international reports **on** socio-informatics

Requirements Report of the INSEMTIVES Seekda! Use Case

Guest Editors:

Roberta Cuel, Marc Herbrechter, Markus Rohde, Martin Stein, Oksana Tokarchuk, Torben Wiedenhöfer, Fahri Yetim, and Marco Zamarian

INSEMTIVES
FP7-ICT-2007-3
Contract no.: 231181
www.insemtives.eu

Disclaimer

This document contains material, which is the copyright of certain INSEMTIVES consortium parties, and may not be reproduced or copied without permission.

All INSEMTIVES consortium parties have agreed to full publication of this document.

The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the INSEMTIVES consortium as a whole, nor a certain party of the INSEMTIVES consortium warrant that the information contained in this document is capable of use, or that use of the information is free from risk, and accept no liability for loss or damage suffered by any person using this information.

[Full project title] INSEMTIVES – Incentives for Semantics

[Short project title] INSEMTIVES

[Number and title of work-package] WP6: Marketplace for Web Services

[Document title] Requirements Report of the INSEMTIVES Seekda! Use Case

[Editor: Name, company] Markus Rohde, Siegen University and IISI

[Authors: Name, company] Roberta Cuel (Trento University), Marc Herbrechter (Siegen University), Markus Rohde (Siegen University), Martin Stein (Siegen University), Oksana Tokarchuk (Trento University), Torben Wiedenhöfer (Siegen University), Fahri Yetim (Siegen University), and Marco Zamarian (Trento University)

Copyright notice

Table of Content

Requirements Report of the INSEMTIVES Seekda! Use Case

Roberta Cuel*, Marc Herbrechter**, Markus Rohde**, Martin Stein**, Oksana Tokarchuk*, Torben Wiedenhöfer**, Fahri Yetim**, and Marco Zamarian*

* Department of Computer and Management Sciences, University of Trento, Via Inama, 1, 38100 Trento, Italy, {roberta.cuel, oksana.tokarchuk, marco.zamarian}@unitn.it;

** Department of Information Systems and New Media, University of Siegen, Hölderlinstr. 3, 57076 Germany, {marc.herbrechter, markus.rohde, martin.stein, torben.wiedenhoefer, fahri.yetim }@uni-siegen.de

Abstract. This report presents some first results we obtained from interview sessions and a focus group conducted at Seekda! in Innsbruck on March 12th, 2010. The requirements came out from 8 interviews and a focus group with 14 participants all representative employees of Seekda! and users of their web services search engine. The results focus on the analysis of the Seekda! Webservice search engine prototype, and tend to identify design requirements with respect to: Usability requirements, community requirements and incentive requirements, intrinsic motivation aspects that might drive users to contribute to the portal and variables that should taken into account in order to design incentives mechanisms such as players; rules and social context; expected outcome and payoffs; goals and tasks.

1 Introduction

"I have a dream for the Web [in which computers] become capable of analyzing all the data on the Web – the content, links, and transactions between people and computers. A 'Semantic Web', which should make this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The 'intelligent agents' people have touted for ages will finally materialize." (Berners-Lee & Fischetti 1999)

Making machines and applications capable of understanding the information behind data will enable them to co-operate and thereby create applications which will vastly simplify information retrieval processes and other complex tasks. Resulting benefits of this combination of computational power and human creative intelligence are barely predictable. But bridging the gap between human and computational intelligence is a difficult task which first has to find a solution for creating meta-data about data objects. One technology addressing this problem is semantic technology. It offers powerful concepts to describe the information provided by data and the structure which lies beneath the different information artifacts. The most common way of describing the structures are ontologies. They enable us to provide a shared specification of a domain, consistent of involved entities and their relations (Gruber 1995). The concrete information based on the ontologies can be provided for example via tags, which are keywords describing the content of information objects. One approach for enabling users to easily create and edit ontologies using RDF (RDF Working Group 2004) has been introduced by the Protégé system (Natalya Fridman Noy et al. 2000).

The adaption of semantic technology has progressed over the last few years but is far from large-scale dimensions. Without a proper usage of semantic technologies the value of semantic content like annotated Web pages, semantically enhanced multimedia repositories or widely accepted ontologies, will remain unseen, as the practical advantage of this information relies on the gathering of a critical mass of semantic content. This information brings benefit to all kinds of application and will cause an immediately added value for the users of such applications. It is therefore understandable that in the past decade the question of how to create semantic content, ontologies as well as semantic meta-data, effectively and efficiently has received considerable attention in several areas related to semantic technologies, from ontology engineering, ontology learning and ontology population to the semantic annotation of multimedia and Web service resources. But even by creating effective and efficient tools for semantic content generations, problems concerning qualitative control mechanisms for the created semantic content remain unaddressed. Therefore, users have to actively engage in the process of creation to ensure valuable and adequate outcome.

By not only focusing on quantity but also on quality and by not trying to generate semantic content automatically as is the case with other approaches, INSEMTIVES tries to motivate the user to actively participate in the process of information creation. Therefore, the different approaches of motivational concepts have been examined reaching from intrinsic motivation caused by online community participation arising within Web 2.0 applications (Huysman et al. 2003; Rohde et al. 2004; Rheingold 2000; Wenger et al. 2002) to extrinsic motivational factors or incentives which have a long tradition in economy and organizational science. The approach integrates concepts for community support, participation management, usability engineering, and incentives theory in order to guarantee a higher user involvement and motivation.

Further, INSEMTIVES investigates social and economic incentives models for fostering user participation and motivation by relying on the expertise of the project partners reaching from theoretical foundations in semantic technology and data (respectively knowledge) management concepts to profound methodological knowledge in the measurement of economic incentives schemes and user-centered computing approaches. The practical and scientific relevance is founded on the usage of methods from different disciplines, like usability engineering and participatory design, applied on three case studies in the sectors of telecommunications, online marketplaces and online games.

In this report, we present the results of requirements engineering processes at the SEEKDA! company, which offers a web service search engine that partly relies on user-generated semantic contents. On the one hand, this process focused on usability tests, expert walkthroughs and practice orientation, but on the other hand addressed incentive mechanisms and human behaviour. Therefore, the collected empirical data were analyzed separately by two groups of the project, each with a profound knowledge in the investigated issues.

2 Structure of the report

The work has been carried on by two different groups of researchers with different point of views.

The first group focuses the attention more on user-centered approach and practice orientation, Participatory design, organization and technology design, usability tests and expert walkthroughs.

The second group focuses more on lab and field experiments, incentive mechanisms, and human behaviors.

Mainly for the fact that the two groups work from different perspectives, pointing out various aspects of incentives and motivations, the report will be composed by two main paragraphs:
-    Requirements from University of Siegen
-    Requirements from University of Trento

3 Method of analysis

To investigate this topic, we made one day interviews with 8 representative employees and experts of SEEKDA!. Each semi-structured interview was conducted by two interviewers, took 60 to 90 minutes and was recorded on audio tape. These recordings have been transcribed and analyzed descriptively according to ex-post categories. Additionally, a focus group discussion was conducted, focusing on usage problems of the existing system and on possible design solutions to overcome these problems. The interviewees tried to explain whether and to what extent semantic annotation can actually improve the SEEKDA Portal. SEEKDA! interviewees' feedbacks were decisive for the direction of the design, depending both on their impressions and their usage along the way.

4 Requirements from University of Siegen

These incentive and software requirements for the Seekda Use Case are structured into:
1. Usability requirements
2. Community requirements
3. Incentives requirements
4. Motivation requirements
5. Critical incidents of the search engine

The following tables show the task requirement on the left and the resulting software requirements on the right.

4.1 Usability Requirements

#	Task requirement	Software requirement
1	Developing internal web services	The System needs to offer technical documentation for professional users (Perhaps different views for professionals and non-professionals, maybe an abstract description of the WSDL).
2	Finding appropriate web services	The System needs to provide categories and tags to provide information about the functions of the web service.
3		The System needs to show more self explaining search results. (E.g. ordered by relevance according to search keywords or alphabetical etc.).

4	Search for a web service	The System needs to provide a keyword based search [Already provided by current prototype (Maybe emphasizing needed)] (E.g. search for specific vendors of web services)
5		The system needs to support recommendations for web services (perhaps automated sending of emails with recommendations, kind of "tell friends").
6		The System needs to include REST-Services in search results to offer a richer pool of services (SOAP services are mostly for internal use and therefore not widespread)
7		The system should support the user in finding the appropriate search term (e.g. real-time query expansion, like Google suggestions).
8	Interpret search results	The system needs to offer a check routine for user provided information. Options: Expert checks information or community adds/ changes information and the results get rated and thereby validated or removed.
9		The user should be able to sort/filter search results according to further categories such as latency, which are marked as relevant by interviewees. [Already provided in general, needs extension though]
10		The System needs to be able to filter web service by their providers (e.g. important if using web service of a partner). [Already provided by current prototype (Maybe emphasizing needed)]
11	Non-professional user: display of search results	The System needs to offer an easy to use mechanism for the web services, because normal (non-developer) users want to use the service they find to test. [Already provided in general, needs extension though]

		(i.e. let users annotate cryptic function names with more speaking descriptions)
12	Using a matching web service	The System needs to show examples of usage and integration. (I.e. code examples)
13	Interpreting specific web service pages	The system needs to provide an overview of the history of specific web services.
14		The system must provide at least the following facts: 1. Usefulness of the web service 2. Reliability of the web service 3. Easiness to implement the web service 4. Providing company of the web service 5. Rating of the web service 6. Technology of the web service 7. Latency of the web service
15		The rating option has to be a general one and results should contain an image providing a fast overview.
16		The System needs to show what the web service offers (functions) and how to use them (e.g. with examples of usage) and what's required to use it (costs, registration, etc.). (Already provided in general, needs extension though)
17		The system needs to show a clear title / description. Currently, titles are mostly the web service name an often non-speaking to the user.
18		The System needs to provide mechanisms to check the correctness and actuality of the web service.
19		The system should make clear statements for sources and origins of content.
20		The system should provide clear

		information (i.e. # of voting users to get a feeling how to interpret the ratings).
21	Annotate (e.g. tag) specific web services	Consumers need to be able to add annotations to extend the understanding of an object or to clarify a different view (e.g. let other interested users know that a specific web service worked in a whole other way than described for me)
22		The system should only provide the option to add tags instead of tags and categories, because categories can be derived from tags.
23		The system must provide a way to let the community evaluate and remove tags if desired.
24		The System needs to show benefit of annotating for the users (E.g. finding information quickly)
25		The system should provide options for the user to mark the current state of the web service (available, broken etc.)
26		The system needs to motivate the user to add a comment why he/she is disappointed resp. why the web service is not matching his/her needs.
27		The System should remind the user to annotate, without annoyance.
28		The system should provide a clearly visible way to add semantic content. (I.e. the user should instantly be aware that he/she can add semantic content)
29		The system could provide a way of letting the user make private notes and anonymously make them public, because some interviewees stated to do annotations just for themselves.

30		The system should provide automated mechanisms of annotation. That could be relevant for the portal operator. (I.e. time spent viewing specific web service pages).
31	Differentiation between web service types	The system should provide a way to distinguish the type of web service (REST / SOAP).
32		The system must provide a way to search for all kinds of web services.
33	Ontology support	Developer's statement: *The system has to provide the possibility to add ontologies created by the developer or the providing company. Users shouldn't be able to create them.* *(There were varying opinions whether ontology building should be done by developers or end-users)*
34		The system needs to enable the user to provide different kinds of information (I.e. structured/unstructured, formal/informal with checkboxes or dropdown fields), which could ease a user into annotating by guiding him
35		Description for finding appropriate web service, usage examples and ontologies to share a common understanding
36		The system should enable the user to suggest an ontology which has to be validated by a professional (perhaps: Role Model of Users)
37		The System must use existing annotation to guide users to relevant web service.
38		The System must make use of ontologies to display hierarchical and semantic structures between web service (Semantic: book travel: hotel, car, flight etc.

#	Task requirement	Software requirement
		Hierarchical: book a vehicle: car, bike, train etc).
39	Enlistment of web services / Crawler control	The Producer needs to be able to add initial information like a title and a description.
40		The System needs to support the initial process of describing / annotating to help producer to publish their services. (E.g. let web service providers use meta data in WSDLs to provide additional information)
41		The System must be able to notify the provider about information given by consumers to extend their documentation.

4.2 Community requirements

#	Task requirement	Software requirement
42	Community oriented annotation	The System has to provide suggested tags. (The creation of tags should be done by single users themselves, but the community can support the process with inspiring by showing other tags).
43		Community sourced annotations have to be evaluated by the community. Users must be able to create ontologies which are presented to the community to improve it.
44	Supporting community building	The System should create a social awareness to show users the benefit of their contribution for other users (Motivation). (E.g. creating a facebook app that uses the search of the portal).
45		The system should provide a way for users to communicate directly.
46		The system should provide the feeling of belonging to a group or community.

47	Participation awareness	The System should publish the most active user or users with high credibility (good rated annotations).
48	Community support	The system needs to enable a single person to add information and provide this information to a group of people.
49		The System can create social awareness by showing annotations of others, which can be utilized as motivation.

4.3 External Incentive Requirements

50	Motivate users to participate	The System should create social awareness to show users the benefit of their contribution for other users (Motivation). (E.g. creating a facebook app that uses the search of the portal).
51		The system needs to motivate users to annotate in negative and positive cases (i.e. the system should prepare for the case a user forgets to annotate, e.g. by remembering which service page the user exits by and reminding him/her at the next login)
52		The system should provide mechanisms that will show users the advantage of annotating. Annotating can be used as payment for search operations ("tip for tag")
53		Direct positive feedback for contributions. The user should get immediate feedback from the system. (E.g. a thank you note, credits etc.)
54		The system should provide a ranking system of participants.

4.4 Internal Motivation Requirements

55	Motivate users to participate	The System needs to visualize benefit of annotating for the users in **real-time**. (E.g. finding related information quickly).
56		The System needs to publish the most active user or users with high credibility (good rated annotations).
57		The system should make it fun to contribute. Annotation functions should be best integrated into the whole site. (E.g. suggest search term as tag for selected specific web services)
58		The system should provide a ranking system of participants.

4.5 Critical Incidents

In this section we list several critical incidences within the Seekda search engine.

#	Critical incident	DIN Violation
59	Users may get confused by Google Adword-like boxes on the right (Front-page)	Conformity with User-Expectations
60	Doesn't understand the actions of the buttons (Prototype mockup)	Self descriptiveness
61	The system lacks professional appearance to some interviewees (General)	Conformity with user expectations
62	The system has to distinguish between professional and non professional users	Suitability for individualization
63	The interface design for direct usage of services must be pleasant and easy to use (Specific web service page / use now page)	Controllability
64	The Interface of the System has to be more self-explaining and pleasant	Self descriptiveness

65	Usage of "professional-vocabulary" should be desisted	Self descriptiveness, conformity with user expectations
66	The system should provide a clearly visible way to add semantic content.	Suitability for learning
67	The system should provide a clear navigation structure.	Controllability
68	The system should provide a simple interface, hiding complex functions. (Specific web service page)	Controllability, conformity with user expectations

5 Requirements from University of Trento

SEEKDA! Case is is a typical case of public good. The main target group that uses search engine consists of professional software developers. Professional software developers look for web services when they are on a hurry to finish a project and do not have time to develop their own application or address open source solutions. For search of web services google is mostly used. Otherwise, websites that already use web service of interest are consulted and information about the web service is retrieved from there. When the task is functionality of web service SEEKDA is better than google- you can check parameters, if the service is on line, it's quite unique and very useful.

When SEEKDA! Web services search engine is used for search only web services that have community description are consulted. Some interviewers stress that the more comments are there for the web service the higher probability that they will consider trying out this web service. For other users it is enough to have a decent description of what the service is for considering it. Web services without a description are not consulted unless there are no services with description and they cannot be found otherwise. Only a couple of web services without description may be tried out and the user switches the search to sources different from SEEKDA!

While the presence of community description is an important factor in the use of the search engine, most of interviewers would not add such a description themselves. They might do so only if the service they found did not have a description and they were highly satisfied with it. However, given that services without description will not be tried out, this kind of contributions are highly improbable. Interviewers stress that they are busy and short of time. They don't mind sharing something they do for themselves. Thus, if adding the webservice to bookmarks they tag it or add a description they don't mind sharing it with the community. The same concerns their applications of web services. A frequent motivation of sharing something with community or contribution is reciprocity

among community members. Most of interviewers believe that they fill they have to share because others shared or will share.

Users like to receive some recognition of their contribution. A simple thank you note is enough. It feels really bad when a comment is not saved by the system. The system SHOULD save the comment and give a thank you note.

It makes sense to have an expert that provides descriptions for an initial set of services to make the community start working. These experts may be motivated with a pay, reputation issues, challenge competition, etc.

Information with stars is considered useless. It is not clear what for stars were given, how many users valued the service and upon which criteria.

Tags presently available in the system are pointless and do not encourage the use of tags neither by tagging nor by searching with the use of tags.

REQUIREMENTS	HOW TO ADDRESS REQUIREMENTS
The web services are still few on the Internet.	Not addressable by SEEKDA
The web services on SEEKDA portal are "toy web services". Most of them are not important for the work developers carry on	Not addressable by SEEKDA
Description of service is very useful and important especially if provided by community members. Services without description are not consulted. Problem: the target users (professional programmers) are not interested in providing the description of services	Take advantage of the academic interest in SEEKDA portal and make competition among members of the academic community in providing quality descriptions of services. Possible motivations can be: reputation, some prizes (like i-phone, etc), possibility to be employed by your company, contribution to research and development of web services
The description of the service should be very informative. Problem: professional programmers don't want to spend time refining the information on the portal, they would only attempt to understand the whole functionality of the web service and use it for their own purpose	Assure the quality of the description of the services: • Provide guidelines or the standard for the good community description • Possibility of rating of the description by the user (did you find the description helpful ratings), including the history of descriptions • Provider of the description receives based on the use of

	his description
History of the description (and evaluations) of the web service is hardly accessible, now it is in wiki and can be consulted only one at a time with long waiting time	Make the history of the community description easily accessible (hidden under the latest definition and displaying all the history at one click)
Understand if the service is doing what I need	Service description
What technology is used And how the web services can be implemented in the system	Documentation
Availability	Automatic searching and (re)calling of the web service
License of the service	Since the services are provided by third parts cannot be addressable by SEEKDA
The tags and tag cloud presented in the system are not informative and is useless for search	Make sure that tags that are entered in the system are informative: - Definition of a list of possible tags during description of the service. Users can choose the tag from this list (a sort of dynamic ontology) - Peer-review of tags (motivations for peers to review?) - tagging for personal use and enrich system by these personal tags - Other?
Rating with stars is not informative: - Not clear criteria upon which the rating was provided - How many people (or the system) provided the rating - What was the task of the person that provided the rating	Identify criteria that are important for users. Provide ratings based on these criteria and analytics with these ratings (how many people provided certain rating, did they leave comments, do the system evaluate accessibility, etc.). or Eliminate ratings at all
Encourage feedback both positive and	SEEKDA should provide a sort of

negative	artifact for the community of web services developers and users through which actors interact
I'm not a constant user of the portal	Make the person come back to the portal as often as possible: make the portal a useful place of meeting for the users so that if they have any problem, question or interest in web service they immediately think of SEEKDA portal
Understand if the service is doing what I need What technology is used Availability	

5.1 Motivation

The target users are professional software developers that are on hurry, they look for web services to solve their concrete problem but are not interested in contribution in general. In the following we list motivations to contribute to other communities, i.e. open source project, etc.

MOTIVATIONS	HOW TO ADDRESS INCENTIVES
Reciprocity: I make use of other people's contributions and feel that I have to contribute to retaliate	Make it clear that what you use (community description, search by tags, etc.) was created by others and you can contribute to help others as well
I need to have personal benefit, I don't have time just to be part of community	Make it possible to do the work for yourself that is then shared with the community (add in bookmarks with tags, description, rating, etc. that are shared with the community
I do something that gives me personal benefit, i.e. improve open source program, I don't mind sharing it with others. I will never do something that is not of direct use for me just to help the community	Make it possible to upload personal applications (or other) that implement webservices found with SEEKDA search engine
Reputation: I contribute to make others know what my competences are in order for them to address me if they	Possibility to enter in personal info areas of competences and interest, make search of friends in areas of

have similar problem	competences. Give possibility to community members to lead groups dealing with some specific questions, etc. – encourage visibility of competences of single members and building of their reputation
I'm obliged to contribute, i.e. answer questions, if I'm a head of the open source project	Give possibility to lead a discussion group on a specific question, lead a project within the community, etc. – make members responsible for something in what they are competent
I will hardly contribute on an open source project, i.e. answer the question, if I know that there are other people, that have more knowledge, experience, entitlement (i.e., head of the project) that may contribute	
A simple thank you not is enough to recognize my contribution	Recognize contribution: thank you note, points for contribution, board with members that contribute the most, etc.
It feels really bad if the system doesn't save the comment or other contribution	There shouldn't be any bugs with saving and listing of comments and other constributions
Something that makes your life fuller	

In the case of SEEKDA! it makes sense to create a community of professional software developers around the web service portal. The aims of the community:
- Provide web services available and easily to retrieve and compare
- Provide a place of communication between the members of community regarding the webservice
- Provides a place where to build reputation in the field of web services
- Builds on reciprocity among actors
- It is a "useful" place where it is easy to find what you are looking for.

5.2 Weakensses that Seekda should take into consideration

Most of the inner motivations that drives individuals in free-libre open source software are in contrast with any private utility the company can obtain by the portal. Since the Seekda portal is considered as a community portal, the information and the services should be freely available to the community (and to the ones want to join the community).

6 References

Berners-Lee, T. & Fischetti, M., 1999. *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*, Harper San Francisco. Available at: http://portal.acm.org/citation.cfm?id=554813 [Accessed October 27, 2010].

Gruber, T.R., 1995. Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud.*, 43(5-6), pp.907-928. Available at: [Accessed October 27, 2010].

Huysman, M., Wenger, E. & Wulf, V., 2003. *Communities and technologies: proceedings of the first International Conference on Communities and Technologies, C&T 2003*, Springer.

Natalya Fridman Noy, 2000. The knowledge model of protege-2000: Combining interoperability and flexibility. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=?doi=10.1.1.18.2129 [Accessed October 27, 2010].

RDF Working Group, 2004. RDF - Resource Description Framework. Available at: http://www.w3.org/RDF/ [Accessed October 27, 2010].

Rheingold, H., 2000. *The virtual community: homesteading on the electronic frontier*, MIT Press.

Rohde, M. et al., 2004. Community-Building with Web-Based Systems ? Investigating a Hybrid Community of Students. *Computer Supported Cooperative Work (CSCW)*, 13(5-6), pp.471-499.

Wenger, E., McDermott, R.A. & Snyder, W., 2002. *Cultivating communities of practice: a guide to managing knowledge*, Harvard Business Press.