End-User Requirements for Wisdom-Aware EUD

Antonella De Angeli, Alberto Battocchi, Soudip Roy Chowdhury, Carlos Rodriguez,
Florian Daniel, Fabio Casati

Department of Information Engineering and Computer Science
University of Trento
Via Sommarive, 14 – 38123 Povo, Trento (Italy)
{antonella.deangeli, alberto.battocchi, soudip.roychowdhury, carlos.rodriguez, florian.daniel,
fabio.casati}@unitn.it

Abstract. This paper presents requirements elicitation study for a EUD tool for composing service-based applications. WIRE aims at enabling EUD by harvesting and recommending community composition knowledge (the wisdom), thus facilitating knowledge transfer from developers to end-users. The idea was evaluated with 10 contextual interviews to accountants, eliciting a rich set of information, which can lead to requirements for Wisdom-Aware EUD.

1 Introduction

There are two common approaches to enable less skilled users to develop software artifacts. Development can be eased by simplifying it or by reusing knowledge. Among the *simplification approaches*, the business process management and service computing communities have focused on abstracting process development and service composition into activities, as well as control and data flows. However, these are still challenging tasks even for expert developers [1,2]. Traditional *reuse approaches*, in the form of program libraries, services, or templates (such as generics in Java or process templates in workflows) have targeted developers rather than end-users. Recently, some effort has been invested into knowledge reuse techniques for end-users. In programming by demonstration [3], the system auto-completes a process definition, starting from a set of examples chosen by the user. Goal-oriented approaches [5] assist the users by automatically composing solutions that satisfy user-specified goals. Pattern-based development [4] proposes the use of libraries of patterns provided by experts to represent good development practices, yet patterns, such as the glue patterns in [7], may also be derived from existing compositions. Syntactic approaches [11], for instance, suggest operators based on syntactic similarity (comparing output and input data types), while semantic-based approaches [6] annotate ingredients to support the retrieval of semantically matching elements.

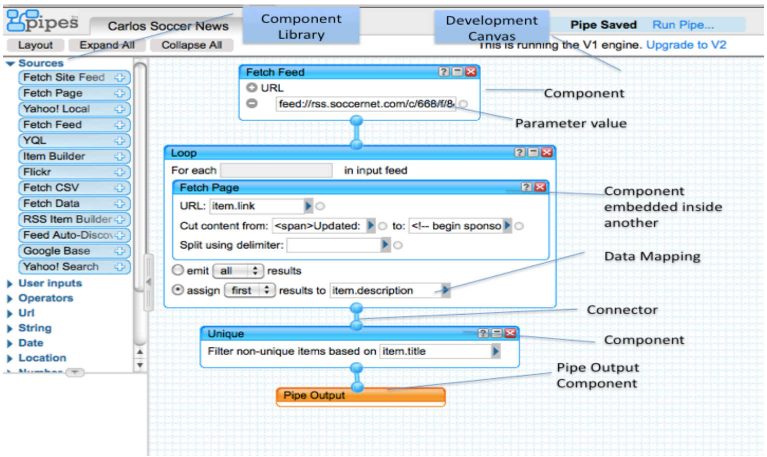
While some of these approaches support end-users with reusable knowledge, they all suffer from some shortcomings. Programming by demonstration and goal-based approaches propose "best", complete solutions, not allowing the user to control which exact ingredients the solution should contain. Pattern- and semantics-based approaches are hard to maintain, in that they require explicit input from human experts.

The final authenticated version is available online at:
https://doi.org/10.1007/978-3-642-21530-8_21

In this paper we present the results of a requirement study for *WIRE* (*WIsdom-awaRE development environment)* a EUD tool to exploit the benefits of simplification *and* reuse. WIRE targets process-oriented, mashup-like applications that are characterized by relatively simple composition logics and complex tasks or components. This class of programs provides the benefit of simplicity (composition, not coding) and a sufficient information base (the compositions themselves). The idea is to *learn from existing compositions* developed by expert IT developers and provide learned knowledge in the form of interactive recommendations to facilitate EUD.

2 WIRE

The motivation behind the idea of WIRE has derived by the analysis of the shortcomings of existing mashup development tools. To exemplify this claim, let us consider a simple application created by Yahoo! Pipes, which retrieves news feeds from a specified website, filters the content based on user-specified criteria, and publishes the filtered content for viewing (Fig. 1). Such a simple application requires 5 components. The user has to set the value of the configuration parameters of a component (e.g., the *URL* Parameter of the *Fetch Feed* component) and define the data-flow logic between components. Assuming that an end-user has this kind of technical knowledge is not realistic.



Fetch Feed
URL: feed://rss.soccernet.com/c/668/f/8...

Loop
For each in input feed

Fetch Page
URL: item.link
Cut content from: <span>Updated: to: <!-- begin sponso
Split using delimiter:
○ emit all results
● assign first results to item.description

Unique
Filter non-unique items based on item.title

Fig. 1. Implementation of the example scenario in Yahoo! Pipes.

WIRE is aimed at discovering technical knowledge by analyzing existing, successful applications, storing knowledge as development advice ("*community composition knowledge"*[8]), and delivering it in the form of contextual *interactive recommendations* to the end-user. The intuition is that this knowledge can be captured through *composition patterns* and reused as recommendations. The patterns we identified include Parameter Values (e.g., values for the *URL* parameter in the *Fetch Feed* component), Component Associations (e.g., suggest that a *Loop* component should be added together with a *Fetch Feed* component), Connectors (e.g., possible connections between components), Data Mapping (e.g., suggest that the *item.link* element coming from the *Fetch Feed* component should be mapped to the *URL* parameter of the *Fetch Page* component), or Complex patterns (e.g., suggest adding components based on a Component Association pattern together with the wiring among them based on a Connector pattern). A detailed explanation of the conceptual model and architecture of WIRE is presented in [8].

3 User study

An evaluation of the conceptual design of WIRE was run in order to address benefits and limitations of the proposal and elicit user requirements [12]. The evaluation was based on contextual interviews to 10 University accountants (7 F, 3 M; mean age = 37 years of age), which lasted approximately one hour. None of them had a background in computer science. Participation was rewarded with 15 Euros. The interview addressed two main topics. Section A targeted the strategies that people use for overcoming the difficulties that emerge while using computers during day-to-day work, and their attitudes towards computer-provided help and advice with particular focus on the comparison between automatic/contextual and on-demand help. Participants were shown a slideshow of familiar examples of automatic/contextual advice (i.e., *word completion* in the Google search box, *friend suggestion* in Facebook, *book suggestions* in Amazon, *passwords auto-save* in web-browsers, *pop-up reminder* on calendars, *related videos* sidebar on YouTube), invited to comment on each example, and report their understanding on how the advice was created.

Section B collected opinions and suggestions about WIRE by a plus and a minus scenario [9] reporting on an accountant who is using WIRE for automating the process of management of travel reimbursement. Both scenarios described the effects brought forward by WIRE on a new user. These effects were taken to the positive or negative extreme to help users to think what consequences the approach could have in their work practices. In the *Positive Scenario*, the accountant had a successful experience, which helped him to save time and speed up repetitive work leading to adoption. In the *Negative Scenario*, the accountant encountered serious difficulties and eventually decided to go back to his traditional work procedures. Scenarios were presented with a counterbalanced order. Interviewees were asked specific questions addressing their willingness to use the system, advantages and drawbacks, preference for contextual or on-request help, and for the way the help was presented.

4 Results

Asking to colleagues and technicians represented the most common option used by half of the interviewees to seek for help and advice. The person to whom they asked for help was usually chosen on the basis of his/her level of expertise or on friendship/acquaintanceship. Google represented the first choice of help for four of the participants and the second choice for those participants who could not find a solution to their problems asking colleagues or technicians. Participants reported using online help and help menus rarely, and this was the first choice only for one interviewee. When asked which source of help was the most effective, eight participants indicated colleagues and technicians. Their choice was motivated by the fact that technicians are professional and helpful, and that providing support is part of their job. One participant indicated Google as the best source of information *"because you can use it at any time, also when you are at home"* (P10).

Seven participants reported a preference for automatic/contextual help rather than help on-request, but two of them also specified that this method works better for new or simple applications. Participants suggested that the automatic/help function should be customizable in order to be really useful. One participant provided an interesting observation about the function of automatic/contextual help:

> *"Automatic/contextual help has a double function: it appears when you need help and reminds you of potential errors; help on request covers only the first function"* (P10).

Participants provided valuable comments on the effectiveness and usefulness of common examples of contextual advice. People favoured less intrusive contextual advice, that do not try to guess the user's preferences or opinions, and that do not present risks for data security, such as Google's *automatic word completion*, *pop-up reminders* in Google Calendar, and the *related videos sidebar* in YouTube. Contextual advice was valued mainly in the case of "objective" suggestions (e.g., YouTube) but perceived as less accurate when it tries to enter users' private space (e.g., Facebook). When asked to formulate their "naïve theories" about contextual help is generated, all the participants reported that they are created on the basis of the inserted keywords. One participant also made a distinction between general, or simple, and particular, or complex, suggestions:

> *"For simple queries, the system works on simple analogies with the inserted keywords; for more complex issues, the system does a matching with your personal characteristics (provided while registering to a service"* (P8).

Participants provided useful information about their attitude toward WIRE. When reading the positive scenario, participants recognized several similarities with their work practices and perceived the system as potentially very useful. Two participants expressed a common concern about the introduction of WIRE into their work practices and suggested that, in order to benefit of its potentialities, the use of WIRE should totally replace previous practices, without leaving space for overlapping. The Negative Scenario was also perceived as very plausible as it described well fears and frustrations that may emerge when something goes wrong dealing with new systems or procedures. In particular the interviewees stressed the need for a system which is well designed and thoroughly tested before being introduced into the work practice:

> *"I gave for granted that this technology was previously tested and approved by the central administration office. […]. In the case of dealing with sensitive or finan-*

> *cial matters, I would trust the system only if I am 100% sure that it is effective and functional"* (P7).

Participants were asked if they would be interested in using WIRE. Nine of the interviewees responded positively and one was openly sceptical stating that *"using WIRE would take the same time it takes doing the procedure manually"* (P1). Anyway, formal training was indicated by two participants as a fundamental prerequisite for adoption. Drivers to adoption were identified in *better organization of work*, *optimization of time*, *reduction of errors*, and *sharing of procedures and methodologies with colleagues*. Major obstacles were connected to *loss of control over work processes*, in the case that these were entirely completed in an automatic way:

> *"I would like to keep track of each step of the process; if everything is made automatically, the users misses the logic that stays behind the process"* (P4).

All the participants declared that the advice provided by WIRE in the scenarios would be effective in meeting their needs, as they were generated on the basis of past experience of colleagues that share the same work procedures and possibly the same difficulties. Interviewees showed a marked preference for contextual help (8 participants) over help on-request; two of them added that the possibility of personalizing the way suggestions are provided would be a very important feature in order to make help messages really effective. Help messages provided *during the task* were preferred to messages provided before the task by nine of the interviewees. One participant suggested that the two modalities could be combined:

> *"I can see the two modalities as complementary. At the beginning of the activity the system asks what your needs are in general; during the activity, pop-up windows provide you solutions when the system feels that you are stuck"* (P8).

5 Conclusion

End-users acknowledged that the idea of WIRE for providing assistance, which was derived from the experience of colleagues working in a similar context, was useful. However, issues related to *trust*, *timing* and *usefulness* of the advice still remained as concerns to the users. During the design of WIRE we will need to find new strategies to make its' operations transparent e.g. showing users how the advice is generated and why a particular advice is suggested in a given context. Transparency will also help to build up the trust of the users to use a recommendation tool like WIRE. This is particularly important when people deal with the sensitive and financial issues. *Personalization* is another desired feature, which enables users to receive optimized advice based upon their expertise level. Helping users with the personalized advice can certainly reduce the barrier for adopting this kind of EUD tools to a larger end-user community.

The study provides support to the proposal of collaborative tailoring, discussed in [10], as often participants mentioned that their willingness to engage in EUD was mediated by having support from other people and technical help easily available to them. This help was meant not only to alleviate some of the technical difficulties they had to face during development but also to take the responsibility out of their hands, making them less accountable in case of software failures. Issues related to organizational regulations and corporate processes also emerged as barriers to general EUD uptake, as people often mentioned the need to have explicit approval from their manager as a fundamental step towards making them willing to explore new techniques and tools to automatise their work practices.

Acknowledgments. This work was partially supported by funds from the European Commission (project OMELETTE, contract no. 257635; project SERVFACE, contract no. 216699).

References

1. Namoun, A., Nestler, T., De Angeli, A.: Conceptual and Usability Issues in the Composable Web of Software Services. *Current Trends in Web Engineering - ICWE 2010 Workshops*. LNCS, vol. 6385, pp. 396-407, Springer, Heidelberg (2010)
2. Namoun A., Nestler T., De Angeli, A.: Service Composition for Non Programmers: Prospects, Problems, and Design Recommendations. ECOWS (2010)
3. Cypher, I., Halbert, D.C., Kurlander, D., Lieberman, H., Maulsby, D., Myers, B.A., Turransky, A. (eds.): Watch what I do: Programming by Demonstration. MIT Press, Cambridge, MA, USA (1993)
4. Van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow Patterns. Distributed and Parallel Databases. 14(3), 5--51 (July2003)
5. Henneberger, M., Heinrich, B., Lautenbacher, F., Bauer, B.: Semantic-based Planning Process Models. In: MKWI'08, Munich, Germany (2008)
6. Ngu, A., Carlson, M., Sheng, Q., Paik, H.: Semantic-Based Mashup of Composite Applications. IEEE Transactions on Services Computing. 3(1), 2--15 (2010)
7. Greenshpan, O., Milo, T., Polyzotis, N.: Autocompletion for Mashups. In: Proc. VLDB Endow. 2, no. 1, pp. 538-549 (2009)
8. Roy Chowdhury, S., Rodríguez, C., Daniel, F., Casati, F.: Wisdom-Aware Computing: On the Interactive Recommendation of Composition Knowledge. Proc. of WESOA (2010)
9. Bødker, S.: Scenarios in user-centred design - setting the stage for reflection and action. Interacting with Computers. 13, 61--75 (2000)
10. Wulf, V., Pipek, V., and Won, M.: Component-based tailorability: enabling highly flexible software applications, Int. Journal of Human-Computer Studies. 66(1), 1-22 (2008)
11. Jeffrey Wong and Jason I. Hong. 2007. Making mashups with marmite: towards end-user programming for the web. In Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '07). ACM, New York, NY, USA, 1435-1444.
12. De Angeli, A., Battocchi, A., Roy Chowdhury, S., Rodriguez, C., Daniel, F. and Casati, Fabio Conceptual Design and Evaluation of WIRE: A Wisdom-Aware EUD Tool. Technical Report DISI-11-353, Ingegneria e Scienza dell'Informazione, University of Trento.