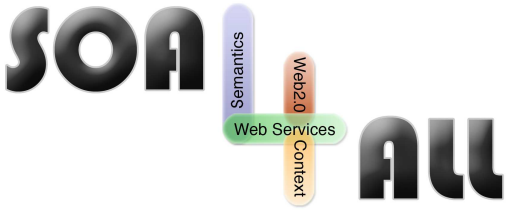


Project Number: **215219**
Project Acronym: **SOA4All**
Project Title: **Service Oriented Architectures for All**
Instrument: **Integrated Project**
Thematic Priority: **Information and Communication Technologies**

D8.7. Evaluation Prototype

Activity:	3- Use Case Activities	
Work Package:	8- BT W21C	
Due Date:		M37
Submission Date:		01/04/2011
Start Date of Project:		01/03/2006
Duration of Project:		38 Months
Organization Responsible of Deliverable:		UNIMAN
Revision:		1.0
Authors:	Abdallah Namoune (UNIMAN) Nikolay Mehandjiev (UNIMAN) Ali Owrak (UNIMAN) Usman Wajid (UNIMAN) Sandra Stinčić Clarke (BT) Alistair Duke (BT) John Davies (BT)	
Reviewers:	Sven Abels (TIE) Georgi Pavlov (SAP)	

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	23/2/2011	ToC and Structure outlined	Abdallah Namoune
0.2		ToC updated	
0.3	25/2/2011	First draft with initial contributions	
0.4	3/3/2011	Change of TOC, Initial BT contribution	Sandra Stinčić Clarke
0.5	7/3/11	Edits of BT contribution	John Davies
0.6	28/2/2011 18/3/2011	Usability evaluation results Evaluation Workbook, ServiceWave Paper, References Additional corrections	Abdallah Namoune
0.7	18/3/2011	Updated following internal comments	Abdallah Namoune, Ali Owrak, Usman Wajid
0.8	18/3/2011	Additional corrections	Nikolay Mehandjiev
0.9	22/3/2011	Final draft ready	Alistair Duke
	28/3/2011	Review	Sven Abels, Georgi Pavlov
1.0	31/3/2011	Address reviewer comments & minor fixes	Alistair Duke, Sandra Stinčić Clarke

Table of Contents

List of Figures

List of Tables

Glossary of Acronyms

Acronym	Definition
API	*Application Programme Interface*
CeBP	*Communication-enabled Business Process*
D	*Deliverable*
ReST	*Representational State Transfer*
RDF	*Resource Description Framework*
RDFS	*Resource Description Framework Schema*
SMS	*Short Messaging Service*
SPARQL	*SPARQL Protocol and RDF Query Language*
SPICES	*Semantic Platform for the Interaction and Consumption of Enriched Services*
SWEET	*Semantic Web sErvice Editing Tool*
WSDL	Web Services Description Language
WSMO	Web Services Modelling Ontology
XML	eXtensible Markup Language
XSLT	XML Stylesheet Transformation

Executive summary

In work package 8, the deliverable D 8.7 Evaluation Prototype presents a case study of the SOA4All studio building on the current BT Ribbit infrastructure, which provides a set of Web accessible Telco services, and leverages SOA4All research and technology to allow end-users to access, use and create services based on BT's capabilities (such as Voice, SMSs, etc). The validation of this case study has been conducted through three evaluation studies that encompass *economic, technical and usability perspectives*.

The aim of the *economic evaluation* (conducted through two workshops) was to test the attitudes of users to the ideas behind SOA4All in terms of user-led service development, and to identify their main concerns as well as efficient ways to represent service development to end users. Feedback received from the participants of the first workshop demonstrated a positive balance between benefits and risks which is considered a necessary pre-requisite for uptake of such user-driven service composition. A second workshop was undertaken with the premise to identify the utilisation of service-based technologies and seek business models for exploiting the commercial benefits of new value-added services. The outcome demonstrated a high willingness to use the SOA4All platform as a medium to generate profits and create business opportunities as well as to buy and sell services.

The *technical evaluation* study undertaken looks at how the SOA4All studio satisfied the WP8 functional and non-functional requirements. In particular this focuses on the degree to which the technology would be able to support a BT platform allowing BT's partners and customers to build communications focussed service mashups and business processes. The evaluation is an outcome of the experience gained in applying the technology to the scenarios developed in the work package. The findings are that the technology generally supports the requirements which further development required in the support of RESTful APIs and the level of maturity of the tools.

The *usability evaluation* aimed to ascertain how well SOA4All Studio satisfies the objectives associated with its role within the WP8 Use Case and to measure user acceptance of the SOA4All design features within a holistic end-to-end scenario of use. The study revealed that the annotation tool was perceived by the participants as innovative whilst the visual appearance of the studio was praised for its simplicity and high level of abstraction.

1. Introduction

1.1 Introductory explanation of the deliverable

This deliverable primarily reports on and concludes the evaluation efforts conducted within Work Package 8. The evaluation studies undertaken focus on the usability, technical, and economic aspects of SOA4All Studio within the context of the WP8 Case Study. The general aim of these evaluation exercises is to ascertain user acceptance of 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 project and the results gained from these evaluations. We systematically started with the economic evaluation to ensure business goals of the project are satisfied by conducting discussion focus groups with potential business users where different business models were explored. Following this we carried out a summative usability evaluation with a total of 6 representative users to inspect the usability of the SOA4All Studio. Finally, we performed a technical evaluation to ensure that the technical requirements outlined in previous deliverables have been satisfied.

1.2 Purpose and Scope

The purpose of this document is to present a case study detailing the application of the Ribbit infrastructure within the SOA4All studio and validate the suitability of SOA4All Studio to the representative end users.

This document

- describes the evaluation methods, techniques and rationale behind them
- presents the results of the evaluation workshops to seek business models for exploiting the commercial benefit of new value-added services that can be realised by using SOA4All Studio and results
- presents the results of technical evaluation of the WP8 requirements
- details the usability scores and user opinion about the SOA4All Studio

1.3 Structure of the document

The document structure is as follows:

- Section 2 presents the BT Use Case in addition to the business, functional and usability requirements.
- Section 3 describes the aim and resulting outcome of a series of workshops undertaken to assess the economic evaluation of the SOA4All studio.
- Section 4 presents a technical evaluation, reporting on how the SOA4All along with its functionalities and features satisfies Work Package 8 functional and non-functional requirements.
- Section 5 describes the usability evaluation carried out, focusing on how well the objectives of the SOA4All Studio were satisfied in addition to determining the user acceptance of the studios design features within the context of WP8 Use Case.

1.4 Methodology

The approaches behind our evaluation studies are in alignment with the overall evaluation strategy of SOA4All as described in the Description of Work and also in the D2.5.1 deliverable. The three types of evaluation used three different evaluation methods:

(a) The economic evaluation was based on focus group discussions and thematic analysis.
(b) The technical evaluation was based on reflective study and external testing of the functionality developed in SOA4All against the requirements formulated by WP8.
(c) The usability evaluation focused on the results of a summative usability testing using the SOA4All Studio within the context of WP8 Use Case. This was preceded by earlier formative evaluation exercises shared with other work-packages, reported in earlier deliverables. The final summative evaluation investigated user acceptance of the SOA4All Studio and summarised their usability scores after users have performed a holistic end-to-end scenario of use of SOA4All Studio, as envisioned by WP8.

Further details of all these methods are given in the respective sections for each approach.

2. BT Use Case and Requirements

The telecommunications sector is rapidly changing from both a business and technology perspective. Telecommunication operators, or telcos, have encountered an urgent and pressing need to change if they want to maintain their traditional strong position in the market, whilst increasing numbers of web and/or software based companies (also known as webcos) are delivering services over the Internet. Various Telcos have recognised the need of moving into the so-called 'Telco 2.0' world, thus bridging the gap between traditional telecoms and the Web 2.0 world. Exposing their capabilities thru APIs to the wider developer community and enabling creation of applications with third party services, new opportunities emerge constantly as current market offers variety of non-Telco companies offering Telco-type services. The D8.5, Telco 2.0 Recommendations, explains how telcos can exploit their existing assets and strengths in new ways by moving from a 1-sided to a 2-sided business model, thereby tapping into new revenue streams from novel products and services. Technologically, making this change involves exposing telco capabilities via open APIs on the web in an 'open services ecosystem.' The relevance and role of SOA4All project technology is clear: a major goal of SOA4All is to make it easier for non-experts to access, combine and use services, thus assisting in the uptake of the open service ecosystem.

BT has identified that Internet based communication is important in the vision of a 21st century Telco business and therefore released a web based API for accessing a variety of telephony services over the web, named Web21C SDK. Following that, BT acquired the Ribbit, open platform for multi-protocol communication, primarily focused on a market of voiceware applications and services. BT's Ribbit platform is providing next generation services on top of its all IP-based 21st Century Network (BT 21CN).

The required transformation of a telco proceeds along a number of identified axes: network transformation, cost transformation, process transformation, IT transformation, product transformation and business model transformation. The first three of these are necessary but not on their own sufficient; the last three are the changes which have the potential to deliver increased revenues and improved customer experiences in the Internet world.

All of the mentioned resulted in the recognition of need for a platform that will include a service creation environment providing non-software developers with the ability to create processes using an intuitive interface. This should be coupled with access to a service repository and discovery mechanism with semantic capability to allow identification of services that will meet the requirements as well as a facility to deploy the service to a scalable run-time infrastructure. Fulfilling those criteria could greatly reduce the time and cost involved in developing and launching new applications.

BT's Case Study looked into a number of scenarios that would be of interest to different user groups. Two prototypes were created, the first was focussed on supporting consumers or 'prosumers' in building service mashups while the second was focussed on supporting businesses in rapidly building and monetising business propositions to address an indentifiable market.

The first prototype used a number of Ribbit services and combined them with services from a number of other providers such as Facebook and the BBC to create a location- and weather-aware friends' meeting service. SOA4All technology was used to overcome some of the problems that limit the uptake of the Ribbit services, primarily the technical knowledge required and familiarity with programming languages such as PHP or JavaScript. A detailed description of the prototype and subsequent findings was provided in deliverable D8.4.

The second prototype, described in D8.6, implemented a B2B scenario showing how SOA4All technology can enable telecommunications companies and their partners to generate value through building and offering innovative services to their customers in an agile way. SOA4All technology can be used to provide a platform that enables services to be

developed and updated with greatly reduced software development effort. In addition, uptake of the services can be widely increased, services can be operated in a scalable, efficient manner and all of the listed will contribute to reduced time-to-market and costs and a higher quality of services. Specifically, a service ("Offers4All") encompassing a set of business processes was developed to allow retailers and other offer providers to easily and flexibly create and manage offers targeted at particular sets of users using the most appropriate communications channels for those users.

Defining aforementioned scenarios provided a set of requirements that are described in the following sections. These reflect the requirements introduced in Section 2 of Deliverable 8.1, "Web21C Requirements".

2.1 Business requirements

We first describe some general business objectives for BT, explaining how SOA4All technology has the potential to contribute to them and then set out some more specific objectives related to the uptake of exposed web services.

Potential Contribution to General Business Objectives

Concept To Market

A big challenge for BT is to reduce time to market for new products and services in the highly competitive Telco and ISP business. SOA4All technology offers the opportunity to build working products from a number of services quickly, with the associated management and OSS infrastructure. A reduced reliance on software engineering skills would enable a much wider set of propositions to be launched at reduced cost.

Trouble To Resolve

Service Level Agreements and Quality of Service are important considerations for businesses, as they need some levels of guarantee about the deployed services they are paying for. Defining and managing SLAs in a distributed SOA infrastructure on the web is an important consideration that needs to be addressed by SOA4All technology.

Critical Mass

To ensure success of the platform, a critical mass of services needs to be created. This should be both in terms of the discrete services that are available (and annotated) such that rich mashups can be created and in terms of the number and quality of created mashups and apps that are available in order to attract end users. To get that, it is important to satisfy functional requirements for the platform (see Section 2.1.2) and to choose the correct business model which will attract more developers and service providers onto the platform and ultimately users.

Specific Objectives related to Web Services

We can identify a number of more specific requirements relating to the exploitation of telecommunications capabilities exposed as web services as follows:

- Encourage greater uptake of exposed communications services by providing tools to enable easier use of the services;
- Provide tools with a focus on end users with limited technical experience;
- Create a community of users, to encourage collaboration and innovation in creating new telco-centric applications;
- Create an infrastructure to allow a third party businesses to resell BTs exposed services, providing support in design and management of the thirds party services;

• Increase overall use of BT's exposed services, and hence increase revenue.

2.2 Functional Requirements

Import and link to Industry standard ontologies

The telecommunications industry has a number of standardised models and ontologies that it will be necessary to use when describing services semantically and linking to other services. There is a requirement that industry standard ontologies described in OWL or RDF can be imported and used within the SOA4All platform. This will enable existing services and newly created mashups to be described in terms of these ontologies and also to allow developers and end users to search for services and mashups using concepts from the same set of ontologies.

Import and mark-up third party Web Services

In the case study scenarios, a combination of BT services (such as Voice or SMS) and third party services (such as billing or recommendation services) will be used to create a final product. This requires that third parties can bring their own services into the system and mark them up semantically, perhaps then building a product by composing them with BT services. This process should be possible without detailed knowlede of ontologies or web service description languages. It should be possible using a standard web browser without the need to install specialist software. It should incorporate an easy to use interface employing, for example, a drag and drop facility to ease the process of annotation.

Service Discovery

Annotation of the services should allow subsequent search on the functional and non-functional aspects of service descriptions. Support for service ranking and help with design time selections based on user context should be provided.

Service Composition

A service composition tool should allow discovered services to be combined into a process with support for the definition of data and control flow. It should provide suggestions for compatible services in web service compositions and automated or semi-automated assistance with the linking of services in a composition. The tool should also provide the facility to deploy compositions (allowing them to be executed) and to export/share compositions with other users.

Support the use of RESTful services

The RESTful appoach has emerged as the dominant method for publishing APIs on the Web (vs WSDL which although being dominant within enterprises has not been widely adopted on the Web). [1] found that 74% of Web APIs were RESTful vs. only 15% that were SOAP based. Ribbit provides a RESTful API[1] as well as abstractions from the API using various programming languages. In order to exploit the momentum in the number of REST services on the Web, the SOA4All toolset needs to fully support them i.e. allow them to be described, discovered, included in compositions and executed.

Fault handling of Service Failures and Error reporting

Business critical services need a suitable level of fault handling and error management for service compositions. If a particular service in a composition fails, it will be necessary to record this and understand if any corrective action needs to be undertaken (such as roll back, or service replacement).

[1] http://docs.ribbit.com/restful-api

Monitoring of process state and Quality of Services (QoS) for service compositions

For critical applications there is the necessity to give some level of guarantees to end customers about the Quality of Service they can expect, which is detailed in a service level agreement (SLA). There is therefore the requirement to monitor service executions and assess the QoS (such as response time) of services, in order to ensure they meet SLA.

Integrated Toolset

The SOA4All Studio aims to provide an integrated environment allowing all aspects from service annotation to execution to be addressed. The Studio should provide a consistent 'look and feel', support user preferences in a consistent manner and allow service descriptions and compositions to be shared seamlessly between the tools.

2.3 Usability Requirements

The usability exercise is described in Section 5. The aim is to test the SOA4All tools with prospective users and in particular the claim that software development effort can be reduced with the use of SOA4All. The exercise is based around a scenario where a user is required to build a communication-centric process using services provided by Ribbit and others.

This scenario envisages creation of a composite service to contact parents as regards school closures due to severe weather conditions (e.g. snow). This could be a service created for a single school by a 'prosumer' akin to the first scenario of our case study or it could form part of an offering from a commercial organisation offering services to local authorities and supporting many schools which is more akin to the Offers4All scenario. The key aspect is that services would be developed upon a BT platform, enabled by SOA4All technology and would include both BT and 3rd party services and APIs.

In the scenario, the user is required to annotate a web service and then use assisted modelling in SOA4All Studio and create a process which finds parents' contact number and notifies them of school closures via text messages or phone calls.

The process is concerned with contacting people e.g. parents about school closures due to snow and is intended to follow the following sequence.

1. Parents can be contacted via SMS or phone calls
2. If all parents have entered they prefer to be contacted by SMS then that method should be used exclusively.
3. If some parents have entered they prefer phone calls:
    a. Set up a call to a list of phone numbers and upload an audio file to explain the closure reasons
    b. Parents will then be able to hear the audio message when the call is answered.

The evaluation exercise was carried out with 6 BT employees with a variety of software development expertise and web services knowledge. The participants were first provided with a tutorial including the background and aims of the SOA4All project and instruction on how to use each of the tools that were under evaluation. They were then required to carry out a number of tasks related to the scenario. This consisted of annotation of a web service using the WSMO-lite editor and upload of these descriptions to the iServe repository, simple search and annotation of services using the consumption platform and finally creation of a composition to satisfy the scenario requirement using the process editor. Throughout the exercise the audio and interaction with the PC were recorded. Participants were required to complete a post-test interview and complete a feedback form. A full description of the study together with results is provided in Section 5.

3. Economic Evaluation

Whilst creating prototypes, several consultations with potential users have been undertaken, confirming the business need and helping the refinement of the scenarios.

3.1 May 2009: BT Stakeholders Consultation/June 2009: Semantic Week

The aim of this workshop was to test the attitudes of the SOA4All target users to the ideas behind SOA4All in terms of user-led service development, and to identify the main concerns as well as efficient ways to represent service development to end users. The three main themes of discussion were (a) the business models enabled by the idea of opening up service development (including discovery, annotation and composition) to third parties and non-professionals, (b) risks and benefits from such user-driven service development (UDSD) for the target users and for their organisations and (c) discussing alternative representations and mechanisms to support UDSD. The workshop report is covered in Deliverable 8.4 (Web 21c Prototype v1), with major benefits highlighted as follows:

- The potential for saving time and reusing development effort
- The opportunity to empower the creativity of third parties
- The possibility to exploit a larger number of niche profit generation opportunities compared to conventional service provision models

Perceptions of risks were focused on the following issues:

- Transparency of data use/ Privacy concerns
- Unexpected multiplier effect from community-style interactions
- Trust
- Brand protection

Participants' answers demonstrated a positive balance between benefits and risks, which is considered a necessary pre-requisite for uptake of such user-driven service composition.

The set of recommendations arising from analysis of this consultation was:

(a) Consider providing a two-tier interface, where experienced users are provided with detailed and powerful control-flow mechanisms for controlling service compositions, whilst novice users are provided with simple templates and asked to select pre-discovered services for each aspect of the template.
(b) Provide examples of services composed by novice users, and a sandbox environment for other novice users to test and modify these examples.
(c) Support the community dimension of the user development activities, using it for sharing community-specific solutions and templates, for feedback on actual services used, and for general support and awareness.
(d) Provide facilities for template and service evolution, using a combination of (a) and (c) approaches above.

The subsequent developments took these into consideration and created a specialist module called user assisted composition, which provided the user-facing second tier of the interface, and was based on templates acting as service examples and a repository of knowledge shared between the community of users.

3.2 December 2010: ESTC

The WP8 workshop at ESTC 2010 intended to perceive the uptake of service-based technologies and seek business models for exploiting the commercial benefit of new value-added services that can be realised by employing service-based technologies. The event was organised as a focus group and the two main themes of discussions were (a) the risks

and benefits of using SOA4All tools by internet users (b) business models for web services.

The perceptions of risks were focused on the following issues:

- Complexity of tools and difficulty of using the tools by users with non-technical background. In this respect, participants expressed concerns that business people will find it difficult to model and annotate web services in SOA4All process editor since such aspects are normally dealt by the technical team or software developers.
- IP and brand protection, who owns what, who operates the service-based applications and who gets accountable when the application stops working. Participants were concerned that the users will not be able to find out the source of the problem in a service-based application that brings together services from various service providers.
- Sustainability, provide examples and support for a sustainable application development approach to discourage the development of applications that are quick to develop but difficult to maintain.

On the other hand, the major highlighted benefits were as follows:

- The potential of quickly developing service-based applications and reusing development efforts.
- Adding value to traditional business models

During the focus groups participants also discussed the social dimension of using web services and issues related to establishing trust in service-based applications. Participants suggested that some of the trust related issues can be alleviated by outsourcing trust in a way of cloud model that looks after the integrity and reliability of those provisional services at the level of service level agreement.

The other topics discussed in the focus groups centred on the need to find business models to exploit the commercial benefit of web services and service-based tools. In general participants agreed on introducing payment schemes for the use of service-based applications. Various charging schemes and revenue generation models were brought up in the discussions on the commercial incentives of service-based technologies. Suggestions for suitable payment schemes included charging per usage or transaction which was considered safe and manageable compared to organizational licences.

In regard to the business aspects of SOA4All project, service consumers agreed to subscribe to premium services and use web services as an alternative to traditional services. However, participants also expressed concerns that due to lack of clarity over the composition of service-based applications the commercial benefit of service-based technologies might not be leveraged in every sector such as the banks and other financial sector institutions e.g. stock exchange.

Also, focus groups provided the rating of a set of question items to evaluate the SOA4All project. The first part of the questionnaire was designed to capture the participants' role in relation to web services. The second part of the questionnaire aimed at capturing the participants' software and service development experience. The third part of the questionnaire captured the users' attitudes towards various business aspects of the SOA4All services and platform. In this respect, participants' ratings revealed high willingness to use the SOA4All platform as a medium to generate profits and create business opportunities as well as to buy and resell services. Participants' willingness to pay for SOA4All services, the platform, and incurred technical support was rather neutral. However on the positive side, workshop participants did not disagree to pay for the SOA4All tools which support the former point.

3.3 Summary of Economic Evaluation

The evaluation found that there is large potential for SOA4All to create value via widening the scope of people able to participate. This is partly based on the degree to which the tools are able support users through reduced complexity. Recommendations were made with regard to this which were addressed in the project. The risk that users find the tools too complex remains and this should be considered as the tools mature and are exploited.

The ownership and responsibility associated with service compositions is not clear and the evaluation exercise identified issues regarding brand, trust, privacy and quality of service. These aspects would require attention in market trials involving the SOA4All technology.

4. Technical Evaluation

In the technical evaluation section we report on how the SOA4All along with its functionalities and features satisfy the WP8 functional and non-functional requirements in the context of identified requirements. In each case an indicative rating is provided (1-5, where 1 indicates the requirement has not been met at all and 5 indicates the requirement has been fully met)

4.1 Business Requirements

4.1.1 Concept To Market

An important aspect of the BT Case Study, and in particular the second prototype, has been to support the development of Communications-enabled Business Processes (CeBP), where BT along with its partners / customers embeds communications services into the business processes of its partners / customers in order to improve the likelihood of those business processes succeeding, which in turn allows e.g. improved customer service, reduced costs, increased sales, etc. as interactions with the end customer (i.e. those of BT's partners and customers) are improved and made more efficient.

Currently, CeBP as a concept is under development in BT with pilots and trials developed by BT software development teams using the Agile software development methodology[2]. We have engaged closely with our colleagues in BT Strategy who are leading this initiative with the aim of determining the level of fit of SOA4All to their requirements. The Offers4All scenario can be seen as an example of a CeBP so our work on that has enabled us to assess this. In addition we also met with the head of the development team and one of the members of that team was a participant in the Usability Evaluation exercise.

Clearly the CeBP initiative has a requirement to allow BT's customer and partners to develop their own CeBP without relying on BT to develop them for them. This would mean that partners would need to provide software development skills or that a platform providing technology akin to that of SOA4All would be required. BT would still of course need to ensure that partners were sufficiently trained to use the platform and that they were able to annotate their own services. Alternatively BT would provide the annotation skills and effort allowing partners to create the processes.

The SOA4All technology, although not mature enough to allow a trial to take place, does support the requirements in reducing Concept to Market time in a CeBP setting.

Rating: 4

4.1.2 Trouble To Resolve

The monitoring framework in SOA4All allows the status of individual services and APIs to be determined. This is an important feature in terms of supporting a deployed service or business critical process since it allows problems to be identified and corrective action to be taken. Further work is required to allow Service Level Agreements for composite services to be defined and managed. This would include raising alarms when discrete services fail, providing mechanisms to support dynamic binding to alternative services in the event of failure and escalation of alarms when Service Level Agreements are threatened or breached for discrete and composite services and processes.

Rating: 3

4.1.3 Critical Mass

As stated above, a BT platform supported by SOA4All technology would require significant

[2] http://en.wikipedia.org/wiki/Agile_software_development

effort in terms of bootstrapping to ensure that sufficient services and APIs were available in order to attract partners and customers to develop CeBP or service mashups. It is difficult to say whether this would be sufficient to create the critical mass necessary for the platform to thrive. The technology provided by SOA4All does increase the likelihood, specifically via its support for service annotation, the recommendation mechanisms and linked data support. In order to fully test this it would be necessary to carry out a market trial once the maturity of the tools has increased to allow this to take place.

Rating: 3

4.2 Functional Requirements

4.2.1 Import and link to Industry standard ontologies

In WP8, an RDFS taxonomy was created describing the nature of currently available telco and telco-related services. The taxonomy is based upon the results of the extensive survey carried out on current activity in the area (see Deliverable 8.5). The taxonomy, containing around 80 concepts is available at:

http://ngwr.labs.bt.com/Ontologies/TelcoAPITaxonomy.rdfs

In addition, an RDFS ontology describing the data requirements of the APIs of telco services was created using the Ribbit API as a guide. This is available at:

http://ngwr.labs.bt.com/Ontologies/TelcoAPI.rdfs

In prototype version 2, a domain ontology that describes the Offers4All scenario was developed, focussing on the data requirements of the various services involved. The ontology contains about 25 classes and 50 properties and includes the Nepomuk contact ontology[3] which is used to handle the contact information associated with offer providers and users and the W3C Geo ontology[4] which is used to handle location information. This is available at:

http://ngwr.labs.bt.com/Ontologies/Offers4All.rdfs

These ontologies can be imported into SOA4All tools such as WSMO-lite. They can then be browsed and concepts within them can be used to describe service operations and data elements. The SWEET tool which supports the description of RESTful services, although not allowing the ontologies to be imported and browsed, does support the input of URIs from these ontologies when annotating service operations and data elements.

When finding services the support for browsing using scenario specific ontologies is not so good. Both the discovery tool and the consumption platform use the Service Finder taxonomy rather than allowing the user to select their favoured ontology. However, the discovery tool does allow queries to include URIs from any ontology.

Rating: 4

4.2.2 Support of RESTful services

To achieve prototype version 2, semantic service descriptions for the Ribbit API were created. Since support for RESTful services within SOA4All has generally lagged behind that for WSDL service, the decision was taken to wrap the Ribbit Java API as WSDL services and then use the project's WSDL support to annotate these, build compositions and execute them. This involved using WSMO-Lite to create annotations and the grounding editor to create transformations between the data ontologies and the XML Schema Definitions (XSD)

[3] http://www.semanticdesktop.org/ontologies/nco/

[4] http://www.w3.org/2003/01/geo/

required by the WSDL services (allowing lifting and lowering to take place at runtime).

Latterly, support for some kinds of RESTful services has somewhat improved. SOA4All does not support the Ribbit RESTful API since this uses an extended version of the OAUTH authentication process. However, a simplified RESTful wrapper was created in WP8 around the Ribbit Java API. This removes the OAUTH requirement, instead supporting authentication via a username and password provided in the body of the HTTP request. These APIs can be described using the SWEET tool and uploaded to iServe. Unfortunately there is no support for grounding RESTful services in SOA4All (i.e. no model to define different types of grounding and thus no editor to allow a grounding to be created) so it was necessary to hand craft XSLT transforms and URI templates for lowering and SPARQL construct queries for lifting. These can be attached to the inputs and outputs in SWEET before uploading the complete annotation to iServe.

Since the iServe repository, discovery service and process editor operate at the level of semantic service descriptions, they do not care whether the underlying service is WSDL or RESTful and thus are able to support RESTful services.

With service execution, work was carried out to enable the Ribbit RESTful wrappers to be executed by the execution engine, thus providing support for that particular kind of RESTful service (i.e. with GET and POST methods, where inputs are provided as JSON objects in the HTTP request or as URL parameters and where outputs are provided as JSON objects in the HTTP response). At the end of the project it has thus been possible to annotate, store, discover, compose and execute RESTful services albeit with no support for grounding and for a small subset of the RESTful services that are available on the web.

Rating: 2

4.2.3 Import and mark-up third party Web Services

This requirement has been achieved for WSDL services as these are fully supported in SOA4All by the WSMO-lite tool and grounding editor. It is unlikely that arbitrary RESTful services that are found on the web would be suitable for use in SOA4All in its current state. Workarounds do exist if there is a particular need to support services from a 3rd party provider e.g. if they are seen as strategically important. This would involve wrapping the existing interface in a similar manner to that described in Section 4.2 but this involves software development. In addition the grounding for these APIs would need to be hand-crafted.

Rating: 3

4.2.4 Service Discovery

Service Discovery in WP8 has focussed on the facility provided by the consumption platform rather than the more powerful form of discovery which is possible using the Service Discovery interface. The SPICES consumption platform allows users to perform simple keyword search or browse via a service categorisation. Returned services can be chosen based upon their descriptions, the recommendations or specific comments provided by other users. This form of discovery would appear to be sufficient for a telco-specific platform.

Rating 4

4.2.5 Service Composition

The process editor is able to support the creation of the processes that have been defined for the scenarios developed in WP8. Specific requirements that were put forward by WP8 such as the support for iterating over a data structure output of an activity by running the subsequent activity multiple times have been included (this is the loop option in the dataflow editor). At the time of writing it is not possible to deploy and execute a completed composition from the process editor since the work on LPML to BPEL translation is not

complete.

Rating: 4

4.2.6 Fault handling of Service Failures and Error reporting

The monitoring framework allows service failures to be reported which is a useful first step. Further work would be required to support (automated) handling of faults similar to that provided in Service Delivery Platforms. However, this is outside the scope of SOA4All.

Rating: 3

4.2.7 Monitoring of process state and Quality of Services (QoS) for service compositions

As stated above, the monitoring framework enables the status of individual services to be monitored which allows QoS to be measured in terms of service availability. Further work would be required in order to determine other QoS measures such as reliability or QoS for service compositions to be determined.

Rating: N/A

4.2.8 Integrated Toolset

The SOA4All studio provides an integrated environment for service annotation, publishing, discovery, composition and execution. It provides a facility to launch the studio components and a user profile facility allowing user actions e.g. service favourites to be accessed across the components. As stated above the ability to complete a composition and then execute it within the studio has not been completed at the time of writing. The ability to run compositions that have been built is essential is allowing users to rapidly build and deploy processes and mashups.. In addition, it is not currently possible to publish compositions as services in their own right.

Rating: 2

4.3 Summary of Technical Evaluation

The requirements review and ratings in the technical evaluation show that in general the project has enabled the majority of requirements to be met at a good level of support. In particular, the support for service annotation and execution for WSDL services is good and the support for publishing, discovery, and composition is good in general, Areas where further support is required is for REST services, particularly with regard to service grounding and execution. In addition, integration between the composition tool and execution needs attention.

Regarding the business requirements, the technology clearly supports these aims but these would need validation in technical and / or market trials with users but the maturity of the tools and support for functional requirements would need to improve for these to take place. Ongoing work in BT around the CeBP initiative should allow this to occur in 2011/12.

5. Usability Evaluation

As opposed to the earlier formative evaluations reported elsewhere, which focused on inspecting the SOA4All Studio, identifying the most critical usability issues and proposing design recommendations to improve the Studio (e.g. Annex B - ServiceWave Paper (Dec 2010)), this WP8 summative evaluation focuses on:

- evaluating how well the objectives of the SOA4All Studio have been satisfied within the context of Work Package 8 Use Case; and
- gauging user acceptance of the general SOA4All design features and ideas in the form of feedback and opinion data within the context of WP8 Use Case.

Participants interacted with the SOA4All Studio to implement a process model which enables sending school notifications to parents e.g. closure due to bad weather conditions using WP8 services (e.g. Ribbit services). This user testing employed the think-aloud protocol [2] to gather user opinion and questionnaires to capture user satisfaction in the form of usability scores.

5.1 Aim and objectives of the study

We start by enumerating the objectives of the SOA4All Studio and its constituent tools: WSMO-Lite Editor, Consumption Platform, and Process Editor. 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 constituent tools of SOA4All Studio the goals for each tool are enumerated as follows:

WSMO-Lite Editor aims to enable end users to:

- Obj1: Add semantic descriptions, in the form of ontological concepts, to wsdl-based descriptions of web services to allow for dynamic invocation and discovery of the resulting semantic web services
- Obj2: Explore differing service elements, such as data types and operations and their parameters
- Obj3: Publish semantic service annotations in an online repository which can be later retrieved and used in the subsequent steps in the service development lifecycle (i.e. Process modelling)

Consumption Platform aims to enable end users to:

- Obj1: Retrieve and find target services shared through SOA4All using a simple and easy to use service search engine
- Obj2: Explore the different operations and their corresponding details, and view user comments and ratings of the available services
- Obj4: Execute and inspect the behaviour of the available services
- Obj3: Add the desired services to the list of favourites for the follow-up stages

Process Editor aims to enable end users to:

- Obj1: 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

- Obj2: Use the desired services from their list of favourites and bind them to the appropriate activities of the process model
- Obj3: Specify dataflow between different services via a dataflow editor

5.2 Participants' Background

A total of six users took part in the evaluation of WP8 use case. All participants had programming and development experience using a variety of programming languages such as Java, but no modelling expertise was reported. Only one participant mentioned having experience using semantic technologies and developing domain ontologies.

5.3 Evaluation Procedure and 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,
(c) and rating phase.

A detailed description of each phase is presented below.

5.3.1 Training Phase

During the training phase, representative users attended a 2-hour group presentation which explained and demonstrated (1) various features and aspects of each of the SOA4All tools and (2) Ribbit services. This included a tutorial demonstrating one service development example of moderate complexity. Users were then invited to comment and ask questions about what they have seen. In detail, the training phase encompassed the subsequent steps:

1- A group presentation and training providing detailed explanation on how to use the service development tools under test;
2- A post-training session for participants to ask questions about the service development tools; and
3- Giving out hands-on documentation / slides about the tools after the tutorial.

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

5.3.2 Development Phase

At the beginning of this session users were instructed to explore and interact with the SOA4All studio alongside its constituent tools for 5 minutes with the aim of familiarising them with the available functionality and getting their initial impressions. Immediately after that, participants were given an end-to-end WP8 scenario description and instructed to develop a service-based application covering three aspects of the development lifecycle: annotation, consumption, and modelling. In this usability testing we opted the think-aloud protocol [2] 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 performed by the participants 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
- **Interaction with the SOA4All Studio and its tools:** participants interacted with the "SOA4All Studio" and completed a set of diverse development tasks (annotation, consumption, and modelling) in order to fulfil the test scenario.

- **Post-test Questionnaire and Interview:** participated rated different usability dimensions of the SOA4All Studio and its tools. Subsequently, participants discussed their experience and opinions about the SOA4All Studio in a post-test interview.

5.3.2.1 Scenario and Task Descriptions

Prior to conducting the usability testing, we conducted a set of pilot studies to ensure that the development tasks are well described and easily understood by the participants; otherwise it would not be possible to perform the tasks correctly. We refined the description of the tasks so that they do not detail the fine-grained steps required to solve a particular development task. A balance should be struck between clarity and specificity. We also created development tasks that are complex enough to enable full interaction with at least the core functionalities and features of the Studio. The tasks included annotation, consumption and modelling tasks. The annotation tasks instructed the participants to annotate the data types and one operation of the SchoolContact service, and export the resulting annotations to iServe. 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 contacting parents in regard to school closures, binding services to the right activities in the process model, and defining dataflow between two services. Refer to the appendix for the list of tasks and their description.

The overall description of the test scenario is as follows: “Your goal is to contact parents as regards school closures due to severe weather conditions (e.g. snow). For this you need to develop a process model which allows you to find parents’ contact number and notify them of school closures via text messages or phone calls. To realise this application you will use the SOA4All Studio which allows you to annotate, discover, and model a process”.

5.3.3 Rating phase

At the end of the usability testing, participants were invited to rate the differing features of the tools by filling out a questionnaire and report their feedback in a debriefing interview. In this final phase participants:

- rated their development experience, usability of and satisfaction towards each of the SOA4All tools (WSMO-Lite Editor, Consumption Platform, Process Editor). They completed a paper-based questionnaire containing question items about usability and preferences such as “ease of use … etc” to which they expressed their the degree of agreement on a 1 to 5 rating scale, where (1 =disagree and 5 =agree)
- reported their final feedback and opinions about the SOA4All Studio in a de-briefing interview

The WP8 evaluation workbook covering the details of aforementioned phases (i.e. development and rating phases) is included in Annex A.

5.4 Analysis method

User feedback and opinions 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 on 1-5 rating scale. Self-reported data and interviews were initially recorded, transcribed into excel sheets and analysed using the thematic analysis method; a general research technique for analysing qualitative data where themes emerge from user opinion data.

Two Human Factors' experts (both of them conducted the user studies) watched the videos of user interaction with the Studio, listened to users' vocal discourse and transcribed the conversations into an excel file. This step is important as it allowed the experts to familiarise themselves with the data. Each expert worked independently by going through, preparing and analysing the transcripts for each task category (i.e. annotation, consumption, modelling). The experts coded the opinion data and identified corresponding themes following the thematic analysis method [3]. Once finished, the two experts sat together and discussed the emerged themes; they removed the reoccurring themes and refined/paraphrased some themes for better accuracy.

5.5 Results

We divide the results section into three parts focusing on (1) how the aforementioned objectives have been fulfilled, (2) comparison of the current usability scores against those of an earlier formative usability evaluation, and (3) improvements made to from a Studio version of early formative test to the current version.

5.5.1 User acceptance of the SOA4All Studio and its tools

Table 1. User Acceptance of the WSMO-Lite Editor

Annotation	Evidence from opinion data	Fulfilled
Obj1	-The concept of annotating WSDL-based services with semantic information was well received and perceived as easy to understand by the users. For instance, one participant commented: "*the annotation is straightforward; I understand what it is all about*". -Users liked the ability to supplement service parts with semantic descriptions and described this aspect as "*novel and innovative*". They emphasised that this aspect is missing in existing tools. We have noticed that all participants were able to match the correct concepts to the data types of the service SchoolContact. However, they were, in certain occasions, unsure which concepts should go with the operation checkSMSAvailability. This result is anticipated as they were not familiar with the particulars of the used ontology. -Users indicated that there is no need to master programming languages or semantics knowledge to use the WSMO-Lite Editor because most of the low-level details are hidden from the end user. Although our users (except one user) were unfamiliar with semantic technologies and the training session was rather short, they were able to carry out the annotation tasks without serious problems. -Users liked the way by which (drag and drop) concepts can	

	be mapped to service descriptions; they just selected, dragged and dropped the appropriate concepts to the service parts. Users positively commented on the usability of the WSMO-Lite tool: "*usability point of view is ok; drag and drop feature is a good interface for delivering what you want*". -Users have largely relied on the names of the concepts and service parts to make the mapping.	
Obj2	-User interaction with the WSMO-Lite Editor demonstrated that the users used the right hand pane to view and explore the various elements of the service SchoolContact by expanding the trees and branches of the service in order to find the data types and operation requiring annotations. No major problems were reported by the participants in regard to this point.	✓
Obj3	All users were able to successfully publish their semantic annotations and view them on the iServe repository. However, participants indicated that their annotations were not prominent within the published online files.	✓

Table 2 lists user behaviour and comments as to whether the consumption goals were satisfied. The Consumption Platform acts as a service search engine and execution platform aiming to empower end users to search for services using plain search queries, test and execute service, and add them to list of favourites.

Table 2 . User Acceptance of the Consumption Platform

Consumption	Evidence from opinion data	Fulfilled
Obj1	-All users were able to use the service search engine quite comfortable as they are quite accustomed to using search engines in their daily interaction with software and web applications. They inserted the name (or text cues) of their target services (e.g. SchoolContact, RibbitCalls, MediaCalls) in the search text field and pressed the 'search' button. The search engine displayed a list of available services from which the users selected the correct services. Users described the Consumption platform as "*a simple concept which allows finding and adding services to the list favourites*".	✓
Obj2	-Every time users opened a particular service, they inspected its operations along with their input and output parameters. These details were very useful for learning and identifying the functionality of the service and therefore selecting the most appropriate services to their process model. Users chose the highly rated services for their process model which demonstrates the utility of the	✓

	service rating.	
Obj3	This task (i.e. execution of services) was not tested in this evaluation.	Not tested
Obj4	All users have successfully managed to add the required operations to the list of favourites without or with very minimum help from the experimenter. Users simply pressed the "AddOperationToFavourite" button which added the target service to the list of favourites. Simultaneously a pop up message was shown at the bottom of the screen to confirm task success. The list of favourites, on the left hand pane, was constantly accessed by the users to view their favourite services.	✓

The Process Editor aims to empower users to model the logic and data flow of their service-based applications. In our case, users were able to sketch their process model by adding different visual notations (e.g. activities) and wiring them together.

Table 3. Average completion time for annotation, consumption, and modelling task

Process Modelling	Evidence from opinion data	Fulfilled
Obj1	-Users described the process editor as "*clear, straightforward and easy to use*" which might be attributed to users' familiarity with existing process modelling tools. Users indicated that the process editor provides "*a simple way of putting services together*". -The video analysis of the user interaction showed that users were able to model the process model using the 'activity', split and merge notation and connectors. On average, each participant used 5 activities and 1 split. Only one participant used the merge notation. Users performed the modelling activities without having to write a single line of code or deal with arcane technical details. This reflects the high level of abstraction implemented within the Studio aiming to hide technical complexity from the users. -Users praised the drag and drop feature which allowed them to create connections between notations. -In regard to the suitability of the editor for non-technical audience, our users strongly felt that no programming expertise is needed to use the Process Editor but some understanding of logic would be helpful.	✓
Obj2	-All participants performed the binding task successfully; they navigated to their list of favourites in the process editor, browsed the available operations, and bound the appropriate operations to the activities of their process	✓

	model.	
Obj3	-The concept of dataflow specification was perceived as clear, and users were able to understand it quickly but the layout of the dataflow was viewed as rather unintuitive. Users proposed to remove the middle area as it is more natural if input and output relationships are specified directly.	

In regard to user performance, we calculated the average time spent by each participant for completing each development task. The results are reported in the table below showing that the consumption task was the shortest task to perform (13 mins), followed by annotation (23 mins) and modelling task (32 mins).

Table 4. Average completion time for annotation, consumption, and modelling task

	Average (minutes)	Standard Deviation
Annotation	23	6.30
Consumption	13	6.40
Modelling	**32**	**5.70**

5.5.2 Usability scores

Upon completion of the development tasks participants rated, on a 5-point rating scale where 1 signifies disagree and 5 signifies agree, various usability dimensions of the SOA4All Studio and its tools. Overall average rating was calculated and showed that the SOA4All studio is relatively easy to use (mean= 3.16, std= 1.17) and that users are motivated to use it in the future (mean= 3.33, std= 1.03). However, users did not agree that the studio is easy to navigate (mean= 2.66) and felt the navigational structure should be improved.

Similarly, users have also rated a range of features of the WSMO-Lite Editor, Consumption Platform, and Process Editor. The test scores were favourable as these tools were perceived as easier to use than the previous versions. The average usability scores of a preceding formative usability test of the Studio (performed in November 2010) have increased in comparison to the scores of this summative test, as illustrated in table 5.

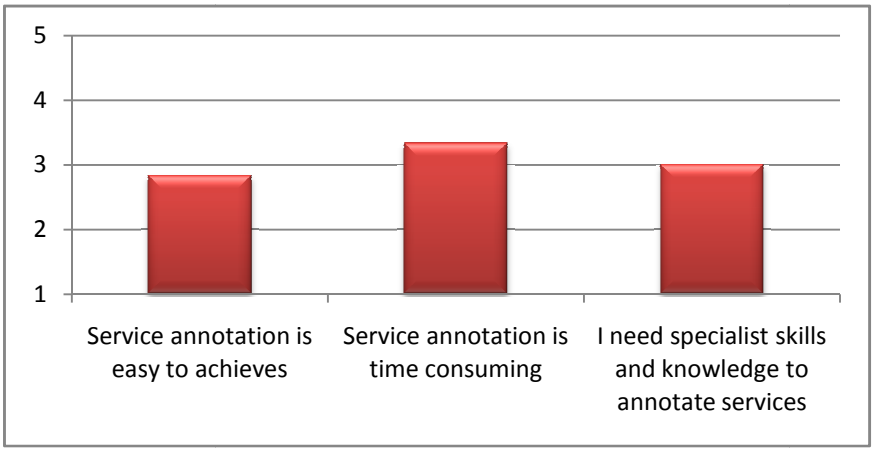
Table 5 . Average usability scores of the WSMO-Lite Editor, Consumption Platform, and Process Editor

	Formative usability scores	Summative usability scores
WSMO-Lite Editor …		
is easy to use	2.72	**3.33**
prevents erroneous annotations	2.63	**2.83**
provides sufficient instructions in case of	2.5	**2.5**

problems		
Consumption Platform …		
is easy to use	2.81	**3.83**
responds quickly to search queries	3.18	**4**
using the consumption platform, I can easily retrieve the services I need	2.90	**3.83**
Process Editor …		
is easy to use	**3.5**	**3.66**

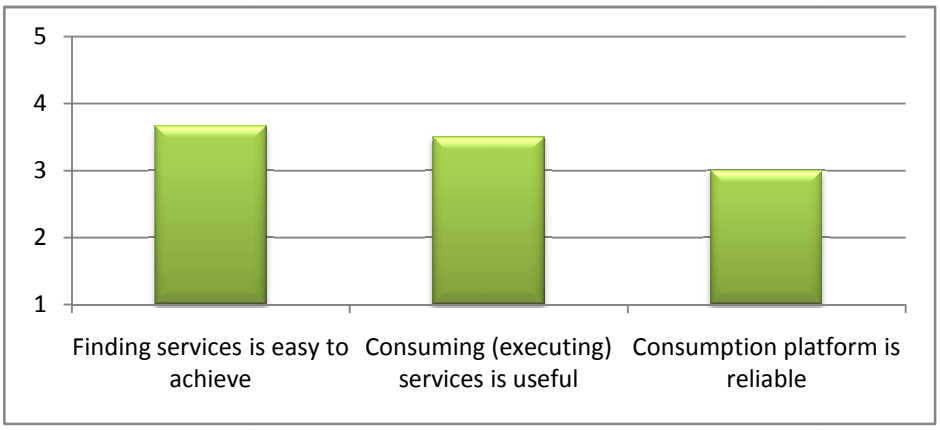
Further usability questions that are related to the WSMO-Lite Editor received ratings revolving around the neutral value (3). As such users did not feel that the annotation tasks required a large amount of time to complete them nor did they think that specialist skills are needed to annotate WSDL-based services (see Figure 1).

Figure 1. Average rating of various usability dimensions of the WSMO-Lite



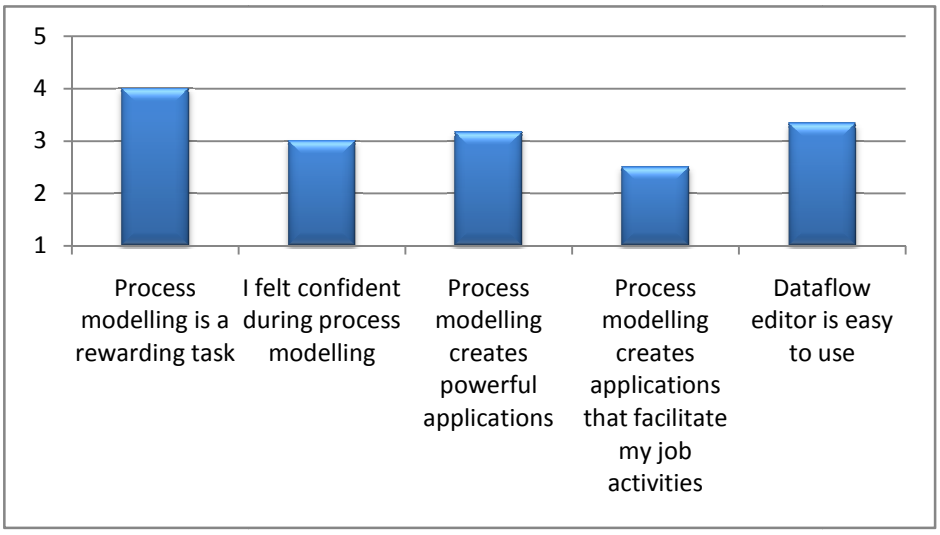
In respect to the Consumption Platform, users found the task of finding services rather easy to achieve (mean= 3.66, std= 0.51). Whilst users were positive that consuming services is a useful activity, they were neutral to the reliability of the consumption platform which is expected since they had very little interaction with it (Figure 2).

Figure 2. Average rating of various usability dimensions of the Consumption Platform



Users agreed that process modelling is a rewarding task (mean = 4) but were less convinced that it will help them create applications that facilitate their job activities (mean = 2.5). This might be attributed to the fact that the execution part of the Studio was not tested in this evaluation, and users were unable to see their resulting applications running. All remaining usability scores averaged to 3 or more as depicted in figure 3.

Figure 3. Average rating of various usability dimensions of the Process Editor



5.5.3 Improvements made from formative test (conducted in Dec 2010) to summative test (conducted in Jan 2011)

Following WP7 formative evaluation (results reported in Deliverable D7.7), SOA4All Studio developers have implemented a number of user recommendations to eliminate the most persistent design problems (in table 6) and improve the overall usability of the SOA4All Studio. Subsequently, the current summative test showed that none of the below negative points were raised during user interaction with the SOA4All tools reflecting the improvements made.

Table 6. The most reoccurring usability problems in a formative test

Tool	Usability issue from formative test	Fixed	Reported in summative test
WSMO-Lite Editor	-Users spent a long time to find their desired service and ontology due to the random way in which repositories and services are listed	✓	No
	-Users were unable to read feedback messages and some UI elements due to the poor colour contrast used	✓	No
Consumption Platform	-When no services were found by the search engine no information was reported to the user which confused them about the status of their request	✓	No
	-Users were unable to read pop up messages as they disappeared quickly	✓	No
	-Users complained that they could not see, upon entering new search terms, the new list of services since they were listed within a newly-created window at the bottom of the screen	✓	No
Process Editor	-Users were unable to understand the error messages and felt they were inconsistent	✓	No
	-Users were unable to save their process models as the system failed to respond	✓	No
	-Users were unable to read the error messages owing to the poor colour scheme used	✓	No

5.5.4 Recommendations for future work

Although this evaluation is of summative nature, there is always room for further work. We therefore devised a set of design suggestions for advancing the SOA4All studio which emerged either during their interaction with the Studio or in the post-experiment interview. Most of the suggestions focused on two primary aspects: (1) system help and documentation and (2) navigation.

- Help and documentation of the Studio: one of the primary concerns of our participants was the lack of definitions, instructions and help in the different parts of the Studio. At present the Studio does not have a user or how-to manual. Definitions should include descriptions of ontological concepts and service elements which will empower end users to map the concepts to the right service parts in the WSMO-Lite Editor more confidently and correctly. Moreover, clear definitions of the various visual notations, such as merge and split, and their functionality will enable designer to select the

appropriate notations and thus design their model in the right way. Provisioning of comprehensive examples of semantic service annotations and process models will help users annotate services and design process models more successfully through learning and reuse. Implementation of intelligent wizards to guide users through the steps they have to perform to fulfil the different tasks of the studio would be beneficial and increase user performance.

- Navigation: some participants stated that navigational structure within the differing tools could be improved by re-arranging the menu items and using representative and self-explanatory names. If used, tool-tips would also explain and elaborate ambiguous menu items.

5.6 Summary of Usability Evaluation

In the usability evaluation, we found good coverage of main objectives behind most of the main SOA4All Studio modules. The subjective (e.g. user ratings) and objective measures (e.g. time-on-task) demonstrated that end users, despite their poor knowledge and experience using SOA technologies and the short training session, were able to complete the annotation, consumption, and modelling tasks without major problems. All three tools (WSMO-Lite Editor, Consumption Platform ,Process Editor) were rated as relatively easy to use (3.33, 3.83, 3.66 on a 5-point Likert scale respectively), and users expressed willingness to use them in the future especially the annotation aspect of the tool which was regarded as innovative. The visual appearance and graphical nature of the Studio was amongst the features praised by the participants alongside simplicity and high level of abstraction which hides the low level technical details from the users.

6. Conclusion

This deliverable has reported on a set of evaluation studies aiming to establish how well SOA4All results fit to the requirements and scenarios of usage developed within WP8. The overall aim of WP8 developments was to evaluate the role of SOA4All within the next generation of Ribbit infrastructure where the process of discovering, integrating, using and sharing BTs services could be undertaken in a more effective manner. The studies focus on three aspects of such a "fit": economic evaluation, technical evaluation and usability evaluation.

In terms of the economic evaluation, we concluded feasibility and desirability of the SOA4All concept and technology, providing good pre-conditions regarding its take-up and use. In terms of technical evaluation, we concluded that SOA4All Studio satisfies most of the technical functionality requirements stipulated by WP8. Better support for REST and increased maturity of the tools are outstanding requirements that would need to be addressed before the technology could be taken to a trial with partners or customers.

Finally, in terms of summary usability evaluation, we found good coverage of main objectives behind most of the main SOA4All Studio modules. The subjective (e.g. user ratings) and objective measures (e.g. time-on-task) demonstrated that end users, despite their poor knowledge and experience using SOA technologies and the short training session, were able to complete the annotation, consumption, and modelling tasks without major problems. All three tools (WSMO-Lite Editor, Consumption Platform ,Process Editor) were rated as relatively easy to use (3.33, 3.83, 3.66 on a 5-point Likert scale respectively), and users expressed willingness to use them in the future especially the annotation aspect of the tool which was regarded as innovative. The visual appearance and graphical nature of the Studio was amongst the features praised by the participants alongside simplicity and high level of abstraction which hides the low level technical details from the users.

7. References

[1] Musser, J. (2010). Glue Conference, Denver, USA, May 2010. http://www.slideshare.net/jmusser/pw-glue-conmay2010

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

[3] Braun, V. and Clarke, V. (2006) Using thematic analysis in psychology. Qualitative Research in Psychology 3 , pp. 77-101

Annex A – Evaluation Workbook

EVALUATION OF SOA4ALL STUDIO

Participant Information Sheet

The aim of this software testing is to collect your opinions about the SOA4All Studio, an integrated service development environment, for opening up software service development to people without technical and programming skills and making it as easy as creating a personal blog or a customised Facebook page.

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

PART A – Pre-test Questionnaire and Interview (~ 10 min)

1. Complete a Participant Background Form
2. Discuss your experience and opinions about Software and Service Development

PART B – Interaction with the SOA4All Studio (~ 1 hour)

1. Interact with a service development environment “SOA4All Studio” and complete a set of development tasks

PART C – Post-test Questionnaire and Interview (~ 20 min)

1. Rate your service development experience and satisfaction towards the SOA4All Studio
2. Discuss your experience and opinions about the SOA4All Studio

PART A – Participant Background Questionnaire

Please state your agreement with the following questions by ticking the appropriate box (✓)

Personal Details:

Age: _____ Gender: M☐ F☐

Current job / Course of Studies: _____

Background:

Please indicate the highest level of education you have completed.

☐ High school ☐ Undergraduate ☐ Diploma/Certificate ☐ Masters ☐ PhD

Others, please specify: _____

Which of the options below best describes the level of your IT training?

☐ none ☐ self-taught ☐ Introduction to office software or similar

☐ non-IT degree ☐ non-IT degree with significant IT training

☐ IT-focused degree or significant vocational training such as Microsoft Certified Professional (MCP), etc.

☐ other

How often do you develop service-based applications?

☐ daily ☐ weekly ☐ monthly ☐ less often ☐ never

What are your favourite service development languages or systems?

1._____

2._____

3._____

Please indicate your knowledge and expertise with the 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
SOA4All Studio	Yes ☐ No ☐	poor ☐ ☐ ☐ ☐ ☐ excellent

PART B – Interaction with the SOA4All Studio

Initial Impressions

Please freely explore and interact with the SOA4All Studio for 5 minutes and report your initial impressions and opinions

Scenario Description

Your goal is to contact parents as regards school closures due to severe weather conditions (e.g. snow). For this you need to develop a process model which allows you to find parents' contact number and notify them of school closures via text messages or phone calls. To realise this application you will use the SOA4All Studio which allows you to annotate, discover, and model a process.

Task 1:

- Start the WSMO-Lite Editor

- 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** from the sequence elements with concepts from the data ontology "SchoolContact.rdfs".

2. Annotate the **operation** 'checkSMSAvailability' in the service WSDL with concept(s) from the functional classification ontology "SchoolContactTaxonomy.rdfs" to make it discoverable by Functional Classification criterion.

- Export the annotated service to iServe.

- Access the iServe browser using the link (http://iserve.kmi.open.ac.uk/browser.html) and ensure that your annotated service exists in iServe**.**

Task 2:

- Start the Consumption Platform

- Find service "SchoolContact[5]"

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

1-3 Refer to the description of this service in the services' descriptions sheet to know what it does and what kind of parameters it is expected to have

- Find service “RibbitCalls[6]”

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

- Find service “RibbitMedia[7]”

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

Task 3:

- Start the Composer

- Build a process model (“Contact parents”) which allows you to contact parents and inform them of school closures due to severe weather conditions. The notification shall be sent following these conditions:

1. Parents can be contacted via SMS or phone calls
2. If all parents have entered they prefer to be contacted by SMS then that method should be used exclusively.
3. If some parents have entered they prefer phone calls:
    a. Set up a call to a list of phone numbers and upload an audio file to explain the closure reasons
    b. Parents will then be able to hear the audio message when the call is answered.

- Manually bind two activities of the process model (“Contact parents”) with the appropriate services/operations.

- Save your process model as WP8_yourname

- Open the complete process model “BT_Evaluation.pe”

- Discuss how the complete survey model differs from your solution

- Select an activity, e.g. WaitForCallSetup, from the complete process model

- Start the Dataflow editor from the ‘Process Element Properties’ panel

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

Descriptions of services and their operations

SchoolContact service:

Operations:

checkSMSAvailability – given the id of a particular school, checks if all parents are contactable by SMS or not. The input is a string indicating the school id, the output is '1' if all parents are SMS contactable and'0' if not

getSMSNumbers – given the id of a particular school, return a list of SMS enabled phone numbers. The input is a string indicating the school id, the output is a list of strings containing the SMS numbers

getPhoneNumbers – given the id of a particular school, return a list of voice only phone numbers. The input is a string indicating the school id, the output is a list of strings containing the phone numbers

RibbitCalls service:

Operations:

createCall – setup a call between two parties. The inputs are the username and password for the ribbit service and the source and destination phone numbers, the output is the id of the call

waitForCallSetup – pauses until all parties have answered the call. The inputs are the username and password for the ribbit service and id of the call, the output is the status of the call.

playMediaToCall – plays a previously uploaded audio file to the call participants. The inputs are the username and password for the ribbit service and the location of the audio file (on the Ribbit server), the output is the status of the play.

RibbitMedia service:

Operation:

uploadMedia – upload an audio file ready for playing to a phonecall. The inputs are the username and password for the ribbit service, the source of the audio file (as a URL) and the folder and filename where the file should be uploaded to on the Ribbit server.

RibbitMessages service:

Operation

sendSMSMessage – sends an SMS message to a phone number. The inputs are the username and password for the ribbit service, the source and destination phone numbers and the title and content of the message

PART C – 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= disagree** and **5= agree**

Rating of the SOA4All Studio	
Annotation	1 2 3 4 5
Service annotation is easy to achieve	disagree ☐ ☐ ☐ ☐ ☐ agree
Service annotation is time consuming	disagree ☐ ☐ ☐ ☐ ☐ agree
Service annotation is a rewarding task	disagree ☐ ☐ ☐ ☐ ☐ agree
I know which parts of the service I need to annotate	disagree ☐ ☐ ☐ ☐ ☐ agree
I know which concepts I need to annotate a service with	disagree ☐ ☐ ☐ ☐ ☐ agree
I need specialist skills and knowledge to annotate services	disagree ☐ ☐ ☐ ☐ ☐ agree
The annotation editor is easy to use	disagree ☐ ☐ ☐ ☐ ☐ agree
The annotation editor prevents erroneous annotations	disagree ☐ ☐ ☐ ☐ ☐ agree
The annotation editor provides sufficient instructions in case of problems	disagree ☐ ☐ ☐ ☐ ☐ agree
Consumption	1 2 3 4 5
Finding services is easy to achieve	disagree ☐ ☐ ☐ ☐ ☐ agree
Consuming (i.e. executing) services is useful	disagree ☐ ☐ ☐ ☐ ☐ agree
Consumption platform responds quickly to my search queries	disagree ☐ ☐ ☐ ☐ ☐ agree
Consumption platform is difficult to use	disagree ☐ ☐ ☐ ☐ ☐ agree
Consumption platform is reliable	disagree ☐ ☐ ☐ ☐ ☐ agree
Using the consumption platform, I can easily retrieve the services I want	disagree ☐ ☐ ☐ ☐ ☐ agree
Process Modelling	1 2 3 4 5

Process modelling is easy to achieve	disagree ☐ ☐ ☐ ☐ ☐ agree
Process modelling is time consuming	disagree ☐ ☐ ☐ ☐ ☐ agree
Process modelling is a rewarding activity	disagree ☐ ☐ ☐ ☐ ☐ agree
The process editor is easy to use	disagree ☐ ☐ ☐ ☐ ☐ agree
I feel confused using the process editor	disagree ☐ ☐ ☐ ☐ ☐ agree
I feel confident using the process editor	disagree ☐ ☐ ☐ ☐ ☐ agree
There were too many steps required to model a process	disagree ☐ ☐ ☐ ☐ ☐ agree
It is difficult to understand the modelling notations (e.g. merge, human task) used in the process editor	disagree ☐ ☐ ☐ ☐ ☐ agree
It is easy to express conditions using the process editor	disagree ☐ ☐ ☐ ☐ ☐ agree
I can create powerful applications using the process editor	disagree ☐ ☐ ☐ ☐ ☐ agree
I can create applications that facilitate my job activities using the process editor	disagree ☐ ☐ ☐ ☐ ☐ agree
You do not need to be a programmer to use the process editor	disagree ☐ ☐ ☐ ☐ ☐ agree
Binding services/operations to activities is a difficult task	disagree ☐ ☐ ☐ ☐ ☐ agree
I know which services/operations to bind to the activities of the process model	disagree ☐ ☐ ☐ ☐ ☐ agree
The Dataflow editor is easy to use	disagree ☐ ☐ ☐ ☐ ☐ agree
I know which input of a service to map to which output of a preceding service in the Dataflow editor	disagree ☐ ☐ ☐ ☐ ☐ agree
I am willing to use the process editor to model processes in the future	disagree ☐ ☐ ☐ ☐ ☐ agree
Overall Rating of the SOA4All Studio	1 2 3 4 5
Overall, the SOA4All Studio is easy to use	disagree ☐ ☐ ☐ ☐ ☐ agree
Overall, the SOA4All Studio is difficult to navigate	disagree ☐ ☐ ☐ ☐ ☐ agree
Overall, the SOA4All Studio provides sufficient help and documentation	disagree ☐ ☐ ☐ ☐ ☐ agree

I am willing to use SOA4All Studio in the future	disagree ☐ ☐ ☐ ☐ ☐ agree
The SOA4All Studio is a good substitute to traditional software development environments such as Eclipse	disagree ☐ ☐ ☐ ☐ ☐ agree

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

1. _____

2. _____

3. _____

4. _____

5. _____

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

1. _____

2. _____

3. _____

4. _____

5. _____

Annex B - ServiceWave Paper (Dec 2010)

A Comparative Study: Service-based Application Development by Ordinary End Users and IT Professionals

Abdallah Namoun, Usman Wajid, Nikolay Mehandjiev

Manchester Business School, The University of Manchester, Manchester, M13 9SS, UK
{*firstname.lastname*}@mbs.ac.uk

Abstract. Service-Oriented Architecture enables users, both ordinary end users and IT professionals, to be part of the development cycle of interactive service-based systems in order to fulfil their desired needs. In this paper we explore and compare the mental model of two different categories of users towards the idea of "service composition by end users". Participants' responses are concluded from 5 separate focus groups, including a total of 64 participants. Results have shown that both groups of users are highly interested in the composition of service-based systems; however, privacy and security concerns and technical complexity of current approaches and service composition environments hinder the diffusion of service-based technologies among users. In this respect, this paper proposes a preliminary model of service composition uptake by end users and discusses user views and requirements to facilitate service composition.

Keywords: EUD, risks and benefits, user study, services, service composition.

1 Introduction

Current trends in Software Engineering, Human Computer Interaction, and Service and Components Research emphasise the need to create software artefacts that are easy to develop and customize [6]. In our view, computer users can be categorised according to their jobs into two major groups, a small proportion of expert software developers whose primary job is to create sophisticated software artefacts and a large proportion of ordinary end users who use those artefacts in support of their jobs. The former group includes people who are skilful programmers and problem-solving experts whereas the latter group includes people who are programming and modelling inept but they maybe domain experts. Thus, the research challenge is to equilibrate this imbalance by empowering ordinary end users, especially domain experts, to uptake software development activities via suitable tools so they can easily develop and customise software artefacts based on their goals and changing requirements.

To address this challenge, service-oriented architecture (SOA) offers suitable means of loosely coupling software services to produce augmented service-based applications through the so called process of "service composition" [1]. Composing services requires specifying what and how services are executed in a composition and how data is passed between them using complex composition languages and tools. This process is time consuming and requires considerable modelling and programming knowledge even for experienced programmers. In this respect, it is vital to simplify the composition process for both groups of users, firstly by offering user-friendly service composition tools, and secondly by reducing the programming efforts and activities usually associated to software development. Such research promises to promote the reuse of web services, especially by ordinary end users.

When creating a user-friendly interface for a service composition system, we need to consider user expectations regarding the trade-off between the costs of learning new tools and the benefits they expect to get from using them. The balance between costs and benefits is likely to differ for different groups of users and different target domains (e.g. [9], [14]), yet we believe that identifying user attitudes and expectations towards service composition is a key to predicting successful uptake of service composition [9], [14] and [15].

In this paper, we endeavour to capture and contrast mental models of both ordinary end users and IT professionals about web services and service composition, their perception of end user development risks and benefits, and requirements in order to build user-friendly composition tools that account for the differences and special needs of each user category. This comparison produced differing requirements for end user composition.

2 Composition of Service-based Systems

Service Composition is broadly supported by two main approaches: workflow-based scripting of service components, and AI-based automatic composition of service components reasoning with pre- and post-conditions. Further details are available elsewhere [4] and [13].

Professional programmers are supplemented with specialized composition languages (BPML, BPEL4WS, WSCDL … etc) to construct service-based systems. However, developing composite services using text and XML editors is complex, error-prone, and time consuming. Therefore, several visual representations for service composition and interaction have been proposed with the aim to make the composition more user-friendly (e.g. Zenflow [7]). However, most of them are *ad hoc*, i.e. they use technology-led representations and metaphors, which are not derived from user studies. Only a few of them have been evaluated in terms of usability and cognitive effectiveness. For example, Lets Dance [15] has been evaluated using the framework of Cognitive Dimensions [2], but iterative testing and enhancement have not been documented in the related references. Another example is Vitabal WS [5], which is a version of an earlier visual language tuned to the needs of web service composition. It has been evaluated using the cognitive dimensions framework, yet it targets experienced web service developers only.

Opening up service use and development to people who are not professional programmers (i.e. end users) requires the delivery of user interfaces that are task-oriented rather than technology-oriented; that is, they should be tuned to the expected skills and foreseen tasks of our target users. Activities such as service construction and composition involve non-trivial problem-solving in a context called End User Development (EUD) [6, 15]. EUD research provides an insight into the type of software interfaces and motivational factors likely to support end user activities.

Sutcliffe *et. al.* [14] see the trade-off between expected benefits and learning costs as a main determinant of uptake of an EUD tool by users. Risks and benefits of EUD have been used to underpin a number of quantitative studies in concrete domains, aiming to elicit the likelihood of uptake for end user development ideas in the specific context of that domain (e.g. [9]). The workshops reported here are examples of an application of this approach to the target domain of SOA4All[8], an EU-funded project that aims to open up service composition to everyone.

Several research studies have attempted to explore end user perception of software development, for instance: McGill and Klisc [8] argue that end users in the Internet domain are aware of the associated risks and benefits and thus it is crucial to involve them in the development of Internet application development approaches to minimize risks. Due to the difficulty of learning traditional programming languages, Myers *et al* [11] report on a number of studies aiming to elicit understanding of how people think about a particular task, natural programming languages and design environments that support the way end user developers are thinking. The generated data about user behaviour is used to build intuitive and usable programming environments. More recently, Namoune *et. al* [12] summarised potential

[8] www.soa4all.eu

problems of service composition showing that end users have difficulty connecting various services and understanding specialized service-related terms such as: operations, parameters etc. Overall, review of existing literature reveals that research in end user development of service-based systems is very rare and most studies are in their infancy.

3 Procedure and Materials

To acquire a better understanding of end users' perception about service composition and their likelihood for uptake of application development, we have conducted *five separate* focus groups; *three focus groups* with students and University staff and *two focus groups* with IT specialists who come from IT companies and research labs from various EU-countries). The focus groups included a total of 64 participants. Of those, 35 represented the mix expected by general consumers of services (range 19 to 40 years, mean 26 years) and further 29 were selected as representative of the specialist application fields like telecoms, where the majority did have IT expertise (range 23 to 60 years, mean 35 years).

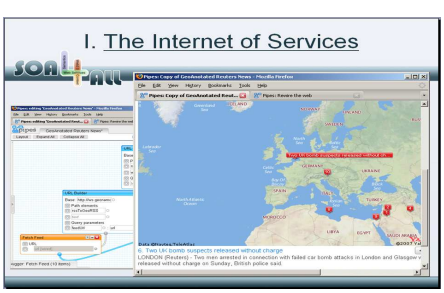
Focus groups were used as a self-contained method to perform this study since no interactive prototype was available to evaluate at this early stage and to collect detailed insights into mental models, opinions and experiences of participants [10]. A qualitative research methodology was followed since we had no hypotheses or knowledge about our end users' perception towards service composition. Each focus group lasted for approximately one hour; participant responses were recorded using audio recorders and questionnaires. The overall strategy was to introduce participants to the topic of "*web service composition by end users"* through a presentation, followed by capturing their subjective judgment about the topic through a questionnaire, and finally discuss several themes in small groups. In details, our participants performed the subsequent tasks.

1- Define software services and fill in a participant background form
2- Listen to a 20-minute presentation to familiarize themselves with software services and service composition; this was facilitated by examples and figures (detailed in section 3.1). It is important to note here that the authors did not, in anyway, present or discuss the merits, drawbacks, or technical details of service composition; they mainly explained the meaning of '*service composition*'
3- Complete a subjective service composition questionnaire to capture initial opinions and rating of service composition aspects (detailed in section 3.2)
4- Discuss the potential risks and benefits of service composition by users and anticipate any composition-related problems; this was carried out in small discussion groups containing 5 participants each
5- Suggest potential solutions to overcome the identified problems

3.1 Service Composition Introductory Presentation

The introductory presentation "The Internet of Services", presented by one author, aimed to introduce the concept of web services along with examples of service composition. It started by explaining the difference between conventional services (human-performed services), software services and hybrid services. The influence of current Web 2.0 technologies was argued to enable end users to take part in the development of the web, and the idea is to move this influence to the Internet of Services. Following this, Yahoo! Pipes (a mash-up tool) was used as an illustrative example of service composition (Figure 1). Next, the motivation behind the SOA4All project was explained to the attendees, with the project aiming to transform the current web of information into a web of services through which users of services could also become producers of services. Then the scenario about the creation of a *Meet Friends* composite service was introduced to drive further discussions. This

hypothetical composite service, which contains four atomic services, allows users to organize a meeting with friends at a short notice. Finally, the presenter showed mockups of a future authoring service composition tool, SOA4All Studio (Figure 2).



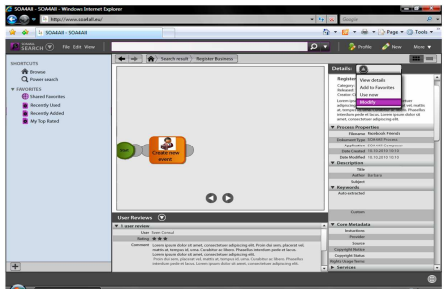


Figure 1. Yahoo! Pipes as an elaborating example (left)

Figure 2. SOA4All Studio – a composition tool under development – (right)

3.2 Service Composition Questionnaire

The service composition questionnaire used in our study consisted of three main parts. Part one captured users' service composition experience and the composition languages and systems they have used. Part two captured users' rating of various aspects of service composition. Part three rated users' opinions on ways for supporting service composition. Questions were rated on a five-point Likert scale where 1 corresponds to "*Disagree*" and 5 corresponds to "*Agree*".

Although the questionnaire contained some questions which are difficult to assess at this stage, for example, it is rather hard to judge whether "composition by end users is easy to achieve" without actually trying it, the principal aim was to drive initial impressions about service composition and check users' acceptability of this innovative idea. Furthermore, the results of this questionnaire provide a reference point to future evaluation stages of our composition authoring tool (SOA4All Studio).

4 Results

We first report on the background of our participants and then on the study results which are divided into three main themes: service perception, risks and benefits of "service composition by users" (for short SCU), and service composition problems.

4.1 Background of Target User Groups

As previously mentioned, two categories of users participated in our study: end users and IT professionals. Analysis of variance (ANOVA) tests (following the Games-Howell procedure since it does not assume sample sizes are equal) showed that IT professionals' experience in software development ($f(1, 62)= 54.64$, $p<0.001$), web service development ($f(1, 62))= 24.06$, $p<0.001$), with analysis and design notations ($f(1, 62)= 14.32$, $p<0.001$), and in service composition ($f(1, 61)= 9.15$, $p<0.01$) were all significantly higher than end users. This is a rather predictable result since 75% of IT professionals had IT-focused degrees. The groups of general end users had few IT-students but with no working experience with software services. Table 1 summarises the background and skills of each of the two user groups.

Table 1. End Users and IT Professionals Background

Criterion	Ordinary End Users	IT Professionals
Number of participants	35 (13 males, 22 females)	29 (27 males, 2 females
Experience in software development	2.31[9]/5 (std= 1.13)	4.07/5 (std=0.65)
Experience in service development	2.11/5 (std= 1.15)	3.44/5 (std=0.98)
Experience with analysis and design notations	2.28/5 (std= 1.31)	3.48/5 (std=1.18)
Experience in service composition	2.34/5 (std=0.93)	3.10/5 (std=1.06)
Service composition languages and systems used before	Facebook, iGoogle	Yahoo!Pipes, OWL-S, BPEL4WS, BPML

4.2 Users' Mental Models of Software Services and Service Composition

The qualitative analysis of end users responses revealed 30 user-oriented definitions and only 1 technical definition of web services; no programming-oriented terms were used to define services. Users' definitions of web services varied between: "*online information and service provisioning to people* (39% of the responses)", "*features assisting users to accomplish their tasks and satisfy their needs* (29%)", "*software that enable service composition* (6%)", and "*reusable components* (3%)".

In general, end users' definitions concentrated on two main aspects, (1) describing specific interactions with users in the form of service consumption, such as: providing users with information and delivering expertise, (2) describing attributes/features of services such as: services are intangible and they have a back end. Four end users were not able to provide any definition or examples of web services.

As for IT professionals, 21 definitions were provided. 76% of which were very technical such as: "*reusable network-based components*", "*self-contained units provided by software*" … etc. Only 14% of the responses perceived web services as "*online software*", whilst the remaining 10% described the business aspects of web services (e.g. "*pay as you go software*"). Eight IT professionals did not provide any definition for web services. Table 2 summarizes the perspectives of both user groups and highlights the implications for the design of development environments.

Table 2. Users' Mental Model of Software Services

Criterion	Ordinary End Users	IT Professionals
Level of understanding	Basic, general	Complex, detailed
Details	User oriented, simple terminology	Technical, specialized terms e.g. self-contained units
Features	Interaction, Consumption. Information-oriented, User Interface	Reusability, software components, functionality-oriented, business model

[9] Ratings were performed on a 5-point Likert scale, where 1= disagree and 5= agree

Level of abstraction	High level of abstraction	Low level of abstraction
Implications for service composition platforms	Services should be presented in a *visual form* and *abstracted* from their technical aspects. Use familiar terminology and enable graphical development of service-based systems.	Complex properties of service composition and of services such as input and output parameters, and operations should be revealed to users who are willing to modify them.

Subjective rating of several service composition questions revealed that both user groups are highly interested in service composition (end users (4.20/5), IT professionals (4.32/5)). Similarly, they agreed that service composition is highly useful (end users (4.44/5), IT professionals (4.44/5)), as well as efficient in promoting the accomplishment of online activities (end users (4.12/5), IT professionals (3.86/5)).

ANOVA tests (the Games-Howell procedure) showed a significant difference in the perception of easiness ($f(1, 61)) = 10.87$, $p<0.01$) and error proneness ($f(1, 60)) = 18.26$, $p<0.001$) of service composition (Figure 3). Indeed, "*service composition by users*" was regarded easier (3.32/5) by end users than IT professionals (2.37/5). However, IT professionals perceived the composition as more likely to make errors (3.55/5) than end users (2.54 /5). Interestingly both groups agreed that service composition can break organizational rules and policies (end users (3.50/5), IT professionals (3.51/5)) which suggests worries about loss of personal information.

Figure 3. Rating of Service Composition aspects by end users and IT professionals, * refers to questions that were rated significantly different according to ANOVA tests

In regard to ways of supporting and encouraging service composition, ordinary users rated that "successful examples (4.69/5) and training courses (4.38/5) could encourage people to be actively involved in the composition of service-based applications" significantly higher than IT professionals (4.34/5 and 3.55/5 respectively), as shown by the ANOVA tests ($f(1, 60)) = 4.88$, $p<0.05$, and $f(1, 61)) = 15.24$, $p<0.001$ respectively). End users strongly thought that quality standards and testing will decrease risks of service composition (4.32/5), whilst IT professionals were less convinced (3.71/5), ANOVA tests were significant ($f(1, 60)) = 8.30$, $p<0.01$). Both groups agreed that recognising and rewarding service composition efforts will increase people's willingness to uptake development activities (end user (4.15), IT professionals (3.82)).

In summary, both end users and IT professionals were highly interested in composing services and strongly agreed that service composition is possible and useful, but expressed uncertainty about the difficulty and potential misuse of service composition by the general public. It is notable that IT professionals view in respect to the idea of '*service composition by end users*' was more critical and realistic than ordinary end users. They were less convinced that examples and training classes could help uptake of service development activities by users, probably owing to their awareness of the difficulty programmers encounter when developing composite services. The implication for service composition platforms is *to identify techniques that simplify the composition process for non-programmers, protect end*

users from making mistakes and help them localize faults, guide users actions during service composition proactively, and provide realistic examples, tutorials and demos about service composition and its concepts.

4.3 Risks and Benefits of Service Composition

Amongst the themes discussed by end users, only 7.2% of the topics were related to the benefits of service composition by users, whereas 25.5% of the topics covered the risks of SCU. Similarly, IT professionals mainly concentrated on the risks of SCU (37.7% of the topics discussed), giving less attention to the benefits (only 7.5%).

In terms of benefits, end users mainly focused on the usefulness of reusing composition knowledge (40% of all benefit responses), and the time users can save as a result of this (30% of all benefit responses). Giving ordinary users control over service composition would empower them to produce various service-oriented applications that can be tailored to their needs (15% of all benefit responses), such as meta-search engines, thus saving time and obtaining richer results. However, IT professionals argued for the efficiency of service composition as it saves time and efforts (41.6% of all benefit responses). Service composition can also be used to generate income (25% of all benefit responses) e.g. "*you need to create a business model around that and you would need to generate income somehow*".

In terms of risks, end users' biggest fear was about losing control over personal information (46% of all risk responses), especially when the effect is mediated through the effect of social interactions (e.g. friends exposing your information), or through the service provider, which may pass personal information (e.g. phone number) to other sub-contracting services e.g. "*I would be concerned whether the details are given to third parties*", which may or may not be bound to the data protection principles. Technical difficulty imposed by service composition was also amongst the top concerns for end users (17% of all risk responses) e.g. "*to build a system that can include all those different services and provide an interface for them is quite difficult*". Errors in putting information together were also possible, especially when the composition is performed by inexperienced users and un-trusted third parties. Moreover, users felt that services may no longer be there when they need them, and that any recommendation support for services may be biased to a set of services. Likewise, service developers specified that their major concerns were about data privacy (21% of all risk responses) e.g. "*once you start allowing components to exchange information, there goes privacy*", followed by security issues when using infected (e.g. virus) or compromised services (18.33% of all risk responses), and trust issues when using services from unknown service providers (6.66% of all risk responses), e.g, "*there can be trust security and trust issues involved using someone else's services*".

We have categorized the risks identified by our users according to two factors: likelihood of occurrence (very likely, likely, unlikely, and highly unlikely) and severity of results if a specific risk occurs (slightly harmful, harmful, and very severe). These two factors determine the probability and seriousness of a risk and its influence and enable us to prioritize and concentrate our efforts on the probable incidents. At the time of focus groups we were not aware of the potential risks of service composition, thus we were not able to ask our users to classify the risks. However, during the analysis process we referred to users' comments to inform our categorisation. Table 3 organizes the potential risks of service composition.

End users and IT professionals also discussed what could be the social and organizational support for user-based service development. For instance:

- "Go with the flow" – once everybody is doing it, people will join, mirroring success in other technologies;
- Efficient examples of successful use will also help (to sell benefits), this was felt quite

strongly;

- Community-level control mechanisms, such as feedback, would ensure validation of services and, together with a validating body/watchdog may help to ensure the trust, which is considered vital for uptake of user-driven service composition.

Table 3: Risk categorization according to severity of outcomes and likelihood of occurrence

	Severity of outcome			
Likelihood of occurrence		Slightly harmful	Harmful	Very severe
	Very likely		Technical difficulty (*end users*)	
	Likely		-Errors in putting information together (*end users*) -Trusting unknown services (*IT professionals*) - Awareness of implications of actions (*Both*)	-Privacy of personal information (*end users and IT professionals*) -Security of services (*IT professionals*)
	Unlikely	Biased recommendation of services (*end users*)	Unavailability of desired services (*end users*)	
	Highly unlikely			

4.4 Problems of Service Composition

Although both categories of users favoured the idea of assembling services to produce interactive applications that fulfil their special needs, several service composition-related issues were raised, in particular:

- *Services complexity*: services are usually represented using their functional elements (operations and parameters) which are often not understood by ordinary web users. End users are not willing to learn complex concepts in order to be able to compose services; they prefer instead traditional and easy-to-use alternatives.
- *Services compatibility*: users expressed frustration in regards to aggregating heterogeneous services from different service providers. How do they ensure the business services they are trying to combine together are technically compatible with each other?
- *Composition steps*: users agreed that it might be problematic to define the single steps required to combine services together and the order in which these services should be executed due to their lack of technical knowledge and modelling skills. This issue becomes more complicated in the case of many services (for example: 100 atomic services). For an ordinary end user or a domain expert, it is quite challenging to identify the right service that fulfils a specific task and the best way to connect the relevant services.
- *Composition for everyone:* users reported worries about designing composition development environments which target all users; this might not work due to diversity of users' experience, requirements, and computing expertise. The 'one size fits all' metaphor does not hold anymore in the evolving world of Internet.

- Other less aggravated *user interface-related concerns* cantered around the service composition editor e.g. direct manipulation of web services (i.e. selection, deletion, etc) within the design space could be the main source of frustration.

In terms of technical support which can be provided by the composition authoring tool, the following themes emerged:

- The difference between novice and professional users was perceived to lie partially in the awareness about the consequences of one's actions
- Full automation such as Google search results will frustrate owing to lack of control by the end users, a balance should be maintained;
- Tools should offer clarity of process in respect to building and using (i.e. context and personalisation, reuse of designs);

5 Discussion

The comparison between the results of end users and IT professionals yielded very interesting findings. End users showed either no or a very limited knowledge of the technical aspects of services. This is anticipated because this target group has no specialist technical skills, as demonstrated by the background questionnaires. Essentially, end users perceived services as elements which deliver online services (be it information, help, solutions … etc) to fulfil specified user needs. This view necessitates that services should to be *highly abstracted* from their technical complexity and presented in ways that efficiently describe their purpose and functionality. On the contrary, IT professionals showed a high level of understanding of the technical features of services which can be attributed to their profound knowledge about programmable aspects of services.

Both target groups showed a high likeability towards the idea of “composing services into personalized interactive applications”, confirming [12]. This agrees with the current trends that end users are becoming proactive about developing the web. IT professionals perceived service composition as more difficult to achieve than end users since they are more aware of the underlying technical issues. They also thought service composition is likely to cause errors more than end users, which can be associated to the programming (compilation and debugging) problems they regularly encounter. This finding urges to develop composition tools that facilitate development tasks and manage development issues for IT professionals.

End users and IT professionals showed high levels of awareness of the risks and benefits of service composition by users which agrees with [8]. Their greatest concerns in regard to service composition revolved around data privacy and security issues. Hence, for people to uptake development activities high levels of data security and privacy policies must be guaranteed.

In regard to perceived benefits, users argued that service composition will save time and enable them to develop applications on the fly through a straightforward process. Hence, it is important that end users are enabled to compose services without the need to learn programming languages and modelling notations.

A Preliminary Model of Service Composition Uptake: figure 4 summarizes the relationship and interaction between user mental model, perceived risks and benefits, and service composition. In principal every user has initial expectations about service composition and is directly affected by what they know, their abilities, domain and working experience. This mental model is improved via practice after users get involved into building service-based applications. Sadly our findings suggest that the risks associated to service composition outweigh the benefits, as perceived by end users. It is thus essential to implement accurate measurements to resolve this unbalance in users’ perception because

simply this is the first step towards realizing successful service composition by and for everyone.

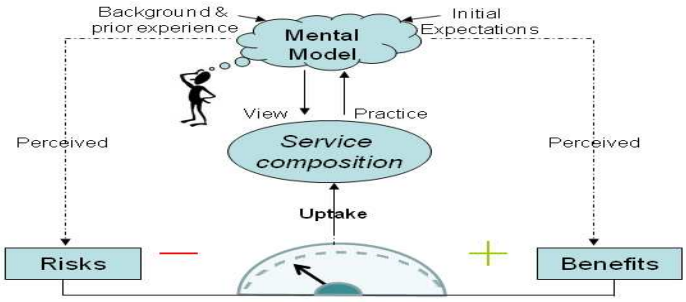


Figure 4: A preliminary model of service composition uptake by end users

To restore the balance between perceived risks and benefits and overcome the above service composition problems, various tentative remedies are proposed.

Guideline 1'Promote service composition awareness': composition editors should clearly communicate "the composition aspect" of services. Users' awareness of the possibility to develop service-based applications should be elevated via the right amount of publicity to familiarize ordinary people with SOA technologies.

Guideline 2'Simplify service composition': it is crucial to simplify service composition by hiding the technical aspects of services from users. Composition should be as easy as dragging and dropping a service into a design space, followed by creating connections between the selected services.

Guideline 3'Guide service composition': users should be supplied with wizards, tutorials, help messages, and composition templates to guide them through the service composition process within an easy to use composition tool. The provisioning of examples and training is also important to support and encourage SCU.

Guideline 4 '*Specialize service composition platforms*': for each category of users with a particular set of characteristics and skills, specialized composition platforms which employ appropriate visual paradigms and metaphors should be offered.

6 Conclusion

This paper summarizes the results of five focus groups aiming to gauge and compare perceptions of two different target user groups, end users and IT professionals, on software services and their willingness to uptake service composition activities. In general both groups of users showed a high willingness to develop interactive service-oriented applications, but expressed concerns that relate to privacy, security, and complexity underlying the composition process and to the knowledge required to build software applications. These concerns should be addressed well in order to restore the balance between perceived risks and benefits (as shown by our preliminary model) and thus motivate and involve ordinary end users in the development of service-based systems. In future research, various composition design approaches of different complexity levels will be offered to accommodate users with various skills and backgrounds levels within an easy to use online service composition tool, formally known as SOA4All studio.

References

1. Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web Services: Concepts, Architectures, and Applications. Springer Verlag (2004)
2. Braun, V., Clarke, V.: Using Thematic Analysis in Psychology. Qualitative Research in Psychology, v 3, n 2, p77-101, (2006)
3. Green, T., Blackwell, A.: Cognitive Dimensions of Information Artefacts: A Tutorial, http://www.cl.cam.ac.uk/~afb21/CognitiveDimensions/CDtutorial.pdf (1998)
4. Jinghai, R., Xiaomeng S.: A Survey of Automated Web Service Composition Methods by: Semantic Web Services and Web Process Composition, Vol. 3387/2005, pp. 43-54 (2005)
5. Li, Karen Na-Liu.: Visual Languages for Event Integration Specification. PhD Thesis, University of Auckland, Department of Computer Science (2008)
6. Lieberman, H., Paterno, F., Wulf, V. (eds.): End User Development. Springer, Germany (2005)
7. Martinez, A., Patino-Martinez, M., Jimenez-Peris, R., and Perez-Sorrosal, F. ZenFlow: A Visual Web Service Composition Tool for BPEL4WS. In Proceedings of the IEEE Symposium on VLHCC. IEEE Computer Society, Washington, DC, 181-188 (2005)
8. McGill, T., Klisc, C.: End User Perceptions of the Benefits and Risks of End User Web Development. Journal of Organizational and End User Computing 18(4): 22-42 (2006)
9. Mehandjiev, N., Stoitsev, T., Grebner, O., Scheidl, S., Riss, U.: End User Development for Task Management: Survey of Attitudes and Practices. In Proceedings of IEEE Symposium on VLHCC. IEEE Press. ISBN : 978-1-4244-2528-0 (2008).
10. Morgan, D. L.: Focus Groups as Qualitative Research. California, Sage Publications, (1997)
11. Myers, B., Pane, J. F., Ko, A.: Natural Programming Languages and Environments. Communications of the ACM. (Special issue on End-User Development). 47 (9) (2004)
12. Namoune, A., Nestler, T., and Angeli, A.D. End User Development of Service-based Applications. 2nd Workshop on HCI and Services at HCI 2009 Cambridge, (2009)
13. Papazoglou, M. P., Traverso, P., Dustdar, S., Leymann, F.: Service-Oriented Computing: State of the Art and Research Challenges. Computer 40 (11), 38-45. (2007)
14. Sutcliffe, A., Lee, D., Mehandjiev, N.: Contributions, Costs and Prospects for End-User Development. Proceedings of HCI International, Lawrence Erlbaum Associates, Inc. New Jersey, USA (2003)
15. Sutcliffe, A.,Mehandjiev, N.: 2004. Introduction: Special Issue on End User Development. The Communications of ACM, 47, 9, 31-32. (2004)
16. Zaha, J., Barros, Alistair P., Dumas, Marlon, ter Hofstede, Arthur H. M.: Let's Dance: A Language for Service Behavior Modeling. OTM Conferences (1) 145-162 (2006)