Visual Workflow Composition through Semantic Orchestration of Web Services

Barbara R. Barricelli, Piero Mussio, Stefano Valtolina
DICo – Università degli Studi di Milano
Via Comelico 39/41
20135 Milano (Italy)
+39 02 503 14009
{barricelli, mussio, valtolin}@dico.unimi.it

Marco Padula, Paolo L. Scala
Istituto per le Tecnologie della Costruzione - CNR
Via Bassini 15
20133 Milano (Italy)
+39 02 23699 549
{padula, scala}@itc.cnr.it

Antonio Piccinno
DI – Università degli Studi di Bari
Via Orabona 4
70125 Bari (Italy)
+39 080 544 2535
piccinno@di.uniba.it

ABSTRACT
There is an increasing request by organizations to allow people outside the IT department to create, shape and adapt the software artifacts they use as they do with real ones in their work practice. This paper capitalizes on the experience gained in the collaboration with a certification institution to propose a novel network of software environments and a Web service-based architecture supporting End-User Development activities. The goal is not to automate the skills of the end users, but to assist them in their activity, permitting them to use the knowledge and expertise they possess.

Categories and Subject Descriptors
D.2.6 [**Software Engineering**]: Programming environments – *interactive environments*. H.3.5 [**Information Storage and Retrieval**]: On-line information services – w*eb-based services*. H.4.1 [**Information Systems Application**]: Office automation – *workflow management*. H.5.2 [**Information Interfaces and Presentation**]: User interfaces – *interaction styles*.

General Terms
Management, Human Factors.

Keywords
Visual interaction, Web services, End-User Development, Workflow, Component-based systems, Semantic orchestration.

1. INTRODUCTION
There is an increasing request by organizations to allow people outside the IT department to create and adapt the software artifacts they use as they do with real ones in their work practice [3]. These people become end user developers, who participate to the continuous development and evolution of the software artifacts they use [4]. This paper capitalizes on the experience gained in the collaboration with a certification institution in order to propose a novel network of software environments and a Web service-based architecture supporting End-User Development activities, i.e. the Task Management System (TMS) network, and a Web service-based architecture to support end users to exploit their knowledge and expertise in their daily work activities [1]. In particular, the network allows the end users of an organization to design, execute, and check the execution of a workflow. A workflow is a model to formalize a complex work process for further assessment and manipulation (e.g. for optimization or iteration of specific sequences of tasks) [11]. A work process is described as a pattern of simple tasks, information flow, resources, and roles. Workflow management of cooperative groups of people is supported by software tools to share information, manage communication, schedule and assign tasks to the co-workers. The proposed network addresses two communities of end users participating to the life cycle of a workflow. The first community is the one of the Workflow Designers, which are domain experts, belonging to an organization, not expert in computer science, in charge of designing the current workflow and of supervising its correct execution. The second community is the one of the Workflow Operators, which belong to the same organization of the Workflow Designer, not expert in computer science, in charge of executing the workflow process. The paper focuses on the Workflow Designers' activity and on the description of the software environment that supports their activity, called TMS editor. A Workflow Designer interacts with TMS editor which supports interactive semi-automatic composition of services representing the components (tComponents) at the base of the current workflow. In this way, the Workflow Designer is able to use her/his expertise – which results in her/his tacit knowledge – to steer the design of the workflow document [3]. The TMS editor supports the design of the workflow document, by interacting with the Semantic Search Engine (SSE) to retrieve tComponents and with the Orchestration Engine (OE), to orchestrate these services in order to produce the final workflow document in BPEL4WS language [9]. The TMS editor presents this document to the Workflow Designer through a GUI offering direct manipulation functionalities, i.e. a set of visual widgets and of technical sketches presented using notations familiar to the Workflow Designer. A first prototype of the TMS Editor is currently under development.

2. WORKFLOW COMPOSITION
Two main different approaches to workflow composition through the use of Web services' technologies can be found in literature: automatic [5, 10] or semi-automatic [2, 7]. Automatic composition realizes services' composition exploiting AI algorithms used to plan how different atomic elements could be composed to create the desired workflow. Semi-automatic composition supports users in the task of manually compose and select Web services, offering a graphical interface allowing her/him to manipulate graphical primitives representing the workflow's atomic components. TMS editor follows a semi-automatic approach. Its goal is not to automate the skills of the end users, but to assist them in their activity, permitting them to use the knowledge and expertise they possess [6]. In fact, the Workflow Designers play a fundamental role for what concerns the design and the development of the workflow, and for this

reason they should be directly involved in all of the activities carried out during its whole development cycle. To this purpose, TMS network is designed to allow Workflow Designers – a community of domain experts – and Workflow Operators – a community of end users – to collaborate in the design and the development of a software artifact together with other communities of domain and IT experts. The TMS network has been designed adopting the Software Shaping Workshop (SSW) design methodology [3].

2.1 The SSW Methodology

The SSW methodology adopts a meta-design approach allowing a team of stakeholders (end users, domain experts, HCI experts, software engineers) to cooperate in the design, implementation, use and evaluation of software artifacts, determining their continuous evolution in time. The software artifacts are designed as a network of software environments called SSWs, or briefly workshops, each of them being either an environment through which the stakeholders perform their activities or an environment through which they participate in the design of the whole system, even at use time. The network of workshops is organized in three different levels based on the different types of activities the workshops are devoted to: 1) the use level, lying at the bottom of the network hierarchy, includes *application workshops* that are used by domain experts to tailor and use their application workshops in order to perform their activity; 2) the design level, located at the middle level of the network hierarchy, includes *system workshops* used to perform the collaborative design and development; 3) the meta-design level, at the top of the network hierarchy, includes system workshops used to create and maintain all the workshops in the network, usually by software engineers, but sometimes also by HCI experts and end users. The description of workflows and their executable composition, proposed in this paper, represents a new application of the SSW methodology.

2.2 Workflow Design through TMS

According to the SSW methodology, TMS is a modular system constituted by software environments – application and system workshops (see Figure 1). TMS coordinates stakeholders in the team and the software engines, SSE and OE, to search, acquire, describe, and aggregate services. Such services perform each one single autonomous tasks in the work processes. For lack of space, only the collaborative creation of a workflow is described (Software engineers and HCI experts are not considered in this paper). At meta-design level, domain experts use a system workshop to define the overall structure of the workflow. These domain experts use tools for task analysis to represent in a structured way concepts and activities that constitute the workflow, without referring to implementation issues. At design level, the Workflow Designer uses the TMS editor workshop by which s/he transform the task analysis document in a description of the needed components (tComponents) and the relationships among them. At use level, the Workshop Operators, the end users at this level, use the TMS instance developed at the design and previously described at the meta-design level. Figure 1 illustrates the documents traffic in the network. Downward arrows, show the flow of the documents from the upper level to the lower one. They are, from meta-design level to design level, the documents describing the workflow task analysis, and from design level to use level, the executable descriptions of the TMS instance to be used. The upward arrows, show the flow of communication among the users at the lower level and the (meta-) designers at the upper level.

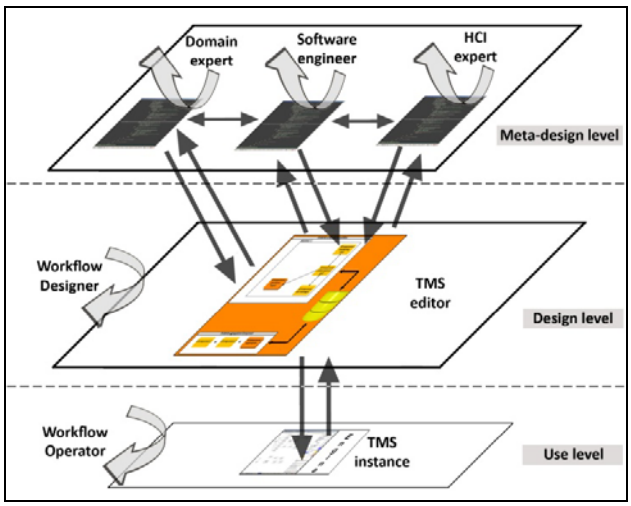

Figure 1: The SSWs network for TMS design, development and use.

All the stakeholders can communicate problems, difficulties or suggestions regarding the workshop in use to the other stakeholders, at the different levels. In this paper, the focus is on the role played by the Workflow Designer, the domain expert that is in charge of building the workflow to be used in her/his organization. The Workflow Designer interacts with the TMS editor by direct manipulation techniques to visually compose the workflow. This composition is semi-automatic, in that the user interface is organized according to her/his tacit knowledge, while the TMS editor interacting with the SSE and OE supports her/him by retrieving and orchestrating the Web services components required to satisfy the Workflow Designer requests.

3. TMS EDITOR ARCHITECTURE

The aim of the TMS editor is to support the design activity of the Workflow Designers allowing them to exploit their tacit knowledge and expertise, without boring them with technical details about the language used to implement, store and transmit the final workflow document. The TMS editor interface is designed so that the Workflow Designers can map their mental models of the activities to be performed into visual commands and widgets [8]. The visual widgets are used to show tComponents and their relations in a graphic way translating the semantic description of the tComponents and their relations into visual forms. On the other side, this semantic description enables a mapping between the WSDL interface of the tComponent with a Conceptual Reference Model (CRM) describing its behavior, goal and functionality. The CRM of a tComponent is a set of classes and properties, expressed in OWL, indicating entities and relationships for describing the developer of the tComponent, its usage field, its inputs and outputs, the interaction style used and a description of its specific algorithmic behavior. Moreover each CRM is extended using specific concepts related to the information domain in which the workflow acts. These classes are used to map tComponents usage field and behavior in specific domain concepts in order to explain, in a semantic way, the function and the role of each tComponent. The mapping between each WSDL description and the related CRM of each tComponent

is carried out with a RDF file while the mapping between the CRM description and the visual widget representing it, is achieved in the TMS editor by means of its visual interface. Therefore, exploiting these visual representations of each tComponent, the Workflow Designer is able to design a workflow through the integration of heterogeneous tComponents in a particular order and under certain conditions. To support this activity the TMS editor has to adopt specific techniques of composition and integration management services. To this end, an Orchestration Engine (OE) translates the composition defined by the Workflow Designer using the TMS editor interface, into a BPEL4WS workflow document describing the correct sequence of operations in the workflow following the structure defined at meta-design by domain experts. The Semantic Search Engine (SSE) retrieves the tComponents useful to compose the final structure of the workflow following the directives of the Workflow Designer interacting with the TMS editor (see Figure 2).

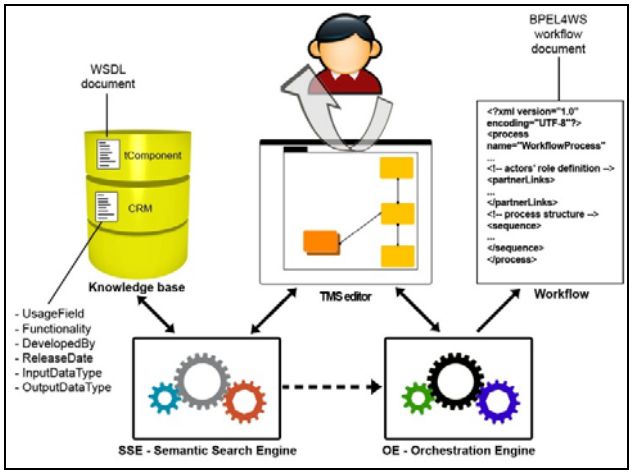

Figure 2: The Workflow Designer-TMS editor system for semi-automatic workflow composition.

These tComponents are gathered by a knowledge base composed by a set of archives made available by various tComponent providers. The tComponents provider, willing to share its tComponents, must first perform a UDDI registration of the tComponents on a tComponents Provider Registry. The registration consists of three different elements:

- a WSDL description of the tComponent to be invoked, describing the syntactical and computational-based aspects about the tComponent;
- a conceptual reference model (CRM) of the tComponent;
- a RDF description that maps the CRM onto the WSDL description of the tComponent.

Interacting with the SSE, Workflow Designers can search the more suitable tComponents, according to the context of use of workflow defined at meta-design. Then they trigger the orchestration activity for producing the final description of the workflow process as a BPEL4WS document. This document is represented on the screen as a sketch of the widgets representing the tComponents and of their relations.

4. CONCLUSIONS

This paper presented a novel network of software environments and a Web service-based architecture aimed at supporting end users of an organization to design, execute, and check the execution of a workflow. The network and the architecture address two communities of end users: the Workflow Designers, who design the workflow and supervise its correct execution, and the Workflow Operators, who execute the workflow process. The paper focuses on the Workflow Designers' activity and on the description of the software environment that supports their activity, called TMS editor. A first prototype of the TMS Editor is currently under development.

5. REFERENCES

[1] Barricelli, B. R., Padula, M., and Scala, P. L. 2009. TMS for Multimodal Information Processing. In Proc. of DEXA Workshops. IEEE Computer Society, Washington, DC, 295-299.

[2] Chinthaka, E., Ekanayake, J., Leake, D., and Plale, B. 2009. CBR Based Workflow Composition Assistant. In Proc. of SERVICES 2009. IEEE Computer Society, Washington, DC, 352-355.

[3] Costabile, M. F., Fogli, D., Mussio, P., and Piccinno, A. 2007. Visual Interactive Systems for End-User Development: a Model-based Design Methodology. IEEE TSMCA 37, 6, 1029-1046.

[4] Lieberman, H., Paternò, F., and Wulf, V. 2006. End User Development. Springer-Verlag New York, Inc.

[5] Pistore, M., Marconi, A., Bertoli, P., and Traverso, P. 2005. Automated Composition of Web Services by Planning at the Knowledge Level. In Proc. of IJCAI 2005. Morgan Kaufmann Publishers, San Francisco, CA, 1252-1259.

[6] Schuler, D., and Namioka, A. 1993. Participatory Design: Principles and Practices. Lawrence Erlbaum Associates, Inc.

[7] Shalil, M., Shields, M., Taylor, I., and Wang, I. 2004. Triana: A Graphical Web Service Composition and Execution Toolkit. In Proc. of ICWS 2004. IEEE Computer Society, Washington, DC, 514.

[8] Valtolina, S. 2008. Design of Knowledge Driven Interfaces in Cultural Contexts. IJSC 2, 4, 525-553.

[9] Web Services Business Process Execution Language: OASIS Standard, Overview, 2007. Retrieved March 9, 2010: http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html.

[10] Wu, Z., Ranabahu, A., Gomadam, K., Sheth, A. P., and Miller, J. A. 2007. Kno.e.sis Center. Technical Report. Wright State University.

[11] Graphic technology -- Database architecture model and control parameter coding for process control and workflow (Database AMPAC), ISO/TR 16044:2004, ver. 2004-06-07.