages (mean= 4.42, standard deviation= 0.51), in using software development environments (m= 3.42, std= 1.16), and designing software applications using design notations (m= 3.5, std= 1.00) than Non-programmers [(m= 1.56, std= 1.13), (m= 1.44, std= 0.88), and (m=1.22, std= 0.66) respectively] as shown in

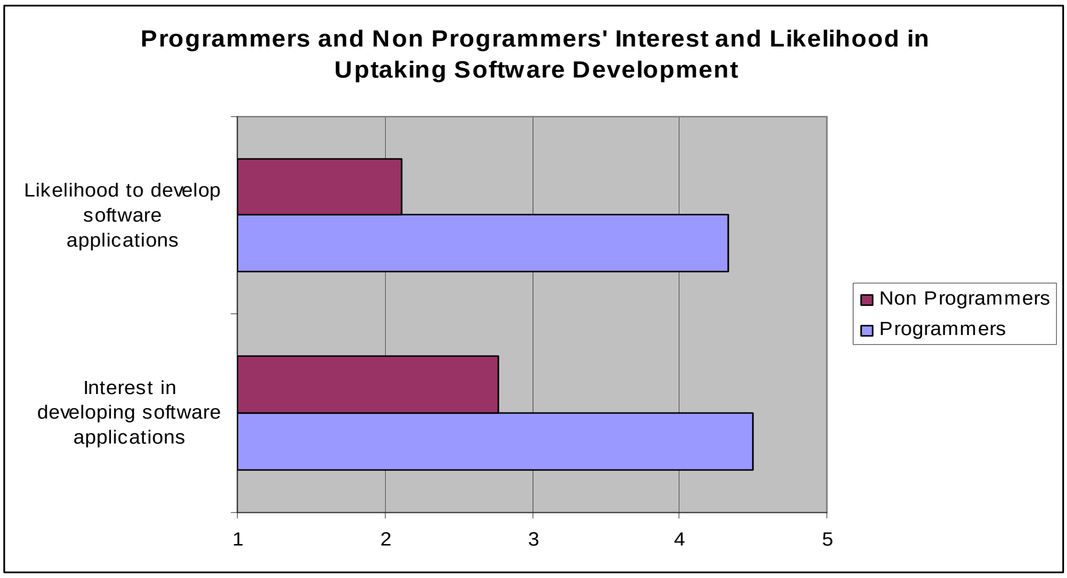
Figure 7.

Figure 7. Programmers and Non-programmers' Development Experience



We also asked our participants to indicate their interest and likelihood of developing software applications as shown in Figure 8. For both questions, Programmers' interest (m= 4.5) and likelihood (m= 4.33) to develop application were rated significantly higher than Non-programmers' (m= 2.77 and m= 2.11 respectively), (T-tests, P<0.001).

Figure 8. Programmers and Non-programmers' Interest and Likelihood in Uptaking Software Development



2.1.2 Evaluation Procedure and Test Scenario

This summative testing, aiming to gather users' attitudes and acceptance of the SOA4All Studio and its tools, consisted of three main phases:

(a) training phase,
(b) development phase, and
(c) rating phase.

A detailed description of each phase is presented below.

2.1.2.1 Training Phase

During the training phase, all representative users attended a 1-hour group presentation which briefly explained the purpose of the SOA4All project and thoroughly demonstrated the various features and aspects of each of the designated SOA4All tools. This included a tutorial demonstrating one service development example of moderate complexity (i.e. contacting friends for an outing). At the end of the presentation, participants were encouraged to comment and ask questions about what they have seen. In summary, the training phase encompassed the subsequent steps:

1. A group presentation and training providing detailed explanation on how to use the find services using the Consumption Platform, model a particular process using the User Assisted Composition Tool and Process Editor, and monitor the progress of their services using the Monitoring Module;
2. A post-training session for participants to ask elaborating questions about the presented tools;

The training phase included 4 training sessions with 6 to 7 participants in each training session. All training sessions were similar in regard to the length and content and given by the same presenter to ensure all participants receive the same treatment. Each participant was allowed to attend only one training session depending on their availability. Following the training sessions, each participant was taken to an individual room where they completed a set of development tasks independently as explained in the next section.

2.1.2.2 Development Phase

At the beginning of this session users were given a workbook containing the University Registration scenario along with the tasks which they were instructed to complete covering three aspects of the service development lifecycle: consumption, modelling, and monitoring of services. In this usability testing we opted the think-aloud protocol [1] through which participants verbalised their thoughts during the development activities to unravel their design strategies and obtain their opinion as they undertake the tasks. We recorded user interaction behaviour for each tool and their vocal discourse using a screen capturing software (i.e. Snag it) for post-experiment analysis. The precise steps undertaken by each participant in the development phase are as follows:

- **Pre-test Interview:** participants discussed their existing experience and opinions about Software and Service Development using different software development environments; this pre-test interview lasted approximately 3-5 minutes.
- **Interaction with the SOA4All Studio and its tools:** participants interacted with the different tools of the “SOA4All Studio” and completed a set of diverse development tasks (consumption, modelling using the UAC tool and Process Editor, and monitoring) in order to fulfil the test scenario reported in Section 2.1.2.2.1.

2.1.2.2.1 Scenario and Task Descriptions

Prior to conducting the usability testing with our end users, we conducted a few pilot studies to ensure that the development tasks are well described and easily understood to ensure they can be performed and interpreted in the right and intended way. Since it is quite important to strike the right balance between clarity and specificity of the tasks, we refined the descriptions of the tasks so that they accurately describe what needs to be achieved by our audience without detailing the fine-grained steps required to solve the development tasks. We also created development tasks that are complex enough to enable full interaction with at least the core functionalities and components of the Studio. The tasks included finding and executing services, modelling a student registration process, and monitoring the performance of existing services. In the Consumption Platform, participants had to search for three relevant services and add their operations to the list of favourites. The modelling tasks focused on modelling the process of registering a student in a UK University using both the

UAC and Process Editor. In the UAC, participants were requested to navigate to and load the appropriate template, whose name was given to them, and customise it according to the test scenario. However, in the Process Editor they were requested to define a condition and parallel activities, bind services to the right activities in the process model, and define dataflow between two services. Finally using the Monitoring Module, users were instructed to find relevant services and report information on specific behaviour such as average response time. Refer to the Annex A for the list of tasks and their description.

We chose a scenario with which our participants are familiar since they have to undergo this process when applying to study in a particular university in the UK. The overall description of the test scenario is as follows: “This scenario describes the registration process that overseas students go through while getting admission in UK universities.

Your goal is to complete an overseas student registration process. For this you need to develop a software application which allows you to search for a UK university, register for a course in the university and find an accommodation. There are two ways for paying the university fee, the first way is to open a bank account and get funds transferred into that account. The bank account can be used to make payment for the university fee. In the second way you can request a letter from a sponsor and submit that letter to the university. You must choose only one way to pay the university fee. After paying the university fee you will register with the NHS”.

2.1.2.3 Rating phase

The rating phase consisted of two parts: rating of the constituent tools and rating of the whole Studio. Following each development phase (i.e. consumption, modelling, monitoring), participated were asked to rate a number of usability dimensions covering various features of the respective tool (i.e. Consumption Platform, User Assisted Composition, Process Editor, and Monitoring Module) by completing a questionnaire. They were also asked to comment on their experience and indicate their opinions about the tools in a post-test interview.

Finally, at the end of the usability testing participants evaluated the SOA4All Studio as a whole by:

- Rating their overall service development experience, overall usability of and satisfaction toward the SOA4All Studio. Participants completed a paper-based questionnaire containing question items about usability and preferences such as “ease of use … etc” to which they expressed their degree of agreement on a 1 to 5 rating scale, where (1 =strongly disagree and 5 =strongly agree). Quantitatively, we sought to demonstrate that the SOA4All tools are easy to learn and use to use by both Programmers and Non-programmers. A satisfactory average rating score is ‘4’ and above, where 3 is the neutral value and 5 is the maximum value in our usability scale.
- Stating their final opinion about the SOA4All Studio in a de-briefing interview highlighting the positive and negative aspects as well as suggestions for future improvements. Qualitatively, we sought to demonstrate that our users are able to perform the tasks and report on a positive user experience and appreciation of the SOA4All tools in the form of positive statements.

The WP2 evaluation workbook covering the details of aforementioned phases (i.e. development and rating phases) of user study one is included in Annex A.

2.1.3 Analysis method

User opinion data were collected through means of think-aloud protocol, interviews and questionnaires where participants commented on the core functionalities of the Studio and rated different usability dimensions, such as ease of learning and ease of use, on 1-5 Likert rating scale. Self-reported data were initially recorded by both the participants who filled out some questions in the workbook and the moderators (i.e. experimenters) who took notes as the experiments unfolded. Researchers walked through the interaction videos and calculated the completion time for each task per user. Quantitative and qualitative data was then transcribed into spreadsheets for subsequent analysis.

Since this is a summative evaluation, the researchers primarily focused on analysing the performance data (e.g. completion time) and usability scores as they reflect users' satisfaction and attitudes toward the implemented tools and concepts of SOA4All project. However, for each section of the results we include a summary of users' qualitative opinion.

2.1.4 Usability Results from the First User Testing

The results from this study are divided into two sections: user performance covering average completion time for each task type (i.e. consumption, goal-based template customisation, manual modelling, and monitoring) and tools' rating covering average usability scores of different features of the tools and user feedback covering positive and negative aspects of the tools as well as ways for improving them. Furthermore, for each tool we discuss user acceptance and attitudes and how the objectives were fulfilled.

2.1.4.1 User Performance

Task completion time was measured by recording the time users started executing each task and the time they completed that particular task. Average completion time for each task type per participant was calculated and is reported in column 3 of Table 2. The results show that manual modelling of the university registration process took the longest time to perform (mean= 20.33 minutes), followed by finding and executing services using the Consumption Platform, customising the goal-based template using the UAC tool, and finally monitoring of services with an average completion time of 3.5 minutes. We have also calculated and compared the average completion time for the Programmers and Non-programmers.

Table 2. Average completion time for service consumption, assisted modelling, manual modelling, and service monitoring

SOA4All Tool	Development task	Average completion time (minutes)	Standard deviation	Programmers' average completion time	Non-programmers' average completion time
Consumption Platform	**Finding and Executing Services**	11.05	3.77	10	12
User Assisted Composition Tool	**Goal-based Template Customisation**	9.89	3.01	10.44	9.4
Process Editor	**Manual Modelling**	20.33	3.49	19.55	21.11
Monitoring Module	**Monitoring and observing the behaviour of services**	3.5	1.79	3	3.9

Performance results were very close to each other and no clear differences were spotted as reported in columns 5 and 6 of Table 2, demonstrating the suitability of these tools for both groups of end users. It is worthwhile to indicate that the tasks differed in complexity and number of steps required to complete them. Therefore this table is not intended to compare the completion time of the different tasks but rather to give an overall picture of how development time was distributed across the tasks. This might be used as a reference for indicating the complexity of process modelling, service consumption and monitoring.

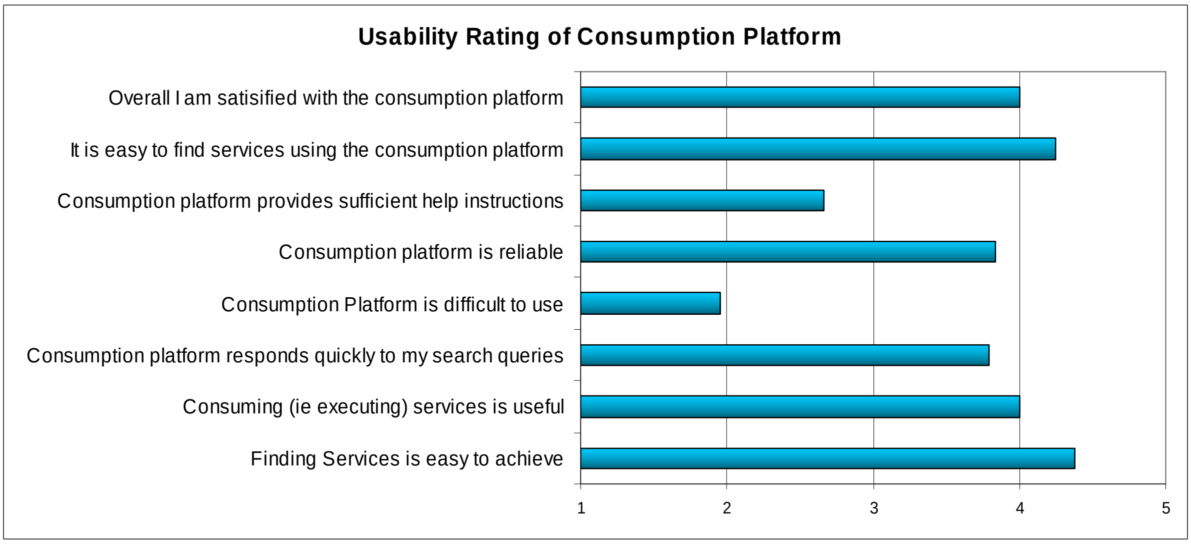
2.1.4.2 Usability Rating and User Feedback toward the SOA4All Tools (Consumption Platform, User Assisted Composition Tool, Process Editor, and Monitoring Module)

This section summarises and presents average usability scores of usability dimensions rated on a 5-point Likert scale, where 1 corresponds to *Strongly Disagree* and 5 corresponds to *Strongly Agree,* and user opinion about the components and functionality of the tools: Consumption Platform, User Assisted Composition Tool, Process Editor, and Monitoring Module. Furthermore, this section compares and contrasts the experience of our two end user groups using the SOA4All tools: Programmers and Non-programmers.

2.1.4.2.1 Consumption Platform

Average usability scores for different dimensions revealed that participants are generally satisfied (mean=4.0, standard deviation= 0.59) with the Consumption Platform. This is attributed to the ease with which services can be found and retrieved using the search engine of the Consumption Platform (m= 4.25, std= 0.67), to its usefulness (m= 4.0, std= 0.78) and reliability (m= 3.83, std= 0.91). Participants disagreed that the Consumption Platform is difficult to use (m= 1.95, std= 1.04). However participants were quite undecided whether the platform provides sufficient amount of help and guidance (m= 2.66, std= 1.00). Rating of other usability questions is depicted in Figure 9.

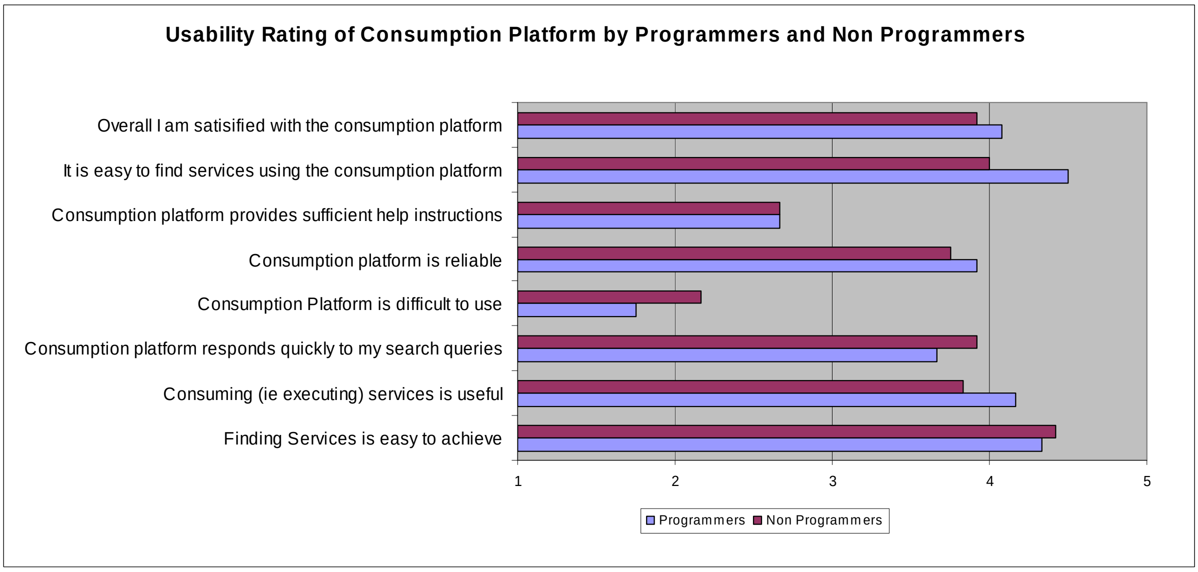
Figure 9. Usability Rating of Consumption Platform



Programmers and Non-programmers' rating scores of the Consumption Platform were similar for all measures and did not differ statistically which might be attributed to their familiarity with similar software and web applications. Both groups of users rated the tool as easy to learn and not difficult to use, as shown in Figure 10.

Qualitatively consumption was perceived a straightforward and easy to do task. Users felt comfortable while performing their task in the Consumption Platform. The layout of the Consumption Platform was appreciated by the majority of our users. In this respect, users liked the choice of colours and the proportion of different windows and elements. Owing to the simplistic layout of Consumption Platform users found it easy to search for services and add service operations to their favourites. Users liked the pop-up messages as they gave them good indication of what is going on. Moreover, some users liked the way search results can be opened in different windows; they found it easy to move around windows in order to do multi-tasking.

Figure 10. Usability Rating of Consumption Platform by Programmers and Non-programmers



On the other hand, some users found the original size of search result window(s) too small as it required scrolling to explore the contents. However, this problem can be rectified by maximizing the window(s). Moreover, to some users the scale of the service ratings (i.e. 1-5) was not clear until explained by the moderator. Apart from the above, our users liked the service rankings/ratings feature and found them helpful in selecting an appropriate service.

User suggestions about further improvements in the Consumption Platform include the use of stars (1-5) rather than numbers, as starts are widely used in popular websites. Also, a user suggested using percentage of satisfaction for ranking the services. Other suggestions include introducing the ability to trigger search by pressing the 'Enter' button on the keyboard and adjusting the size of search results window(s) so that user can see the full content without the need of scrolling or maximizing the window.

2.1.4.2.2 User Assisted Composition Tool

Our end users rated the User Assisted Composition Tool as easy to use (m= 4.08, std= 0.77) and easy to learn (m= 4.26, std= 0.81). They disagreed that the UAC tool is difficult to navigate (m=2.16, std= 1.09) or that it requires support from technical people (m= 2.54, std= 1.21). These scores assert and satisfy our above-listed objectives that the UAC can easily be learnt and used by Non-programmers without the need to master the underlying technical details of service composition and process modelling. Users expressed strong willingness to use the UAC tool more frequently in the future (m= 3.83, std= 1.00) and overall were satisfied with it (m= 3.79, std= 0.72) as shown in *Figure 11*.

We have also calculated average user rating of the UAC Tool by Programmers and Non-programmers and results showed, Figure 12, that very similar scores were assigned to all usability dimensions as no apparent statistical differences were found. Programmers disagreed that using the tool needs to be supported by technical people whereas Non-programmers were undecided about this question and were not convinced that enough help and documentation is made available to assist them in accomplishing the tasks. This finding is not surprising if we take into consideration the background of the two groups and the level of technical details they deal with on a daily basis; Non technical users usually require help and support from the software systems they use to perform their jobs. Both groups of end users were equally satisfied with the UAC Tool.

Figure 11. Usability Rating of User Assisted Composition Tool

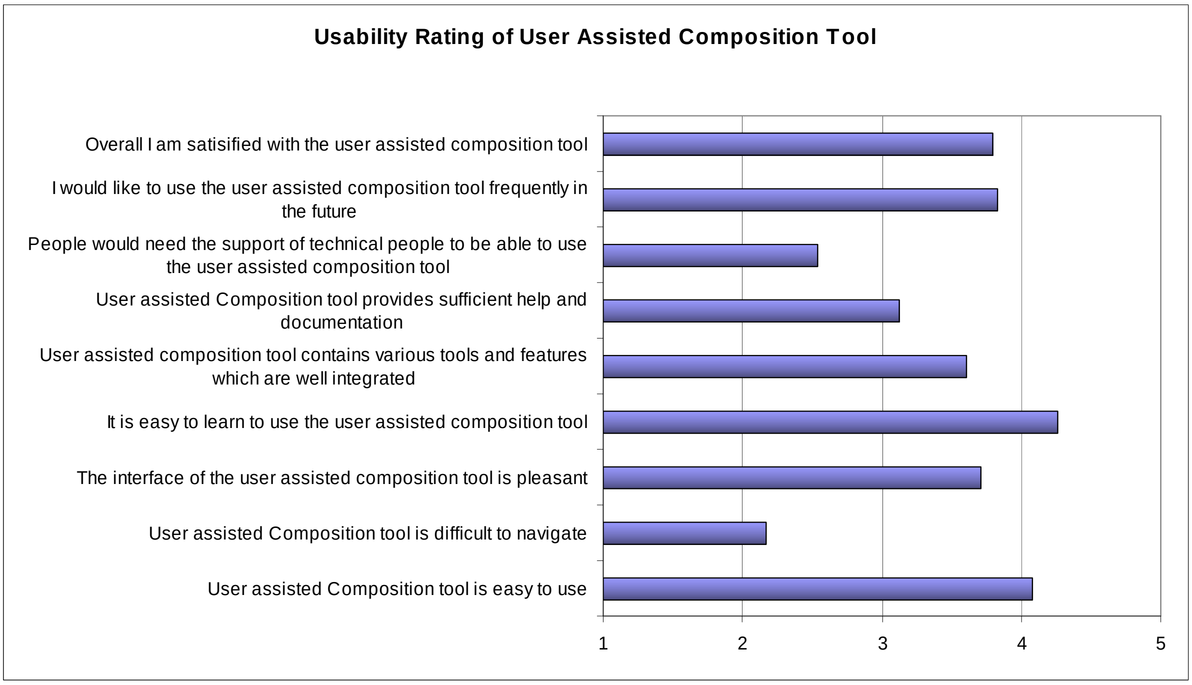
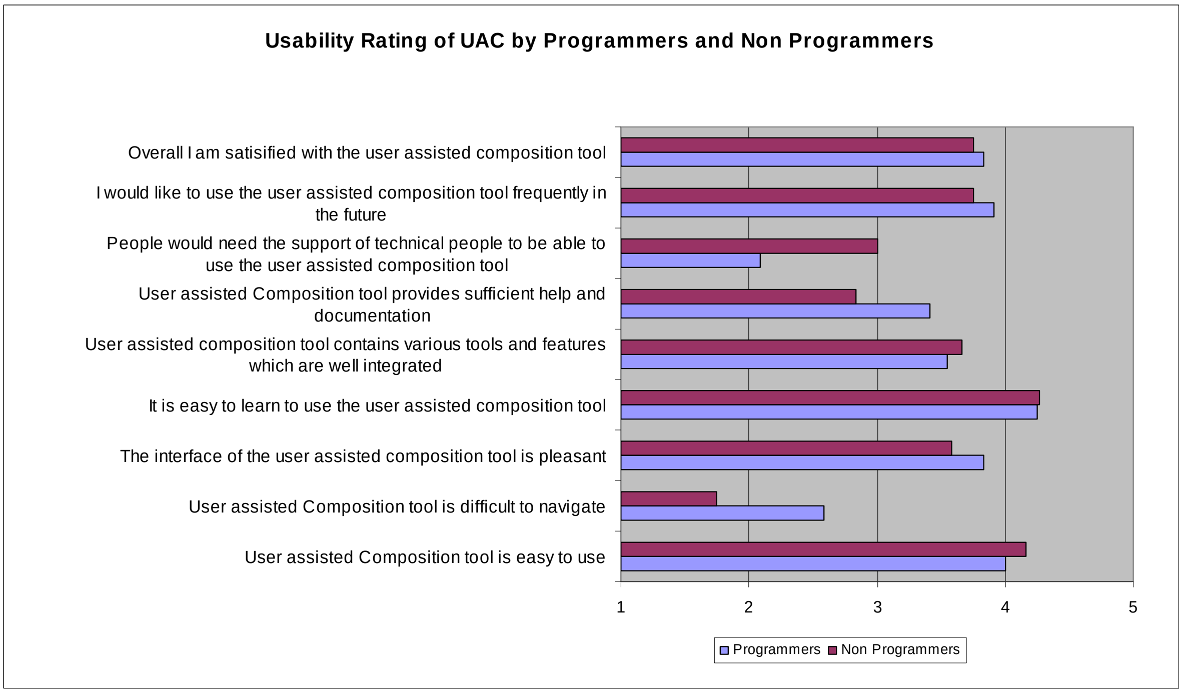


Figure 12. Usability Rating of UAC Tool by Programmers and Non-programmers



The User Assisted Composition tool received positive feedback from the participants of our study. While performing their task on the User Assisted Composition tool the users found the composition process easy and user friendly as it does not involve coding or use of programming languages. Users were particularly pleased by the organization of services and the simple click feature of the assisted composition tool. The tool was rated high for its simplistic nature as it allowed users to complete the composition process in an efficient and effective way within a single window. Users also liked the option of selecting an deselecting services/components in an intuitive way. In this respect, for users the tool made it easy to understand the relationship among various services in a composition along with the logical sequence of events underlying a composition process. The automated reasoning mechanism that automatically computes the compatibility of services in a composition was also praised by our users. Users though that the by making everything available at one place the assisted composition tool can enable non technical people to compose service-based applications.

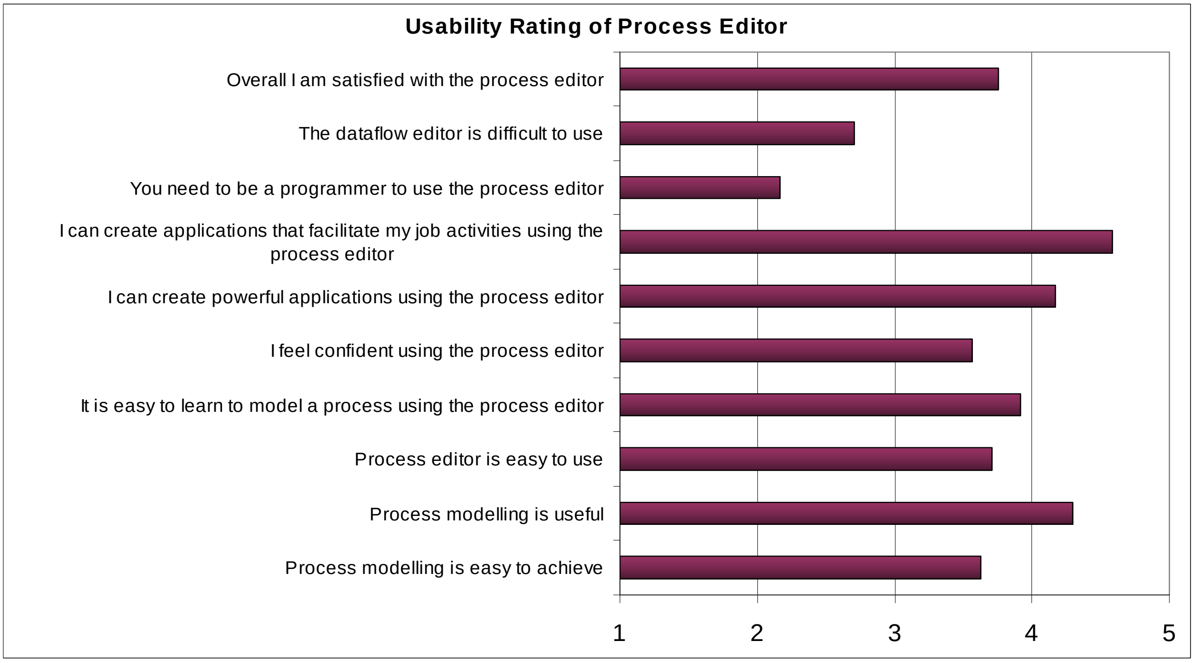
The user interface of assisted composition tool was perceived as simple, intuitive and straightforward by the users. According to one user *the UI makes sense and is logical*. In this respect, the users liked the clear structure of the tool as well as the colour scheme used to represent various elements in the tool.

Apart from the above, some users needed some introductions to the tool while using it for the first time. In this respect, the help features or help material available within the tool was not detailed enough for some users. Furthermore, some users were not able to realize the presence of a left-right scroll bar at the bottom of the tool. To address these issues users suggested including comprehensive help features. Also, a suggestion was to include service rating/rankings for each service in a category that can help users in selecting an appropriate service for a particular task.

2.1.4.2.3 Process Editor

To assess the usability of the Process Editor we devised a questionnaire consisting of 19 questions to cover different features and aspects of the tool (see Workbook in Annex A). The Process Editor of the SOA4All Studio was positively rated as easy to use (m= 3.70, std= 0.85) and easy to learn (m= 3.91, std= 1.17). Users also felt this tool can be used to create powerful applications (m= 4.16, std= 1.00) and applications which facilitate their daily activities (m= 4.58, std= 0.92). However, they disagreed that one needs to be a programmer in order to model processes using the SOA4All Process Editor (m= 2.16, std= 1.12) making the tool suitable for Non-programmers as well.

Figure 13. Usability Rating of SOA4All Process Editor



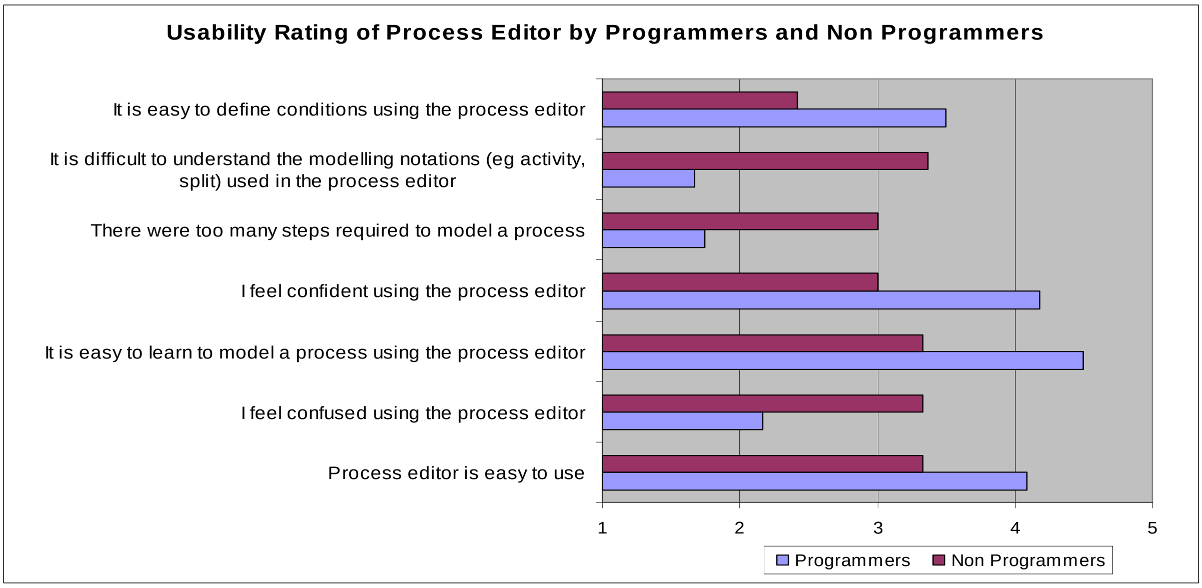
The process modelling activity was regarded as useful (m= 4.29, std= 0.90) and easy to achieve (m= 3.62, std= 0.87). Users were in doubt whether that the Dataflow Editor, a plug-in within the Process Editor that allows specifying dataflow between services of a process, is difficult to use (m= 2.70, std= 1.39). Generally users were satisfied with the Process Editor (m= 3.75, std= 0.84). Figure 13 and Table 3 depict rating of all usability questions related to the Process Editor and its components.

Table 3. Average usability scores of the Process Editor

Usability question	**Average usability score**	**Standard Deviation**
Process modelling is time consuming	2.87	1.15
I feel confused using the Process Editor	2.75	1.29
There were too many steps required to model a process	2.34	1.07
It is difficult to understand the modelling notations (eg activity, split) used in the Process Editor	2.47	1.23
It is easy to define conditions using the Process Editor	2.95	1.16
Binding services/operations to activities is a difficult task	2.12	1.36
I know which services/operations to bind to the activities of the process model	3.50	1.14
I know which input of a service to map to which output of a preceding service in the dataflow editor	3.58	1.06
I would like to use the Process Editor frequently in the future	3.41	1.17

The Process Editor is the SOA4All tool that created most differences in user rating between Programmers and Non-programmers. A total of 7 questions out of 19 were rated differently by the two groups of users as shown in Figure 14, (T-tests, P<0.05). Programmers felt more confident, less confused, and found the Process Editor easier to learn than Non-programmers. On the other hand, ratings revealed that Non-programmers struggled to understand modelling notations, such as parallel split, and define conditional branches, and thought that modelling a process involves too many steps. Overall satisfaction with the Process Editor by the two groups of users (mean of programmers' satisfaction= 4.08 and mean of non-programmers' satisfaction= 3.42) did not differ statistically.

Figure 14. Usability Rating of SOA4All Process Editor by Programmers and Non-programmers



After using the relatively simple UAC tool our users found the Process Editor more detail oriented. The drawing activity in the Process Editor was easy to understand for our users. Users thought that the Process Editor offered good balance of giving users more freedom but still hiding a lot of complexity. Users found it easy to add activities in a process and process model was quite easy to understand even for people with no technical background. The binding of services with activities was straightforward and self explanatory for most of the users. Users were also able to specify conditions in the process model. In this respect, some users found the Process Editor more user-friendly then the User Assisted Composition as it shows the mechanism behind the process. Overall, users felt confident while performing their task in the Process Editor.

On the other hand, some users were unable to define parallel activities using parallel split and parallel merge. Moreover, some users found process modelling daunting and technical owing to the programming like nature of specifying conditions and bindings. However after seeking minimal guidance from the moderator the users were confident that they can design processes without further assistance. To alleviate some of these concerns the users suggested to add more helpful features in the Process Editor e.g. the use of more general words instead of technical terms.

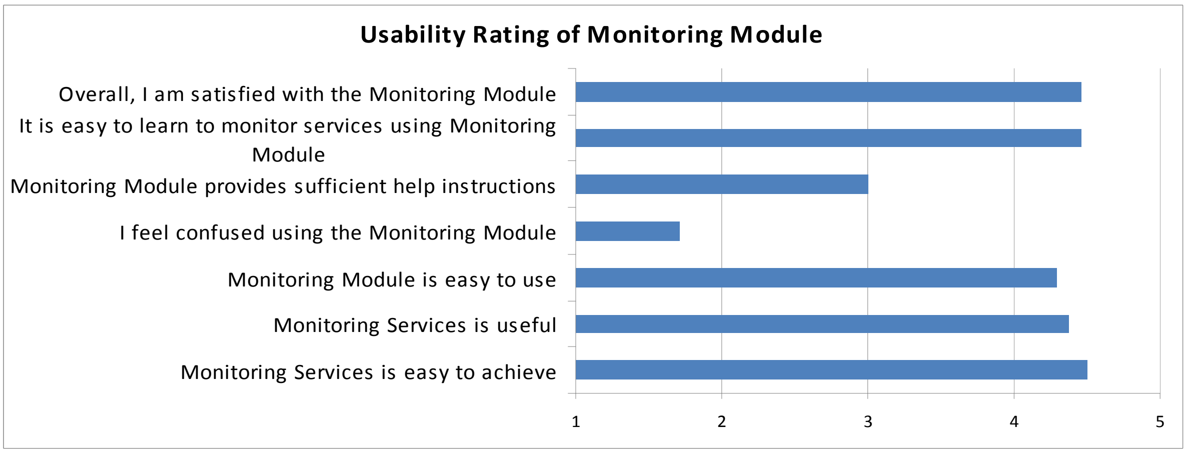
2.1.4.2.4 Monitoring Module

The Monitoring Module was the last tool to be evaluated by our participants in this first user study. Ratings demonstrated great satisfaction with the tool and users found it easy to learn to monitor services (m= 4.46, std= 0.66). The Monitoring Module was regarded as easy to use (m= 4.29, std= 1.04) but participants were unsure about the level of help provided by the Module (m= 3, std= 1.29). Monitoring the behaviour and performance of services was regarded as a useful (m= 4.38, std= 0.71) and easy to achieve task (m= 4.5, std= 0.72) as shown in Figure 15.

Comparing Programmers and Non-programmers' ratings of the Monitoring Module showed no apparent differences in their perception in regard to their satisfaction, ease of learning, ease of use, and usefulness of service monitoring (The task involving Monitoring Module was the easiest part for our users. The Monitoring Module was praised for its GUI which was simple and easy to navigate. Users found the information (within the Monitoring Module) clearly presented and useful. Overall the tool was perceived as user friendly and easy to use by non-programmers. For some users the information provided in the Monitoring Module represented very important statistics that can be used to differentiate between services that they can use in their applications. However, for some users the information provided in the Monitoring Module was not very important. Moreover, some users reported that the font size is too small for them. On the other hand, some users pointed out that by introducing units (e.g. milliseconds for average response time) the information could be made easier to understand.

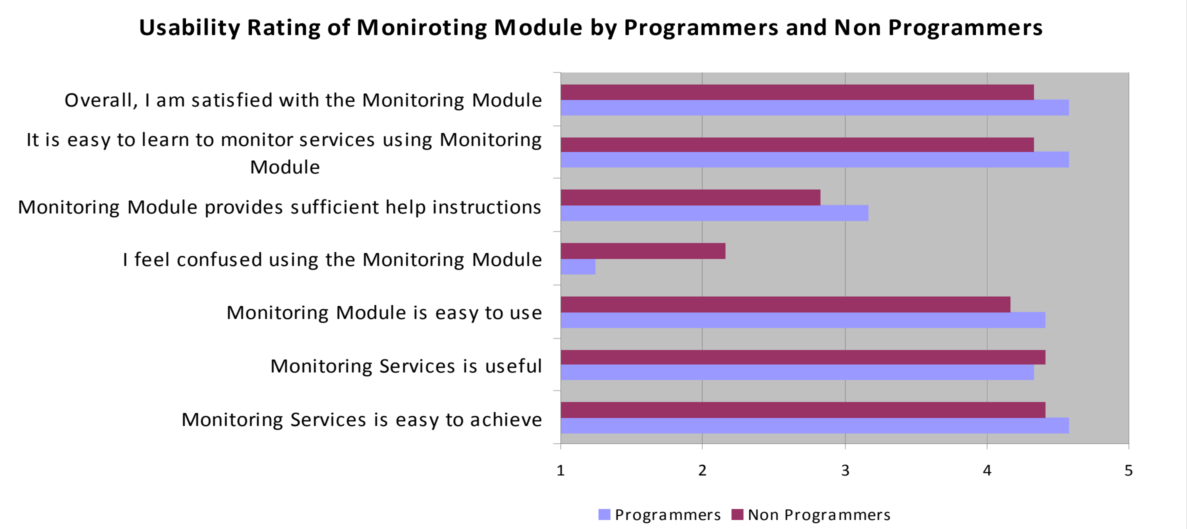
Figure 16). The only slight difference was related to the confusion when using the Monitoring Platform; Non-programmers (m= 2.16) were slightly more confused than Programmers (m= 1.25), (T-test, P=0.051).

Figure 15. Usability Rating of Monitoring Module



The task involving Monitoring Module was the easiest part for our users. The Monitoring Module was praised for its GUI which was simple and easy to navigate. Users found the information (within the Monitoring Module) clearly presented and useful. Overall the tool was perceived as user friendly and easy to use by non-programmers. For some users the information provided in the Monitoring Module represented very important statistics that can be used to differentiate between services that they can use in their applications. However, for some users the information provided in the Monitoring Module was not very important. Moreover, some users reported that the font size is too small for them. On the other hand, some users pointed out that by introducing units (e.g. milliseconds for average response time) the information could be made easier to understand.

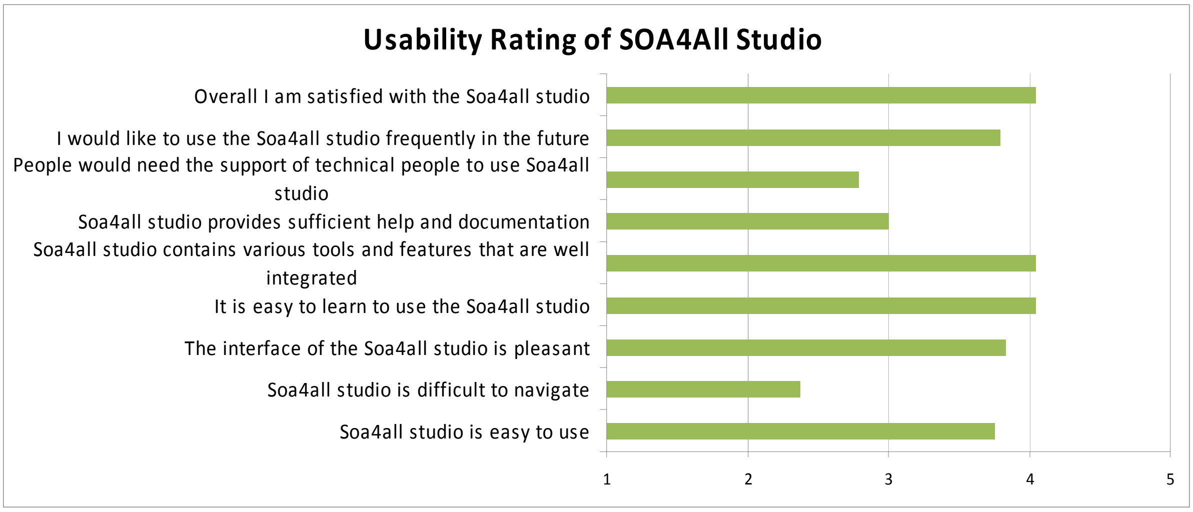
Figure 16. Usability Rating of Monitoring Platform by Programmers and Non-programmers



2.1.4.2.5 SOA4All Studio

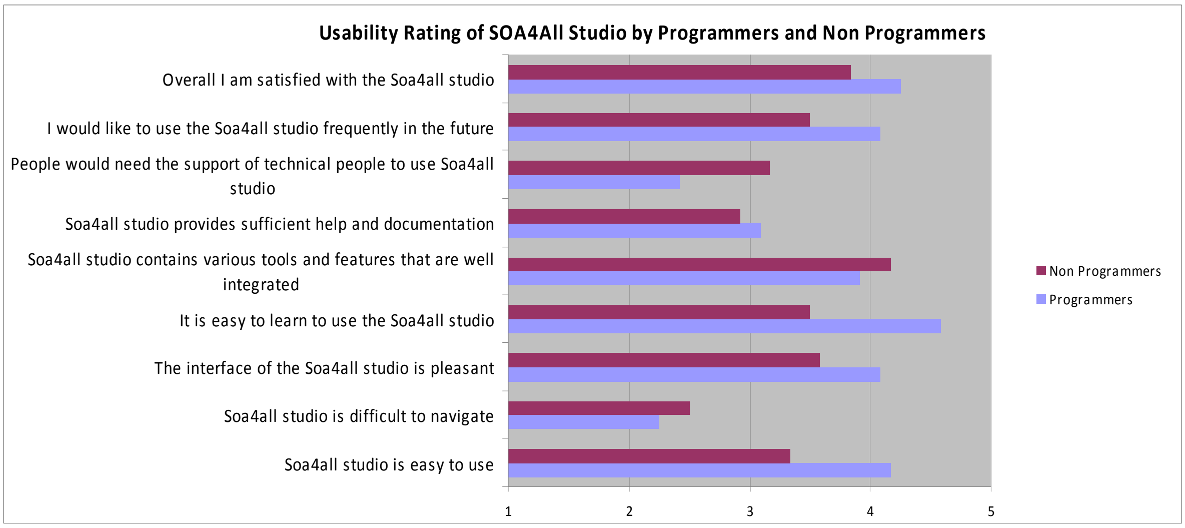
Upon completing all development tasks, we instructed our 24 participants to evaluate their whole experience using the SOA4All Studio and report on their opinion openly and honestly. This holistic evaluation included rating their ***degree of agreement*** with the usability questions on a 5-point Likert scale where 1 signifies “Strongly disagree” and 5 signifies “Strongly agree” and enumerating the top positive and negative features of the Studio. Participants found the Studio easy to use (m= 3.75, std= 0.74), easy to learn (m= 4.04, std= 1.08), pleasant (m= 3.83, std= 0.96), and expressed willingness to use it regularly in the future (m= 3.79, std= 1.14). The different tools of the SOA4All Studio were deemed to be well integrated together (m= 4.04, std= 0.91). Overall satisfaction with the SOA4All Studio was positive (m= 4.04, std= 0.86) demonstrating strong acceptance of the Studio by our participants as depicted in Figure 17.

Figure 17. Usability Rating of SOA4All Studio



Overall rating of *ease of use* and *ease of learning* of the SOA4All Studio significantly differed between the two groups of end users with Programmers' scores to these questions (m= 4.17 and m= 4.58 respectively) averaging higher than Non-programmers' (m= 3.33 and m= 3.5 respectively), (T-tests, P<0.01). Such finding is anticipated due to the technical knowledge of programmers and their use of more advanced tools on a daily basis. Programmers (m= 2.42) and Non-programmers' (m= 3.17) need for technical support from experts to successfully use the SOA4All Studio was similar since the means were not statistically different. Those values were (e.g. 3.17) still neutral confirming that even Non-programmers do not require special skills to operate the Studio. Other usability dimensions were rated similarly and no statistical differences were found, as depicted in Figure 18.

Figure 18. Usability Rating of SOA4All Studio by Programmers and Non-programmers



The SOA4All Studio received positive reviews and feedback from our participants agreeing with the subjective usability scores. They found the user interface straightforward and easy to understand; some users compared the presentation to more familiar software like Windows. The graphics were considered user-friendly. In general users found the tools powerful and easy to use and learn. Users believed that the SOA4All tools can empower people without IT knowledge to create complex applications. Users also reported that the tools can enable non-programmers to perform various activities at a single platform. In this respect, many users believed that the Studio has the potential to be adopted for wide spread use and they could see themselves using the Studio in the future.

On the other hand, users also pointed out some weaknesses in the Studio that can be improved in the future. In this regard, most of user concerns revolved around the navigation aspects and the GUI e.g. some users wanted improvements in the visibility and font size at certain places in the Studio. Other concerns were largely related to help features, where users wanted to get more help or assistance in performing their tasks in the Studio.

2.2 User Testing Two

2.2.1 Participants' Background

Following the first user study which evaluated the Consumption Platform, User Assisted Composition Tool, Process Editor, and Monitoring Module, we conducted a follow-up user study to evaluate the rest of the SOA4All tools covering the Consumption Platform, WSMO-Lite Editor, and Discovery Module. We re-iterated the evaluation of the Consumption Platform to show the power and usefulness of annotations in finding desired services as the Consumption Platform, serving as a benchmark, searches for non-annotated services whereas the Discovery Module retrieves annotated services.

A total of 10 end users took part in the second evaluation of SOA4All tools. The majority of these participants (7 out of 10) come from Computer Science disciplines and had extensive programming and development experience using a variety of programming languages such as Java, C, and C++. The remaining 3 participants had no prior software development experience and come from Non-Computer Science disciplines. We used the responses to the screening questionnaire (in Annex C) to select and invite suitable candidates for the purpose of this evaluation based on their technical expertise. The primary target user group of this second user study is programmers as both the WSMO-Lite Editor and Discovery Module are specialised tools intended for use by a technical audience. The goals of this evaluation are as follows:

1. Testing the usability of the Consumption Platform, WSMO-Lite Editor and Discovery Module and evaluating user experience in using these specialised tools
2. Showing the usefulness of semantic annotations in finding annotated services more efficiently

2.2.2 Evaluation Procedure

Similar to the first user study, this second summative testing aimed to gather users' attitudes and acceptance toward the SOA4All Studio and its tools and consisted of three main phases:

(d) training phase,
(e) development phase, and
(f) rating phase.

A detailed description of each phase is presented below.

2.2.2.1 Training Phase

During the training phase, all participants attended a 1-hour group presentation which briefly explained the purpose of the SOA4All project and thoroughly demonstrated the various features and aspects of each of the designated SOA4All tools (i.e. Consumption Platform, WSMO-Lite Editor and Discovery Module). This included a tutorial demonstrating how to search for services and how to annotate WSDL-based services using the WSMO-Lite editor. At the end of the presentation, participants were encouraged to comment and ask questions about what they have seen. In summary, the training phase encompassed the subsequent steps:

1. A group presentation and training explaining how one can use the Consumption Platform and Discovery Module to find services, and WSMO-Lite Editor to annotate services;
2. A post-training session for participants to ask elaborating questions about the presented tools;

Following the training sessions, each participant was taken to an individual room where they completed a set of development tasks independently as explained in the next section.

2.2.2.2 Development Phase

Initially users were provided with the workbook listing the tasks to be performed covering three aspects of the service development lifecycle: consumption, annotation, and discovery of services. In this usability testing we opted the think-aloud protocol [1] through which participants verbalised their thoughts during the development activities to unravel their development strategies and obtain their opinion as they undertake the tasks. We recorded user interaction behaviour for each tool and their vocal discourse using a screen capturing software (i.e. Snag it) for post-experiment analysis. The precise steps undertaken by each participant in the development phase are as follows:

- **Pre-test Interview:** participants discussed their existing experience and opinions about Software and Service Development using different software development environments; this pre-test interview lasted approximately 3-5 minutes.
- **Interaction with the SOA4All Studio and its tools:** participants interacted with the relevant tools of the “SOA4All Studio” and completed a set of diverse development tasks (consumption, annotation, and discovery) in order to fulfil the tasks described in Annex B.

2.2.2.2.1 Description of Development Tasks

The tasks of this user testing included: finding and adding two services (SearchUniversity and BanckAccount) to the list of favourites using the basic search functionality of the Consumption Platform, adding semantic descriptions to the SchoolContact service data types and its operation (checkSMSAvailability), and finally finding annotated services including the annotated SchoolContact by specifying its input and out parameters. Refer to Annex B for the complete list of tasks and their accurate description.

The aim is to enable end users to interact with the features of the Consumption Platform, WSMO-Lite Editor, and Discovery Module. Users start by looking for non annotated services, annotate services with semantic descriptions, and find annotated services. This enables users to see the affect of annotation by specifying the input and output parameters of the service to be found, one of the unique features of SOA4All Studio.

2.2.2.3 Rating phase

The rating phase focused on evaluating user development experience and usability of constituent tools of the SOA4All Studio. Following each development task (i.e. consumption, annotation, and discovery), participants were asked to rate a number of usability dimensions covering various features of the respective tool (i.e. Consumption Platform, WSMO-Lite Editor, Discovery Module) by completing a paper-based questionnaire. Participants expressed their degree of agreement to usability questions, such as ease of use and ease of learning, on a 1 to 5 rating scale where (1 =strongly disagree and 5 =strongly agree). They were also asked to comment on their experience and indicate their opinions about the tools in a post-test interview highlighting the positive and negative aspects as well as suggestions for future improvements.

The WP2 evaluation workbook covering the details of aforementioned phases (i.e. development and rating phases) of user study two is included in Annex B.

2.2.3 Analysis method

User opinion data were collected through means of think-aloud protocol, interviews and questionnaires where participants commented on the core functionalities of the Studio and rated different usability dimensions, such as ease of learning and ease of use, on 1-5 Likert rating scale. Self-reported data were initially recorded by both the participants who filled out some questions in the workbook and the moderators (i.e. experimenters) who took notes as the experiments unfolded. Researchers walked through the interaction videos and calculated the completion time for each task per user. Quantitative and qualitative data was then transcribed into spreadsheets for subsequent analysis.

Since this is a summative evaluation, the researchers primarily focused on analysing the performance data (e.g. completion time) and usability scores as they reflect users' satisfaction and attitudes toward the implemented tools and concepts of SOA4All project. However, for each tool we include a summary of users' opinion.

2.2.4 Usability Results from the Second User Testing

We organised and summarised the results from this study into two sections: user performance covering average completion time for each task type and tools' rating covering average usability scores of different features of the tools and user feedback covering positive and negative aspects of the tools as well as ways for improving them.

2.2.4.1 User Performance

We measured the time users spent to complete the consumption, annotation, and discovery tasks and calculated the average completion time. The results (Table 4) show that users spent the longest time annotating the service data types and its operation with ontology concepts (m=19.1 minutes), followed by searching for annotated services using ontology concepts (m=16.1 minutes), and finally finding services using a common text matching engine by simply specifying the name of the service (m= 5.3 minutes). The development tasks differed in complexity and number of steps required to complete them.

Table 4. Average completion time for service consumption, annotation, and discovery

SOA4All Tool	Task	Average completion time (minutes)	Standard deviation
Consumption Platform	**Finding non annotated Services**	5.3	2.21
WSMO-Lite Editor	**Annotating Services**	19.1	4.2
Discovery Module	**Discovering annotated services**	16.6	6.63

2.2.4.2 Usability Rating and User Feedback toward the SOA4All Tools (Consumption Platform, WSMO-Lite Editor and Discovery Module)

This section summarises and presents average usability scores of usability dimensions rated on a 5-point Likert scale, where 1 corresponds to *Strongly Disagree* and 5 corresponds to

Strongly Agree, and user opinion about the components and functionality of each of the following tools: Consumption Platform, WSMO-Lite Editor and Discovery Module.

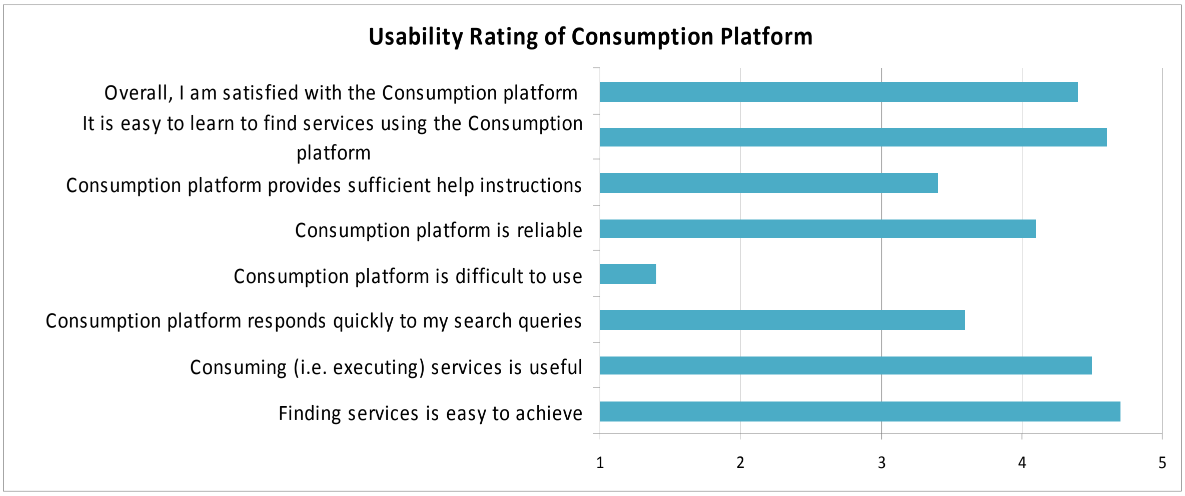
2.2.4.2.1 Consumption Platform

Users' rating of the Consumption Platform in the second user study closely mirrored those of the first user study. Participants found the tool reliable (m= 4.1, std= 0.57) and easy to learn (m= 4.6, std= 0.52) probably since it operates as most commonly used search engines. They strongly disagreed that the Consumption Platform is difficult to use (m= 1.4, std= 0.97). Overall, participants were satisfied with the Consumption Platform (m= 4.4, std= 0.7) as shown in *Figure 19*.

Overall the users liked the Consumption Platform interface, finding it to be straightforward, fast and easy to use, with the search mechanisms felt to be especially practical. Users also liked the way different search results can be opened in different windows so that a comparison can be made. Owing to its simplicity, users pointed out that they can quickly learn how to interact with the platform in order to perform their tasks. Moreover, users found it easy to look for the details of services and add service operations in their favourites. The added feature of service ratings was also praised by the users, who found it useful in selecting appropriate services for a particular task.

While there were a few complaints about specific features, the main drawback of the scenario being tested was the difficulty of distinguishing between the different services that could potentially fill a specific role. These were only distinguished by their names, and the provision of information such as reliability was felt to be useful.

Figure 19. Usability Rating of Consumption Platform

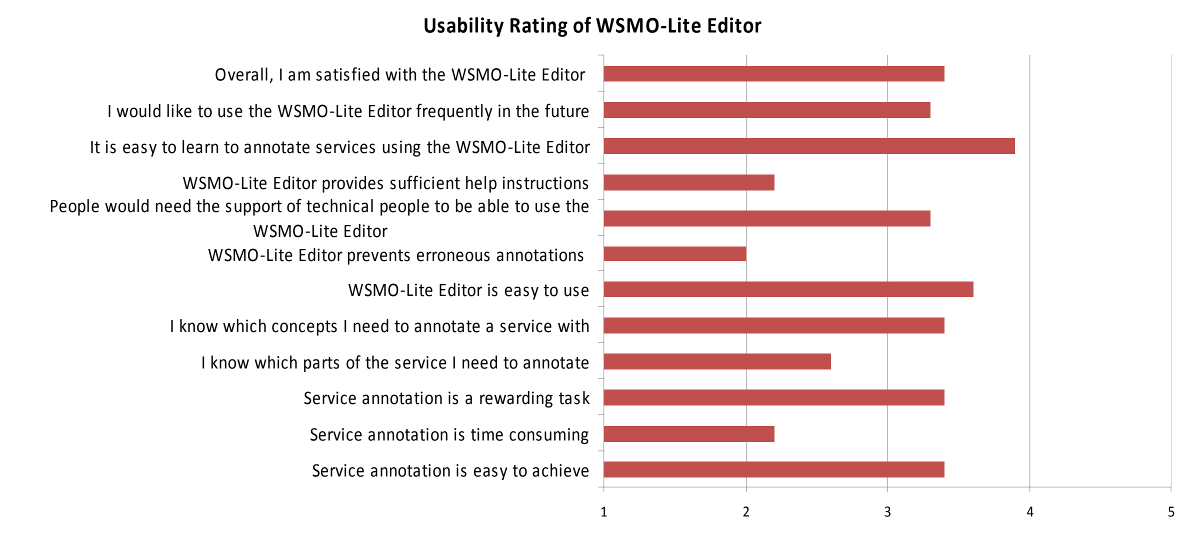


Apart from the above, users wanted to see latest help features to assist them in the search. For example a user wanted to see auto search suggestions when typing out search terms. Other suggestions were mostly related to the cosmetic aspects of the Consumption Platform e.g. the placement of 'add to favourites' button and the order of search results.

2.2.4.2.2 WSMO-Lite Editor

Participants indicated that the WSMO-Lite Editor is easy to learn (m= 3.9, std= 0.99) and easy to use (m= 3.6, std= 0.97). However, they disagreed that the WSMO-Lite Editor provides enough instructions to assist end users (m= 2.2, std= 1.03) and prevents erroneous annotations (m= 2, std= 0.94). They were also confused as regard to which concepts need to be used (m= 3.4, std= 1.17) and which service parts need to be annotated (m= 2.6, std= 1.17). In general, participants were satisfied with the WSMO-Lite Editor (m= 3.4, std= 0.97). The main difficulties of using the WSMO-Lite Editor stem from its complexity and lack of knowledge within our users regarding semantic annotation, yet the importance and feasibility of semantically annotating services were supported by our test users.

Figure 20. Usability Rating of WSMO-Lite Editor



Overall the interface for the annotation platform was felt to be user friendly with some minor features commended and the drag and drop functionality for attaching annotations liked. Users praised the organization and structure of the annotation platform. The information content in the annotation platform was found to be clear and easy to understand by our users, where some of the users reporting short learning curve to perform their tasks. Users also understood the purpose of annotation and found it a useful activity. The actual annotation was straightforward for most of the users owing to the simplicity of drag and drop feature.

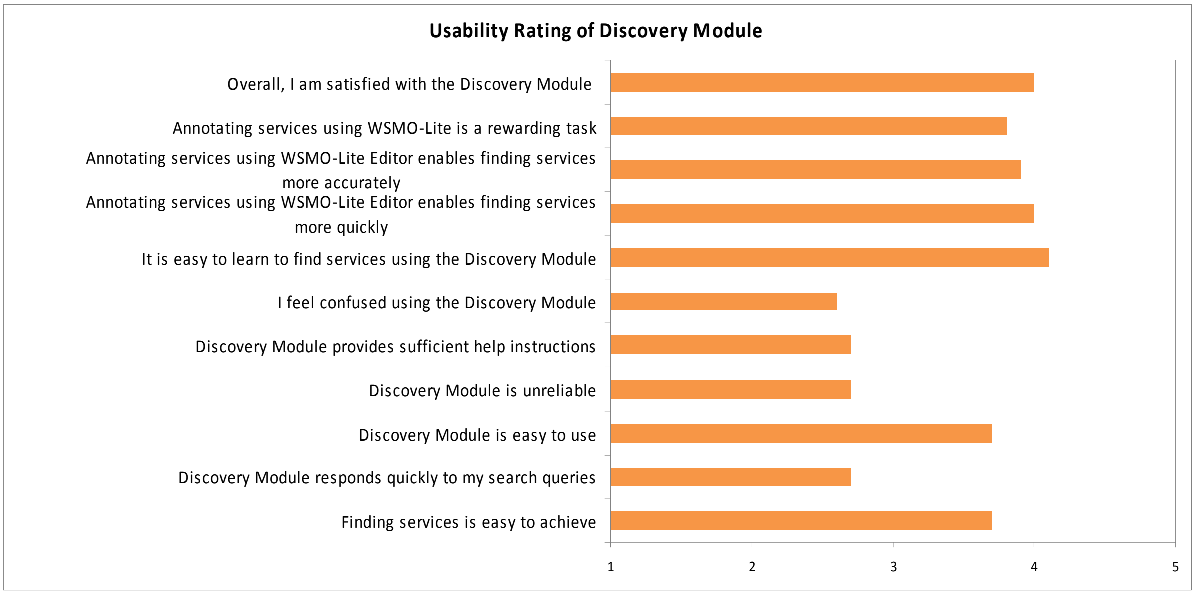
At an interface level the principal user complaint was that the icons were difficult to distinguish, something which tooltips would alleviate. Overall though, the principal complaint seems to be that this was a relatively technical task for which the users seemed to lack sufficient knowledge. Future exploitation work would therefore need to focus on “helper” modules (wizards, help files, tutorials) to alleviate this problem.

2.2.4.2.3 Discovery Module

Participants found it quite easy to learn to find services using the Discovery Module (m= 4.1, std= 0.88) and easy to use (m= 3.7, std= 1.06). However, they felt confused and felt that not enough guidance and documentation are provided by the Discovery Module making the specification of input, output, pre-conditions and post-conditions a more challenging task.

The effect of annotations was positively rated by our participants; they felt that adding semantic descriptions to services enables finding services more accurately (m= 3.9, std= 1.1) and quickly (m= 4.0, std= 1.15). The task of service annotation was viewed as more rewarding (m= 3.8, std= 0.92) after using the Discovery Module (prior to using the Discovery Module m= 3.4, std= 1.17). Participants were generally satisfied with the Discovery Module (m= 4.0, std= 0.47) as shown in Figure 21.

Figure 21. Usability Rating of Discovery Module



Overall the users found the Discovery Module to be both functional and logical, with some preference being shown for the advanced search feature. The GUI was found to be standard and user friendly, while some users reported that the layout of the interface helps in using the tool. The distinction between advanced and basic search features was also liked by the participants. Furthermore, participants praised the categorization as the tree listing of categories maps well with the basic knowledge of non technical users. Users also reported that the categories make it easy to find services.

Apart from the above, the speed of the search and the difficulty encountered in distinguishing the between the services located were highlighted as potential issues. Moreover, some users found the categories not enough to represent potential services; on the other hand some users felt that too many categories would be confusing.

As with annotation the relatively unfamiliar nature of the task meant that the users felt more extensive help based features would be useful and for a real application the development of such features is strongly recommended. Such helper software could be produced as part of future exploitation activities. Other suggestions related to the look and feel of the Discovery Module, for example some users suggested adding a cancel button for situations when search takes too much time.

3. Conclusion

This deliverable (D2.5.3) has concluded the usability evaluation efforts of WP2 and reported on the results of two summative user studies aiming to evaluate the usability of the SOA4All Tools and user interaction and development experience with these tools. These two studies included a total of 34 end users; 24 of those in the first study evaluating the Consumption Platform, the User Assisted Composition view and the full view on the Process Editor, and Monitoring Module and the remaining 10 user in the second study evaluating the Consumption Platform, WSMO-Lite Editor, and Discovery Module. At the beginning we developed and administered an online screening questionnaire that was distributed to staff and student mailing lists in the University of Manchester. The questionnaire focused on collecting information about programming and development experience, modelling tools and technologies used by our end users. For the first study we selected and invited 2 groups of users: 12 Programmers and 12 Non-programmers, whereas for the second study we invited 7 Programmers and 3 Non Programmers. All participants of both studies received the same treatments, they attended a 1-hour group training session where they were tutored how to use the different SOA4All Tools and had the chance to ask elaborating questions. Following the training, each participant took part in a usability test lasting approximately up to 1 hour and a half where they completed a set of development tasks independently. User interaction experience, comments, and ratings were recorded by means of screen capturing programs and questionnaires for subsequent quantitative and qualitative analysis. Throughout the usability studies users were instructed to verbalise their thoughts using the think aloud protocol [1] which revealed the features users favoured in the SOA4All tools.

Figure 22. Usability Rating of SOA4All Studio and its Tools

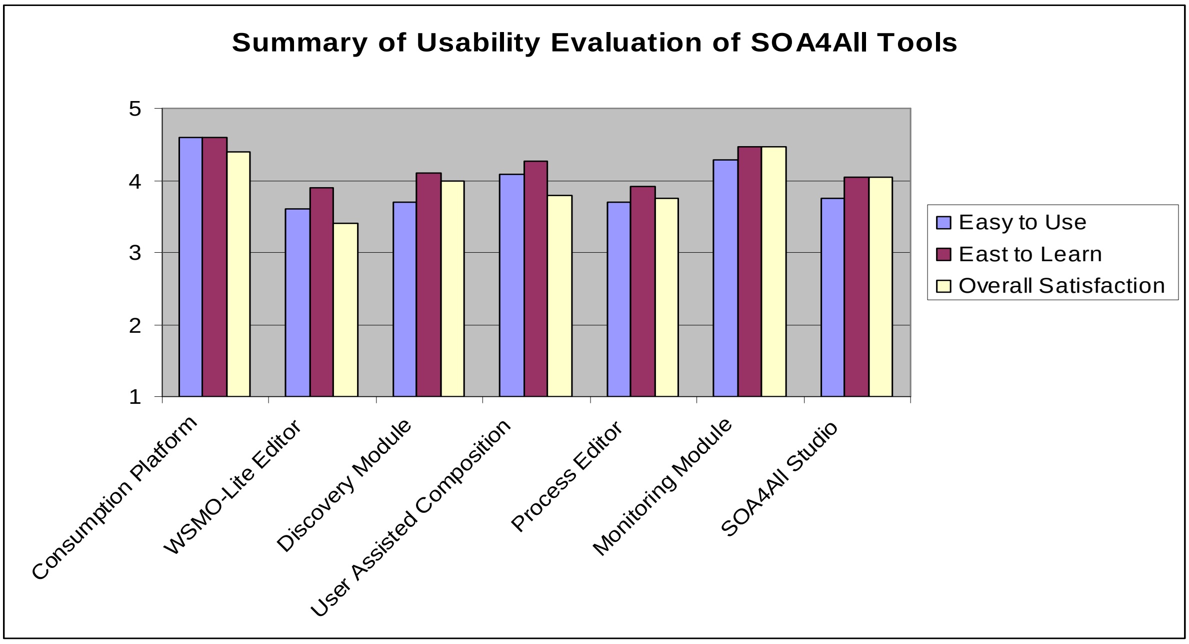


Figure 22 summarises average rating to three main usability questions: “ease of use”, “ease of learning” and “overall user satisfaction with the tool” on a 5-point Likert scale where 1 signifies Strongly Disagree, 5 signifies Strongly Agree, and 3 is the neutral value. Users’ average rating of the Consumption Platform, Discovery Module, WSMO-Lite Editor, User Assisted Composition Tool, Process Editor, and Monitoring Module were favourable and very promising. Most of the usability scores exceeded or approached ‘4’ confirming that our participants were able to learn to use the tool quickly and implement the development tasks easily without problems. The Consumption Platform and Monitoring Module were the easiest tools to learn and use whilst the WSMO-Lite Editor was the most challenging SOA4All tool to use primarily due to users' unfamiliarity with semantic annotation and the formalism used with the WSMO-Lite Editor. We are confident that further training and comprehensive documentation of this tool will increase user acceptance and satisfaction drastically.

Although our participants were unfamiliar with the SOA4All tools and attended a rather short training session, they were able to perform the evaluations tasks fairly quickly and without major problems. The SOA4All tools fulfilled the objectives they promise to fulfil as they enabled our participants to execute the tasks leaving a good impression since the majority of our users indicated that they will return to use the tools more frequently in the future. The simplicity and look and feel of the tools were praised by the users as well as the ability to directly manipulate things (e.g. drag and drop feature). Participants also stated that the high level of abstraction used within the tools will enable a wide spectrum of users to interact with them as the low level technical details are hidden, fulfilling the '4All' objective. This clearly evinced from the results of the first study where Programmers and Non-programmers' usability scores and comments were contrasted. Both target groups were highly satisfied with the SOA4All tools and very few differences were evident from the statistical analysis despite their very differing profiles and technical skills.

4. References

[1] Lewis, C. H. (1982). Using the "Thinking Aloud" Method In Cognitive Interface Design. Technical Report IBM RC-9265

[2] Mehandjiev, N. Lecue, F. Wajid, U. Namoun, A. (2010). Assisted Service Composition for End Users. In Proceedings of the 2010 Eighth IEEE European Conference on Web Services (ECOWS '10). IEEE Computer Society, Washington, DC, USA, 131-138

[3] Deliverable D2.5.1. SOA4All Evaluation

[4] Deliverable D2.5.2. Summative Evaluation Report

[5] Statistical Package for Social Sciences. http://www.spss.com/corpinfo/index.htm

[6] Independent T-Test using SPSS. http://statistics.laerd.com/spss-tutorials/independent-t-test-using-spss-statistics.php

[7] Snagit, A screen capturing software. http://www.techsmith.com/snagit/default.asp

[8] Web Services Description Language (WSDL) 1.1. http://www.w3.org/TR/WSDL

[9] iServe, A platform for publishing Semantic Web Services. http://iserve.kmi.open.ac.uk/

[10] Benyon, D. Turner, P. Turner, S. (2004). Designing Interactive Systems - People, Activities, Contexts, Technologies

[11] Rensis, L. (1932). "A Technique for the Measurement of Attitudes". Archives of Psychology 140: 1–55

[12] Usability Professionals' Association. What is User-Centred Design? http://www.usabilityprofessionals.org/usability_resources/about_usability/what_is_ucd.html

Annex A – Evaluation Workbook 1

EVALUATION OF SOA4ALL STUDIO: CONSUMPTION, USER ASSISTED MODELLING, PROCESS MODELLING, AND MONITORING

Participant Information Sheet

The aim of this software testing is to collect your opinion about SOA4All Studio, an integrated service development environment, and one of its tools, the User Assisted Composition Tool. Their aim is to empower people who are not professional developers to build software services without the need to write programming code.

The current experiment consists of four main parts and will take approximately 1 hour to complete.

PART A – Pre-test Interview (~ 5 min)

1. Discuss your existing experience and opinions about Software and Service Development

PART B – Interaction with SOA4All Studio (~ 55 min)

1. Complete a set of development tasks (Task1, 2, 3, and 4) using SOA4All Studio and the User Assisted Composition Tool.
2. Rate your satisfaction with the User Assisted Composition Tool and SOA4All Studio.
3. Discuss your experience and opinions about the User Assisted Composition Tool and SOA4All Studio

PART A – Prior Knowledge and Experience using the User Assisted Composition Tool and SOA4All Studio

- Full Name of Participant: …………………………………………………………..

- Please indicate your knowledge and expertise with the User Assisted Composition Tool and SOA4All Studio, where 1 corresponds to poor and 5 corresponds to excellent:

Service Development Tool	I have used it before?	Experience 1 2 3 4 5
User Assisted Composition Tool	Yes □ No □	poor □ □ □ □ □ excellent
SOA4All Studio	Yes □ No □	poor □ □ □ □ □ excellent

PART B – Interaction with SOA4All Studio and the User Assisted Composition Tool

Scenario Description

This scenario describes the registration process that overseas students go through while getting admission in UK universities.

Your goal is to complete an overseas student registration process. For this you need to develop a software application which allows you to search for a UK university, register for a course in the university and find an accommodation. There are two ways for paying the university fee, the first way is to open a bank account and get funds transferred into that account. The bank account can be used to make payment for the university fee. In the second way you can request a letter from a sponsor and submit that letter to the university. You must choose only one way to pay the university fee. After paying the university fee you will register with the NHS.

Description of available services and their operations

University Service

Operation:
SearchUniversity- given the name of a city the operation returns a list of universities in that city. The input is a string indicating the city name, the output is a list of strings or '0' if no university is found.

CourseRegistration Service

Operation:
CourseRegistration- register a course in the university. The inputs are the course details, the output is registration confirmation.

FindAccomodation Service

Operation:
FindAccomodation–The input to this operation is the accommodation name (that can be found on the accommodation list on university website), the output is booking confirmation for the accommodation (keep in mind oversees students are guaranteed a place in university accommodation of their choice).

OpenBankAccount Service

Operation:
OpenBankAccount – open a bank account. This operation takes the account request as input and the output is the confirmation of account opening.

TransferFunds Service

Operation:
TransferFunds – Transfer funds between two accounts. The input is the transfer request, the output is transfer conformation.

SponsorLetter Service

Operation:
SponsorLetter- Get letter from sponsor. The input is the letter request. The output is a sponsor letter.

PayUniversityFee Service

Operation:
PayUniFee- Pay university fee. The input for this operation is a request, which can be either bank account details from where money will be debited or a letter from sponsor. The output is 0 (in case student wants to pay using a bank account) or 1 (in case student has a sponsor letter).

RegisterNHS Service

Operation:
RegisterNHS - Register with NHS. The input is registration request and the output is registration confirmation.

Task 1 (Finding Services):

- Start the Consumption Platform (also known as SPICES).

- Find service “LastFmFriends”. This service retrieves social connections of a given user in the last.fm music social network. It takes the input (a last.fm username) and produces, as output, a list of last.fm users who are connected to the given user.

- Execute the above named service by entering the user name “guillelamb” in the first text field (Person).

- Add all of its operations to your list of favourites.

- Find service “SearchUniversity[1]”.

- Add all of its operations to your list of favourites.

- Find service “RegisterNHS[2]”.

- Add all of its operations to your list of favourites.

*Please rate the following questions by expressing your agreement to each of them on a 5-point scale, where **1= strongly disagree** and **5= strongly agree.**

Rating of the Consumption Platform	
	1 2 3 4 5
Finding services is easy to achieve	Strongly disagree □ □ □ □ □ Strongly agree
Consuming (i.e. executing) services is useful	Strongly disagree □ □ □ □ □ Strongly agree
Consumption Platform responds quickly to my search queries	Strongly disagree □ □ □ □ □ Strongly agree
Consumption Platform is difficult to use	Strongly disagree □ □ □ □ □ Strongly agree
Consumption Platform is reliable	Strongly disagree □ □ □ □ □ Strongly agree
Consumption Platform provides sufficient help instructions	Strongly disagree □ □ □ □ □ Strongly agree
It is easy to learn to find services using the Consumption Platform	Strongly disagree □ □ □ □ □ Strongly agree
Overall, I am satisfied with the Consumption Platform	Strongly disagree □ □ □ □ □ Strongly agree

1, 2 Refer to the description of services in the services’ descriptions sheet to know what it does and what kind of parameters it is expected to have

*Please rate the following service quality characteristics by expressing their relevance to you in selecting the appropriate service on a 7-point scale, where **7= Relevant** and **1= Irrelevant**.

Rating of Service Quality Characteristics	
Functionality: Capability of a service to provide functions which meet stated needs (e.g. Suitability and Security).	7 6 5 4 3 2 1 Relevant |___||___||___||___||___||___||___| Irrelevant
Usability Capability of a service to be understood, learned, used and attractive when utilized (e.g. Learnability and Ease of Use).	7 6 5 4 3 2 1 Relevant |___||___||___||___||___||___||___| Irrelevant
Efficiency Capability of a service to provide appropriate levels of performance when invoked (e.g. Time behaviour).	7 6 5 4 3 2 1 Relevant |___||___||___||___||___||___||___| Irrelevant
Reliability Capability of a service to maintain a specified level of Performance (E.g. Fault Tolerance and Recoverability).	7 6 5 4 3 2 1 Relevant |___||___||___||___||___||___||___| Irrelevant
Service Responsiveness Capability of the service provider to demonstrate willingness assist and provide prompt service.	7 6 5 4 3 2 1 Relevant |___||___||___||___||___||___||___| Irrelevant
Service Reputation Credibility of the service provider.	7 6 5 4 3 2 1 Relevant |___||___||___||___||___||___||___| Irrelevant
Service Assurance Capability of the service provider to inspire trust and confidence.	7 6 5 4 3 2 1 Relevant |___||___||___||___||___||___||___| Irrelevant
Service Empathy Capability of the service provider to deliver individualized attention to the user.	7 6 5 4 3 2 1 Relevant |___||___||___||___||___||___||___| Irrelevant

Task 2 (Editing Templates):

- Start the User Assisted Composition Tool

- From the left menu, browse to Student Services sub-menu and select the template “Student Registration”

- Remove the activity ‘Police Registration’ from the template

- For each other activity in the template **apart** from the activity ‘Medical Registration’, select the appropriate service (s) and rearrange the activities according to the scenario outlined

*Please rate the following questions by expressing your agreement to each of them on a 5-point scale, where **1= strongly disagree** and **5= strongly agree**.

Rating of the User Assisted Composition Tool	
	1 2 3 4 5
User Assisted Composition Tool is easy to use	Strongly disagree □ □ □ □ □ Strongly agree
User Assisted Composition Tool is difficult to navigate	Strongly disagree □ □ □ □ □ Strongly agree
The interface of the User Assisted Composition Tool is pleasant	Strongly disagree □ □ □ □ □ Strongly agree
It is easy to learn to use the User Assisted Composition Tool	Strongly disagree □ □ □ □ □ Strongly agree
User Assisted Composition Tool contains various tools and features that are well integrated	Strongly disagree □ □ □ □ □ Strongly agree
User Assisted Composition Tool provides sufficient help and documentation	Strongly disagree □ □ □ □ □ Strongly agree
People would need the support of technical people to be able to use the User Assisted Composition Tool	Strongly disagree □ □ □ □ □ Strongly agree
I would like to use the User Assisted Composition Tool frequently in the Future	Strongly disagree □ □ □ □ □ Strongly agree
Overall, I am satisfied with the User Assisted Composition Tool	Strongly disagree □ □ □ □ □ Strongly agree

A) Please list the **top (positive) five features** you like about the User Assisted Composition Tool

1. ______________________________

2. ______________________________

3. ______________________________

4. ______________________________

5. ______________________________

B) Please list the **worst (negative) five features** you dislike about the User Assisted Composition Tool

1. ______________________________

2. ______________________________

3. ______________________________

4. ______________________________

5. ______________________________

Task 3 (Editing Process Models):

- Switch to the ‘Process Editor’ from the User Assisted Composition Tool

- Explain in your words what the process model signifies.

- Specify the following condition within the right place in the process model:

> ***Condition:*** you can pay university fee in one of two ways. If you have a sponsor letter you submit it to the university, otherwise you pay the fee using a bank account.

- Bind the appropriate activity in the process model (“Student Registration”) with the service “RegisterNHS”.

- Add an activity ‘JoinNUS’ to the process model. This activity should run parallel to the ‘Medical Registration’ activity.

- Select an activity, e.g. Deposit Funds, from the process model.

- Start the Dataflow editor from the ‘Process Element Properties’ panel – left menu.

- For one input, e.g. ‘TransferDetails’, map it to the output of a preceding service.

- **Save As** your modified template ‘Yourname’.

*****Please rate the following questions by expressing your agreement to each of them on a 5-point scale, where **1= strongly disagree** and **5= strongly agree.**

Rating of the Process Editor	
	1 2 3 4 5
Process modelling is easy to achieve	Strongly disagree □ □ □ □ □ Strongly agree
Process modelling is time consuming	Strongly disagree □ □ □ □ □ Strongly agree
Process modelling is useful	Strongly disagree □ □ □ □ □ Strongly agree
Process Editor is easy to use	Strongly disagree □ □ □ □ □ Strongly agree
I feel confused using the Process Editor	Strongly disagree □ □ □ □ □ Strongly agree
It is easy to learn to model a process using the Process Editor	Strongly disagree □ □ □ □ □ Strongly agree
I feel confident using the Process Editor	Strongly disagree □ □ □ □ □ Strongly agree
There were too many steps required to model a process	Strongly disagree □ □ □ □ □ Strongly agree
It is difficult to understand the modelling notations (e.g. activity, split) used in the Process Editor	Strongly disagree □ □ □ □ □ Strongly agree
It is easy to define conditions using the Process Editor	Strongly disagree □ □ □ □ □ Strongly agree
I can create powerful applications using the Process Editor	Strongly disagree □ □ □ □ □ Strongly agree
I can create applications that facilitate my job activities using the Process Editor	Strongly disagree □ □ □ □ □ Strongly agree
You need to be a programmer to use the Process Editor	Strongly disagree □ □ □ □ □ Strongly agree
Binding services/operations to activities is a difficult task	Strongly disagree □ □ □ □ □ Strongly agree
I know which services/operations to bind to the activities of the process model	Strongly disagree □ □ □ □ □ Strongly agree
The Dataflow editor is difficult to use	Strongly disagree □ □ □ □ □ Strongly agree
I know which input of a service to map to which output of a preceding service in the Dataflow editor	Strongly disagree □ □ □ □ □ Strongly agree
I would like to use the Process Editor frequently in the future	Strongly disagree □ □ □ □ □ Strongly agree
Overall, I am satisfied with the Process Editor	Strongly disagree □ □ □ □ □ Strongly agree

Task 4 (Monitoring Services and Processes):

- Start the ‘Monitoring Module’ from the Analysis sub-menu

- Report the average response time of the service ‘HanivalProductWSService’

………………………………………..

- Report the number of successful requests of the service ‘WebShopService’

………………………………………..

*Please rate the following questions by expressing your agreement to each of them on a 5-point scale, where **1= strongly disagree** and **5= strongly agree.**

Rating of the Monitoring Module	
	1 2 3 4 5
Monitoring services is easy to achieve	Strongly disagree □ □ □ □ □ Strongly agree
Monitoring services is useful	Strongly disagree □ □ □ □ □ Strongly agree
Monitoring Module is easy to use	Strongly disagree □ □ □ □ □ Strongly agree
I feel confused using the Monitoring Module	Strongly disagree □ □ □ □ □ Strongly agree
Monitoring Module provides sufficient help instructions	Strongly disagree □ □ □ □ □ Strongly agree
It is easy to learn to monitor services using Monitoring Module	Strongly disagree □ □ □ □ □ Strongly agree
Overall, I am satisfied with the Monitoring Module	Strongly disagree □ □ □ □ □ Strongly agree

Overall Rating of the SOA4All Studio

*Please rate the following questions by expressing your agreement to each of them on a 5-point scale, where **1= strongly disagree** and **5= strongly agree**.

Rating of the SOA4All Studio	
	1 2 3 4 5
SOA4All Studio is easy to use	Strongly disagree □ □ □ □ □ Strongly agree
SOA4All Studio is difficult to navigate	Strongly disagree □ □ □ □ □ Strongly agree
The interface of the SOA4All Studio is pleasant	Strongly disagree □ □ □ □ □ Strongly agree
It is easy to learn to use the SOA4All Studio	Strongly disagree □ □ □ □ □ Strongly agree
SOA4All Studio contains various tools and features that are well integrated	Strongly disagree □ □ □ □ □ Strongly agree
SOA4All Studio provides sufficient help and documentation	Strongly disagree □ □ □ □ □ Strongly agree
People would need the support of technical people to be able to use the SOA4All Studio	Strongly disagree □ □ □ □ □ Strongly agree
I would like to use the SOA4All Studio frequently in the Future	Strongly disagree □ □ □ □ □ Strongly agree
Overall, I am satisfied with the SOA4All Studio	Strongly disagree □ □ □ □ □ Strongly agree

A) Please list the **top (positive) five features** you like about the SOA4All Studio

1. ______________________________

2. ______________________________

3. ______________________________

4. ______________________________

5. ______________________________

B) Please list the **worst (negative) five features** you dislike about the SOA4All Studio

1. ______________________________

2. ______________________________

3. ______________________________

4. ______________________________

5. ______________________________

Annex B – Evaluation Workbook 2

EVALUATION OF SOA4ALL STUDIO: CONSUMPTION, DISCOVERY, AND ANNOTATION

Participant Information Sheet

The aim of this software testing is to collect your opinion and satisfaction about the Consumption Platform, a platform for finding software services using basic text matching, Discovery Module, a platform for finding software services using annotations, and the WSMO-Lite Editor, an authoring tool for annotating WSDL-based services with semantic descriptions (also called concepts). These tools empower technical users to annotate services with descriptions and search for services based on their semantic annotations.

The current evaluation session consists of four main parts and will take approximately 1 hour to complete.

PART A – Pre-test Interview (~ 5 min)

2. Discuss your experience and opinions about Semantic Technologies and Development

PART B – Interaction with the Consumption Platform (~ 15 min)

4. Interact with the Consumption Platform and report on your initial impressions
5. Complete a set of consumption tasks
6. Rate your satisfaction and report on your opinion towards the Consumption Platform

PART C – Interaction with the WSMO-Lite Editor (~ 25 min)

1. Interact with the WSMO-Lite Editor and report on your initial impressions
2. Complete a set of annotation tasks
3. Rate your satisfaction and report on your opinion towards the WSMO-Lite Editor

PART D – Interaction with the Discovery Module (~ 15 min)

1. Interact with the Discovery Module and report on your initial impressions
2. Complete a set of discovery tasks
3. Rate your satisfaction and report on your opinion towards the Discovery Module

PART A – Prior Knowledge and Experience using the Consumption Platform, Discovery Module and WSMO-Lite Editor

- Full Name of Participant: ……………………………………………………………..

- Please indicate your knowledge and expertise with the Consumption Platform, Discovery Module and WSMO-Lite Editor of SOA4All Studio, where 1 corresponds to poor and 5 corresponds to excellent:

Service Development Tool	I have used it before?	Experience 1 2 3 4 5
Consumption Platform of SOA4All Studio	Yes □ No □	poor □ □ □ □ □ excellent
Discovery Module of SOA4All Studio	Yes □ No □	poor □ □ □ □ □ excellent
WSMO-Lite Editor of SOA4All Studio	Yes □ No □	poor □ □ □ □ □ excellent

PART B – Interaction with the Consumption Platform

A) Initial Impressions

- Start the Consumption Platform (also known as SPICES) from the left hand side

Freely explore and interact with the Consumption Platform for 5 minutes and report on your initial impressions and opinions

B) Tasks

- Find service “SearchUniversity” which given the name of a city its operation returns a list of universities in that city.

- Add all of its operations to your list of favourites.

- Find service “BankAccount” which takes account request as input and returns confirmation of account opening as output

- Add all of its operations to your list of favourites.

C) Rating of Consumption Platform

*Please rate the following questions by expressing your agreement to each of them on a 5-point scale, where **1= strongly disagree** and **5= strongly agree.**

Rating of the Consumption Platform	
	1 2 3 4 5
Finding services is easy to achieve	Strongly disagree □ □ □ □ □ Strongly agree
Consuming (i.e. executing) services is useful	Strongly disagree □ □ □ □ □ Strongly agree
Consumption Platform responds quickly to my search queries	Strongly disagree □ □ □ □ □ Strongly agree
Consumption Platform is difficult to use	Strongly disagree □ □ □ □ □ Strongly agree
Consumption Platform is reliable	Strongly disagree □ □ □ □ □ Strongly agree
Consumption Platform provides sufficient help instructions	Strongly disagree □ □ □ □ □ Strongly agree
It is easy to learn to find services using the Consumption Platform	Strongly disagree □ □ □ □ □ Strongly agree
Overall, I am satisfied with the Consumption Platform	Strongly disagree □ □ □ □ □ Strongly agree

A) Please list the **top (positive) five features** you like about the Consumption Platform

1. ____

2. ____

3. ____

4. ____

5. ____

B) Please list the **worst (negative) five features** you dislike about the Consumption Platform

1. ____

2. ____

3. ____

4. ____

5. ____

PART C – Interaction with the WSMO-Lite Editor

A) Initial Impressions

-Start the ‘WSMO-Lite Editor’ from the WSDL Annotation sub-menu on the left hand side

Freely explore and interact with the WSMO-Lite Editor for 5 minutes and report on your initial impressions and opinions

B) Tasks

-Follow the subsequent two steps to annotate the WSDL of service “SchoolContactXSD.WSDL” using the two ontologies “SchoolContact.rdfs” and “SchoolContactTaxonomy.rdfs”. Both the service WSDL file and ontologies are located in the “BT_Evaluation” repository.

1. Annotate the **data types** of the above named service with concepts from the data ontology “SchoolContact.rdfs”; and

2. Annotate the **operation** ‘checkSMSAvailability’ in the service WSDL with the appropriate concept from the functional classification ontology “SchoolContactTaxonomy.rdfs” to make it discoverable by Functional Classification criterion.

- Export the annotated service to iServe.

C) Rating of WSMO-Lite Editor

*Please rate the following questions by expressing your agreement to each of them on a 5-point scale, where **1= strongly disagree** and **5= strongly agree.**

Rating of the WSMO-Lite Editor	
	1 2 3 4 5
Service annotation is easy to achieve	Strongly disagree □ □ □ □ □ Strongly agree
Service annotation is time consuming	Strongly disagree □ □ □ □ □ Strongly agree
Service annotation is a rewarding task	Strongly disagree □ □ □ □ □ Strongly agree
I know which parts of the service I need to annotate	Strongly disagree □ □ □ □ □ Strongly agree
I know which concepts I need to annotate a service with	Strongly disagree □ □ □ □ □ Strongly agree
WSMO-Lite Editor is easy to use	Strongly disagree □ □ □ □ □ Strongly agree
WSMO-Lite Editor prevents erroneous annotations	Strongly disagree □ □ □ □ □ Strongly agree
People would need the support of technical people to be able to use the WSMO-Lite Editor	Strongly disagree □ □ □ □ □ Strongly agree
WSMO-Lite Editor provides sufficient help instructions	Strongly disagree □ □ □ □ □ Strongly agree
It is easy to learn to annotate services using the WSMO-Lite Editor	Strongly disagree □ □ □ □ □ Strongly agree
I would like to use the WSMO-Lite Editor frequently in the Future	Strongly disagree □ □ □ □ □ Strongly agree
Overall, I am satisfied with the WSMO-Lite Editor	Strongly disagree □ □ □ □ □ Strongly agree

A) Please list the **top (positive) five features** you like about the WSMO-Lite Editor

1. ______________________________

2. ______________________________

3. ______________________________

4. ______________________________

5. ______________________________

B) Please list the **worst (negative) five features** you dislike about the WSMO-Lite Editor

1. ______________________________

2. ______________________________

3. ______________________________

4. ______________________________

5. ______________________________

PART D – Interaction with the Discovery Module

A) Initial Impressions

-Start the 'Discovery Module' from the Discovery sub-menu on the left hand side

Freely explore and interact with the Discovery Module for 5 minutes and report on your initial impressions and opinions

B) Tasks

-Retrieve the service 'SPORTS_NATIONALPARK_SERVICE'. This service does not have any particular functional classification.

-Add its operation to list of favourites

-Retrieve the service 'WebPay' whose functional classification is 'SMS'

-Add its operation to list of favourites

-Using the advanced view, retrieve the service 'SchoolContactService' which:

1. Is annotated using the ontology 'SchoolContact' located at: http://ngwr.labs.bt.com/Ontologies/SchoolContact.rdfs ; and
2. Has an operation whose Input (schoolID) and Output (phoneList) are annotated with the concepts: *School and PhoneList* from the above named ontology respectively.

- Add its operations to list of favourites

C) Rating of Discovery Module

*Please rate the following questions by expressing your agreement to each of them on a 5-point scale, where **1= strongly disagree** and **5= strongly agree.**

Rating of the Discovery Module	
	1 2 3 4 5
Finding services is easy to achieve	Strongly disagree □ □ □ □ □ Strongly agree
Discovery Module responds quickly to my search queries	Strongly disagree □ □ □ □ □ Strongly agree
Discovery Module is easy to use	Strongly disagree □ □ □ □ □ Strongly agree
Discovery Module is unreliable	Strongly disagree □ □ □ □ □ Strongly agree
Discovery Module provides sufficient help instructions	Strongly disagree □ □ □ □ □ Strongly agree
I feel confused using the Discovery Module	Strongly disagree □ □ □ □ □ Strongly agree
It is easy to learn to find services using the Discovery Module	Strongly disagree □ □ □ □ □ Strongly agree
Annotating services using WSMO-Lite Editor enables finding services more quickly	Strongly disagree □ □ □ □ □ Strongly agree
Annotating services using WSMO-Lite Editor enables finding services more accurately	Strongly disagree □ □ □ □ □ Strongly agree
Annotating services using WSMO-Lite is a rewarding task	Strongly disagree □ □ □ □ □ Strongly agree
Overall, I am satisfied with the Discovery Module	Strongly disagree □ □ □ □ □ Strongly agree

A) Please list the **top (positive) five features** you like about the Discover Platform

1. ____

2. ____

3. ____

4. ____

5. ____

B) Please list the **worst (negative) five features** you dislike about the Discovery Module

1. ____

2. ____

3. ____

4. ____

5. ____

Annex C – Screening Questionnaire

Service Annotation and Composition Questionnaire

Q1 Service Annotation and Composition Questionnaire

This questionnaire contains a number of questions designed to collect user familiarity and experience in software and service development using various web service annotation and composition/modelling tools. Questions will be rated on a 5-point likert scale, and it will take only 5-10 minutes to complete. Please answer the questions openly and truthfully.All personal information collected using this questionnaire and study will be treated anonymously and with the strictest confidentiality and will be used for solely the purpose of this research. No other researchers, apart from those leading the study, will be allowed to use the collected data.We will use the answers collected from this questionnaire to invite 30-40 participants (both programmers and non-programmers) to take part in the study. Invited participants will take part in a usability testing which include two main sessions. The first session is a group training session where all participants will attend a 1 hour workshop showing and explaining the software development environment under test along with its features. The second session is a 1 hour individual session where each participant will interact with the software development environment, rate and report on their opinion. Those invited to participate in the study will be awarded with a £20 Amazon voucher as a compensation for their time and participation. Please use Firefox or Internet Explorer to complete this questionnaire. Users of Macintosh Operating System and Safari browser might experience some problems.

Q2 1- Please rate each of the following software experience questions on a 1-5 Likert scale from Extremely poor to Excellent.

	Extremely poor (1)	Below average (2)	Average (3)	Above average (4)	Excellent (5)
My experience in software development using programming languages (e.g. Java, C++) is (1)	❍	❍	❍	❍	❍
My experience in using software development environments (e.g. Eclipse, NetBeans, Microsoft Visual Studio) is (2)	❍	❍	❍	❍	❍
My experience in software design using design notations (e.g. UML) is (3)	❍	❍	❍	❍	❍
My experience in using semantic technologies (ontologies, annotations, reasoning) is (4)	❍	❍	❍	❍	❍
My experience in Business Process Modelling (BPM) is (5)	❍	❍	❍	❍	❍
My	❍	❍	❍	❍	❍

experience in web service composition is (6)					
My experience in process life-cycle management (deployment, monitoring, launching) is (7)	❍	❍	❍	❍	❍

Q3 2- For each of the following platforms, please indicate your level of expertise by checking the appropriate check-box. If you have never used the platform before please check the 'Not Applicable' check-box instead. If there are other tools that you have used but are not listed below please write them down in the empty text fields 'Other'.

	Extremely poor (1)	Below average (2)	Average (3)	Above average (4)	Excellent (5)	Not applicable (6)
Adobe LiveCycle (1)	❍	❍	❍	❍	❍	❍
SAP NetWeaver (2)	❍	❍	❍	❍	❍	❍
IBM WebSphere (3)	❍	❍	❍	❍	❍	❍
ARIS (4)	❍	❍	❍	❍	❍	❍
Ultimus BPM (5)	❍	❍	❍	❍	❍	❍
Savvion Business Manage (6)	❍	❍	❍	❍	❍	❍
Semantic Pipes (7)	❍	❍	❍	❍	❍	❍
JBoss Enterprise SOA Platform (8)	❍	❍	❍	❍	❍	❍

SOA4All Process Editor (9)	❍	❍	❍	❍	❍	❍
Taverna Workflow Management System (10)	❍	❍	❍	❍	❍	❍
WSMO BPMO Editor (11)	❍	❍	❍	❍	❍	❍
Other (12)	❍	❍	❍	❍	❍	❍
Other (13)	❍	❍	❍	❍	❍	❍
Other (14)	❍	❍	❍	❍	❍	❍

Q4 3- For each of the following languages, please indicate your level of expertise by checking the appropriate check-box. If you have never used the platform before please check the 'Not Applicable' check-box instead. If there are other tools that you have used but are not listed below please write them down in the empty text fields 'Other'.

	Extremely poor (1)	Below average (2)	Average (3)	Above average (4)	Excellent (5)	Not applicable (6)
BPEL4WS (1)	❍	❍	❍	❍	❍	❍
BPML (2)	❍	❍	❍	❍	❍	❍
BPSS (3)	❍	❍	❍	❍	❍	❍
OWL, OWL-S (4)	❍	❍	❍	❍	❍	❍
WSDL-S (5)	❍	❍	❍	❍	❍	❍
WSMO (6)	❍	❍	❍	❍	❍	❍
RDF, RDFS (7)	❍	❍	❍	❍	❍	❍
SAWSDL (8)	❍	❍	❍	❍	❍	❍
WSDL (9)	❍	❍	❍	❍	❍	❍
Other (10)	❍	❍	❍	❍	❍	❍

Other (11)	❍	❍	❍	❍	❍	❍
Other (12)	❍	❍	❍	❍	❍	❍

Q5 4- For each of the following semantic annotation tools, please indicate your level of expertise by checking the appropriate check-box. If you have never used the platform before please check the 'Not Applicable' check-box instead. If there are other tools that you have used but are not listed below please write them down in the empty text fields 'Other'.

	Extremely poor (1)	Below average (2)	Average (3)	Above average (4)	Excellent (5)	Not applicable (6)
Protege (1)	❍	❍	❍	❍	❍	❍
WSMO SAWSDL Editor (2)	❍	❍	❍	❍	❍	❍
Semantic Media Wiki (3)	❍	❍	❍	❍	❍	❍
SOA4All WSMO-Lite Editor (4)	❍	❍	❍	❍	❍	❍
Other (5)	❍	❍	❍	❍	❍	❍
Other (6)	❍	❍	❍	❍	❍	❍
Other (7)	❍	❍	❍	❍	❍	❍

Q7 5- Name:

Q8 6- Gender:

❍ Male (1)
❍ Female (2)

Q9 7- Current job / Course of studies:

Q10 8- Indicate the highest level of edUACtion you have completed

❍ High school (1)
❍ Undergraduate (2)
❍ Masters (3)
❍ PhD (4)
❍ Diploma/Certificate (5)
❍ Other (6) ____________________

Q11 9- Rate the following questions

	Never (1)	Rarely (2)	Sometimes (3)	Very Often (4)	Always (5)
How often do you program software applications (1)	○	○	○	○	○
How often do you model software applications (2)	○	○	○	○	○
How often do you annotate software applications (3)	○	○	○	○	○

Q16 10- What are your favourite software / service development languages or platforms?

1- (1)
2- (2)
3- (3)

Q12 11- Rate how interested are you in developing software applications

○ Not at All Interested (1)
○ Not Very Interested (2)
○ Neutral (3)
○ Somewhat Interested (4)
○ Very Interested (5)

Q15 12- Rate how likely are you to develop software applications in the future

○ Very Unlikely (1)
○ Somewhat Unlikely (2)
○ Neutral (3)
○ Somewhat Likely (4)
○ Very Likely (5)

Q17 13- Insert your email address which we will use to contact you for participation in our studies

Email address: (1)

Q18 14- Finally, drag and drop the dates when you are available to take part in the study in the appropriate box. Morning hours are 9am to 1pm. Afternoon hours are 2pm to 5pm.

Morning	Afternoon
______ March (1)	______ March (1)

_______ 14th (2)	_______ 14th (2)
_______ 15th (3)	_______ 15th (3)
_______ 16th (4)	_______ 16th (4)
_______ 17th (5)	_______ 17th (5)
_______ 18th (6)	_______ 18th (6)
_______ 21st (7)	_______ 21st (7)
_______ 22nd (8)	_______ 22nd (8)
_______ 23rd (9)	_______ 23rd (9)
_______ 24th (10)	_______ 24th (10)
_______ 25th (11)	_______ 25th (11)
_______ 28th (12)	_______ 28th (12)
_______ 29th (13)	_______ 29th (13)
_______ 30th (14)	_______ 30th (14)
_______ 31st (15)	_______ 31st (15)
_______ April (16)	_______ April (16)
_______ 4th (17)	_______ 4th (17)
_______ 5th (18)	_______ 5th (18)
_______ 6th (19)	_______ 6th (19)
_______ 7th (20)	_______ 7th (20)
_______ 8th (21)	_______ 8th (21)