PEUDOM: A Mashup Platform for the End User Development of Common Information Spaces

Maristella Matera, Matteo Picozzi, Michele Pini, and Marco Tonazzo

Politecnico di Milano
Dipartimento di Elettronica, Informazione e Bioingegneria - DEIB
{matera,picozzi}@elet.polimi.it,
{michele.pini,marco.tonazzo}@mail.polimi.it

Abstract. This paper presents a Web platform for the user-driven, service-based creation of Common Information Spaces (CISs). Two composition environments, characterized by intuitive visual notations, enable i) the integration of services to create UI-rich components and ii) the synchronization of components into interactive workspaces. Collaborative features allow multiple users to collaborate, synchronously and asynchronously, to share and co-create CISs.

Keywords: Collaborative Mashups, End User Development, Common Information Spaces.

1 Introduction

Web 2.0 has accelerated the evolution of the Web, becoming a driver of innovation. End users are now involved in the process of content creation, and this opportunity raised the users' will to become active creators of applications, thus promoting the *mashup* phenomenon. So far, mashups have been conceived as Personal Information Spaces (PISs), i.e., vertical applications solving situational needs, assembled by the end users by integrating ready-to-use resources. Mashups, however, have a great potential to accommodate the sharing and co-creation of knowledge [1]. While collaboration has been extensively investigated in the CSCW (Computer Supported Cooperative Work) area, the co-creation of web-based Common Information Spaces (CISs), especially by means of web mashups approaches, is still scarcely explored. Recent works highlight the need for collaboration [2], but the proposed solutions only cover specific aspects (e.g., awareness in synchronous editing), while they do not offer comprehensive approaches ranging different forms of synchronous and asynchronous collaboration.

1.1 Demo Organization

This demo presents the main ingredients of our approach for the collaborative construction of CISs. We illustrate the collaborative mechanisms introduced in PEUDOM, a platform for the end user development of mashups that

F. Daniel, P. Dolog, and Q. Li (Eds.): ICWE 2013, LNCS 7977, pp. 494–497, 2013.

offers visual paradigms for the creation of UI-rich components and the synchronization of such components to create orchestrated information spaces. The demo introduces the web environments supporting the visual composition of PISs, and the collaborative mechanisms that allow users to make PISs evolve into CISs. The demo also illustrates the models underlying the composition paradigms, the techniques for the automatic generation of application schemas, the pervasive execution of the resulting mashups on different devices to facilitate sharing, and the mechanisms for schema co-evolution in collaborative scenarios. The rest of this paper describes some basic ingredients of our approach that will be also illustrated during the demo. More details on the composition paradigm and the underlying models can be found in [3,4]. A video demonstrating the use of PEUDOM for CIS co-creation is available at `http://home.dei.polimi.it/picozzi/peudom/demoICWE.html`.

2 Models and Tools for PIS Composition

In our mashup platform, component integration complies with an *event-driven, publish-subscribe* paradigm that enables the synchronization of *components*' behaviors. Components wrap (remote or local) services and expose *events* and *operations*. The coupling of components within an integrated workspace is based on the subscription of *operations*, which become *listeners*, for *events* exposed by other components. Subscriptions are expressed in a *composition schema* represented in an XML-based domain specific language [3], which is then used to govern the execution of the PIS, i.e., the synchronization of the different components according to the defined listeners. As shown in Figure 1, a web environment, the *Mashup Dashboard*, enables the creation of PIS schemas. An *Event Handler* on the client-side intercepts the visual composition actions executed by the end-user, and automatically translates them into elements, i.e., listeners and property values, of the PIS schema. Based on the so created schema, the *Mashup Engine* acts as an event bus: it listens to and handles the events raised by the interaction with each single components, and activates the subscribed operations as prescribed by the listeners in the composition schema. A relevant characteristic of our approach is the interleaving of the design and execution phases [3]: users immediately experience the effect of their composition actions (i.e., the composition schema is immediately interpreted and executed); thus they can iteratively and interactively refine the resulting applications.

The platform also provides a web environment for the creation of components (*Component Editor* in Figure 1). Through the *Service Manager* module, it offers support for querying REST services, displaying the retrieved results in a visual format, and visually defining selection and projection queries over such results. A *visual mapping* process indeed allows the user to select data attributes and associate them to user interface elements playing the role of *data collectors*. The association of data from multiple services to a same UI data collector also defines integration queries [4]. The editor, through the *Visual Mapping Manager*, translates the visual actions into an XML-based component schema. The execution of

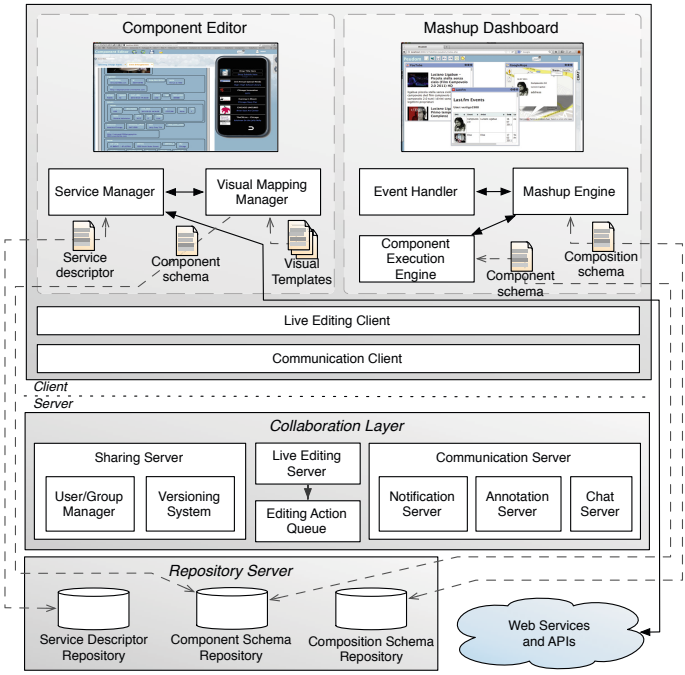

Fig. 1. Platform Architecture

the so created component is then possible within the Mashup Dashboard, as well as standalone apps on different devices where execution engines, coded according to the target technology, interpret and instantiate the component schema.

3 From PIS Composition to CIS Co-Creation

PEUDOM exploits a "lightweight" execution paradigm, hosting all the modules for composing and executing the composite information spaces at the client-side. As highlighted in Figure 1, server side modules instead manage the sharing of resources by multiple users and the synchronous and asynchronous communication. A schema versioning, an annotation system and an activity log system support synchronous communication. Instant messaging and live editing then enable synchronous collaboration. For example, if a user performs an action on a client that modifies locally a shared PIS (i.e., a CIS), this action is propagated (through the *Live Editing Client*) to a *Live Editing Server* in charge of updating the composition schema on each listening client. The server maintains a representation of all the distributed editing actions: every editing session on a CIS is associated with an *Editing Action Queue* from which messages are broadcasted to all the active CIS instances, except the one where the modification originated. On a client listening to modification actions, the Live Editing Client interprets the received actions and actuates the changes on the local schema.

The server-side management of the editing action queue ensures the synchronized evolution of all the active PIS instances. With respect to the paradigm

adopted for PIS construction, the CIS composition schema is now enriched with *status meta-data* (e.g., parameter values to query single components, items selected in a given data set), so that each CIS instance is synchronized not only with respect to the composition structure (i.e., components and listeners), but also with respect to behavioral aspects, e.g., the displayed data set filtered out by different actions of concurrent users. Therefore, the composite application is now long-lasting and stateful: both structure and state variables are maintained across different sessions.

4 Conclusions

Collaboration in mashup-based development can be beneficial in *collective intelligence* scenarios [5], where teams of people co-create knowledge by sharing integrated information spaces with professional peers, in *meta-design environments* [6], where end-users shape up their tools in collaboration with expert developers, or in scenarios where people, not able to develop by themselves their own applications, ask for help and advice from experts within reference communities in a kind of *crowdsourced Web Engineering* [7]. This demo illustrate our solution for the *collaborative development* of CISs. The proposed techniques have been experimented by extending our Web platform for mashup development, but they address elements, such as *services*, *components*, *composition schemas*, that are recurrent in the majority of mashup platforms. Also due to the component-based nature of the collaborative modules, and the customizability of their event-driven logic, we believe the proposed techniques and their supportive software modules can be easily adapted and exploited in the context of other approaches.

References

1. Ardito, C., Costabile, M.F., Desolda, G., Matera, M., Piccinno, A., Picozzi, M.: Composition of situational interactive spaces by end users: a case for cultural heritage. In: Proc. of NordiChi 2012, pp. 79–88. ACM (2012)
2. Heinrich, M., Grüneberger, F.J., Springer, T., Gaedke, M.: Reusable awareness widgets for collaborative web applications - a non-invasive approach. In: Brambilla, M., Tokuda, T., Tolksdorf, R. (eds.) ICWE 2012. LNCS, vol. 7387, pp. 1–15. Springer, Heidelberg (2012)
3. Cappiello, C., Matera, M., Picozzi, M., Sprega, G., Barbagallo, D., Francalanci, C.: Dashmash: A mashup environment for end user development. In: Auer, S., Díaz, O., Papadopoulos, G.A. (eds.) ICWE 2011. LNCS, vol. 6757, pp. 152–166. Springer, Heidelberg (2011)
4. Cappiello, C., Matera, M., Picozzi, M.: End User Development of Mashups. In: Proc. of CHI International. LNCS, Springer (in print, 2013)
5. Grasso, A., Convertino, G.: Collective intelligence in organizations: Tools and studies - introduction. CSCW 21(4-5), 357–369 (2012)
6. Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A.G., Mehandjiev, N.: Meta-design: a manifesto for end-user development. Commun. ACM 47(9), 33–37 (2004)
7. Nebeling, M., Leone, S., Norrie, M.C.: Crowdsourced web engineering and design. In: Brambilla, M., Tokuda, T., Tolksdorf, R. (eds.) ICWE 2012. LNCS, vol. 7387, pp. 31–45. Springer, Heidelberg (2012)