User-driven visual composition of service-based interactive spaces

Carmelo Ardito [a], Maria Francesca Costabile [a], Giuseppe Desolda [a], Rosa Lanzilotti [a], Maristella Matera [b], Antonio Piccinno [a,*], Matteo Picozzi [b]

[a] *Università di Bari Aldo Moro, Dipartimento di Informatica, Via Orabona 4, 70125 Bari, Italy*
[b] *Politecnico di Milano, Dipartimento di Elettronica e Informazione, Via Ponzio 34/5, 20133 Milano, Italy*

Available online 21 February 2014

* Corresponding author.
E-mail addresses: carmelo.ardito@uniba.it (C. Ardito), maria.costabile@uniba.it (M. Francesca Costabile), giuseppe.desolda@uniba.it (G. Desolda), rosa.lanzilotti@uniba.it (R. Lanzilotti), matera@elet.polimi.it (M. Matera), antonio.piccinno@uniba.it (A. Piccinno), picozzi@elet.polimi.it (M. Picozzi).

1. Introduction

Different work contexts and every-day life situations are nowadays characterized by activities where single users or groups of people, through different devices, browse heterogeneous content, capture, synthesize and annotate it to highlight insights and compose it in various ways, in order to create new content and applications. The huge amount of resources available on the Web provide a valuable source of content; but to enable an increasing number of people to make sense of such resources it is necessary to open up the construction of service-based software to non-programmers. Technologies for Web service composition have been proposed since the 1990s in the context of the Service Oriented Architecture (SOA) [1]. More recently, we have seen the proposal of platforms, based on mashup technologies, which claim to be more oriented towards end users [2]. However, this claim has proved to be unrealistic because of the inadequacy, for non-technical people, of the composition languages on which such platforms are based [3–5].

The overall goal of the research presented in this article is to investigate models, methods and architectures for supporting people, who are not software developers and have diverse needs, to co-create Personal Information Spaces (PISs), by integrating heterogeneous contents and artifacts. By PIS we mean an interactive space personalized by its creator that can facilitate access to and manipulation of contents and functionality, since it provides integrated views over disparate, distributed resources. Our goal is in line with the interest, growing both in academia and in industry, in *elastic systems*, able to support paradigms in which applications can be flexibly shaped up at different layers (data, functions and presentation) at use time, based on users' actions in specific usage contexts. One important ingredient to achieve such a flexibility is separation of concerns: "contents, applications, and devices need to be decoupled as much as possible to allow users to focus on information without being confined to a particular pre-packaged application context" [6].

This articlediscussestheaboveissuesand showshowwe addressed them by developing a platform where a visual paradigm enables the lightweight construction of multi-device applications, through which users can easily access, integrate and manipulate information to satisfy their situa-tional needs. We illustrate composition environments where users, by means of "content-exploratory" actions, seamlessly create applications, without the need for distinguishing among what design and execution are (which is typical of professional software design and programming), and are not forced to master technicalities to invoke and integrate data and services (which is typical of service management).

Empowering people, who may not have technical skills, with the possibility of composing content and services is a very critical challenge. The approach presented in this paper builds on recent experience in investigating paradigms for mashup composition [7,8] and on the lessons learnt on End-User Development (EUD) [9–11], and tries to combine the advantages of both fields. This work is the result of an iterative set of experiences, which led us to identify some key points. One is about specializing a platform for mashup composition to a specific domain, thus capitalizing on the knowledge of people working in the domain, in order to offer a composition process that makes sense for a community of users [4]. In [12], we discussed the need for composition approaches to foster EUD and we reported some preliminary studies conducted to assess the adequacy of a composition platform in a specific domain, namely Cultural Heritage by better understanding how the envisioned platform could bring practical value to different stakeholders in the context of visits to sites of cultural interest. Based on these studies, we identified in particular the need for *domain-specific resources*, providing sensible, non-generic content, and for *user interface* (*UI*) *templates*, able to guide the composition, thanks to some basic visual elements that can be easily manipulated by non-technical people to create and modify their PIS. Thus, the significant, new contribution of this paper is a comprehensive methodology for service and data composition by end users, based on a meta-design approach and a novel "stratification" of the composition platform into layers, so that:

- In accordance with *meta-design*, the overall approach enables the involvement of different stakeholders: the first phase (the meta-design phase) consists of designing software environments that allow some stakeholders to create templates, basic elements, and software environments appropriate for end users in the specific application domain; in the second phase, using such environments, end users are able to compose and manipulate their PIS.
- In accordance with the need for *domain-specific approaches*, it enhances customization processes, easing the adoption of the approach by specific communities of end users.
- In accordance with the need for *elastic systems*, it offers an "un-packaged" environment, where end users have the freedom to select the most adequate source of contents, visual templates and execution devices, and the logic to integrate contents deriving from different sources.

The article is organized as follows. Section 2 illustrates the motivations of the overall research work and Section 3 reports related work on service composition. Section 4 illustrates the meta-design approach to PIS composition. Section 5 describes the platform developed, addressing its main components and showing, with examples taken from the case study referring to visits to sites of cultural interest, the adopted visual interaction paradigms and how the platform is customizable to specific application domains. Sections 6 and 7 report some formative studies we have performed to evaluate the platform components devoted to non-technical users. In particular, the study described in Section 7 involved professional guides and visitors of an archeological park. Section 8 finally outlines our conclusions.

2. Motivation

The emerging need for making software systems flexible, so as to increase their ability to support a large variety of tasks, is highlighted in recent works published in the literature. The idea is to replace fixed applications with *elastic systems*, where contents, functionality and access devices are totally decoupled from specific contexts of use and can be determined at runtime; *elasticity* is, in other words, introduced to accommodate multiple and variable contextual needs. New design principles are therefore emerging under the name of *transformative user experience*, to promote paradigms in which end users can access contents and also flexibly use such contents in several situations and across several applications [6,13].

Software design patterns, first of all MVC (Model-View-Controller), already address this separation of concerns. However, the emphasis in the new emerging paradigms is not on design or programming practices to facilitate the development and maintenance of an interactive system; rather they want to stress the opportunity, to be given to end users, to shape up their applications dynamically, depending on their actual needs. Such an approach is especially effective in current scenarios for content access and fruition, characterized, on the one hand, by a data deluge deriving from the availability of a huge number of distributed services and resources, and, on the other hand, by a multiplicity of devices and usage contexts where this multiplicity of data is accessed by users.

In the "participatory" context of the Web 2.0, in the attempt to enable end users to make sense of all the available resources and let them create applications through an easy access to these resources, in the last decade the phenomenon of *Web mashups* has emerged. Web mashups are composite applications, where the "components" are as heterogeneous as SOAP/WSDL Web services, RESTful Web services, RSS/Atom feeds, JavaScript libraries, or simply content extracted (wrapped) from common HTML Web pages (and many more). Web mashup development is characterized by the conjunction of Web languages and technologies enabling the lightweight creation on the client-side, i.e., without the need for ad hoc, server-side execution platforms, of full-fledged Web applications achieved by reusing and integrating ready-to-use resources. Especially when supported by mashup-maker tools [2,8,14–16], to some extent mashup development practices can offer the possibility to create applications with reduced effort. However, the flexibility and elasticity of the systems we mentioned above go well beyond the opportunity to create applications by composing reusable components in different ways. Sensible paradigms are needed to allow users to take advantage of information without being confined to a particular pre-packaged presentation, functionality or application context for the delivery of such information. The new vision, which is gaining momentum in both research and industry contexts, is that the application logics for accessing and managing contents and functionality relevant to the users, together with the presentation layer, e.g., the different ways of presenting results, should actually emerge at run time from the exploratory actions of users. This requires new principles on how data objects (content) interact with their contexts and are transformed as required by the usage situations. It also requires new models and methods for content extraction, integration and reuse, since content and functionality are not to be considered any longer fixed ingredients as in traditionally designed applications.

This trend is also enforced by the emergence of new research lines within HCI, which focus on *appropriation*, *End-User Development* and *meta-design. Appropriation* is the capability of a system to be valid beyond a core set of use cases, able to be adapted even to unexpected uses by the end users (e.g., see [17]). *End-User Development* (EUD) refers to the involvement of end users in the software development process, in order to modify and even create software artifacts [9–11]. EUD activities may go from simple parameter setting to integration of pre-packaged components, up to extending the system by developing new components. The design of systems that enable EUD activities requires a shift in the design paradigm, which must move from user-centered and participatory design to *meta-design*, which literally means "design for designers" [18,19]. This new paradigm allows various stakeholders, including end users, to act as co-designers; thus, software engineers do not design the final application, as in tradi-tional design, but they create software environments through which different stakeholders can contribute to the design of the final application.

Meta-design is characterized by two main phases. The first phase consists of creating the design environments that allow system stakeholders to participate in the design (meta-design phase). Most of these stakeholders are non-technical people. Thus, they use software environments adequate to their skills and to the tasks they have to perform. The second phase consists of the design of the final applications, carried out as joint work by the various stakeholders, who collaborate through their design environments (design phase) [20,21]. Thus, professional developers (software engineers) face new challenges, since they have to create software environments that can in turn empower non-technical people to shape the software they use, without obliging them to become programmers.

Our approach addresses all such lines of action, since it is principally aimed at empowering people to create personalized interactive environments for information fruition. This is also in line with the so-called *cultures of participation*, to which a lot of attention has recently been devoted [22–24]: it promotes a shift from consumer cultures, where produced artifacts are passively consumed, to participatory approaches that greatly exploit computational media to support collaboration and communication, providing users with the means to become co-creators of new ideas, knowledge and products that can satisfy their specific needs [25]. Indeed, we propose a redefinition (and also a seamless fusion) of roles that go beyond the conventional user-designer dichotomy, in a context where system design and system execution are interwoven to let users create, immediately execute and iteratively evolve their own applications.

The solution presented in this paper tries to exploit the technology potential offered by the mashup paradigm and bring it to the end users in a form that can be understood by them. Indeed, the mashup maker tools proposed so far have failed in this respect. This very much depended on the adoption of composition paradigms that did not adequately abstract from technical details, re-proposing to users the same models that programmers adopt for service invocation and orchestrations within platforms unable to capture domain-specific requirements.

3. Composition of service-based interactive spaces

Our work addresses the construction by the end users of Personal Information Spaces (PISs). This term was used in [26] to define an informal space that consists of both artifacts and assigned meanings, constructed, interpreted and manipulated by only one person. In the context of our research, we define PISs as personalized service-based interactive environments, where people, who are not software developers, *integrate*, *manipulate* and *share* heterogeneous contents and artifacts coming from different sources. With respect to the definition reported in [26], our research stresses the service-based nature of the environments created and the collaborative situations where PISs especially enable communication flows and information sharing among different stakeholders. Our definition of PIS is thus close to the one of *Common Information Space* introduced in [27] in relation to collaborative activities. However, we prefer the word *personal* in order to stress that the user is creating his/her own information space.

We now contextualize our approach for PIS construction with respect to some works addressing the composition of service-based interactive spaces. Integration technologies (for workflow composition, service integration, data and application integration) have been around for the last two decades. The adoption of a combination of BPEL (Business Process Engineering Language) and SOA has been also investigated to allow end users to participate in the development of technical workflow models based on their perception of business processes [28]. More recently, Web mashup methods have been proposed for the creation of Web applications starting from reusable Web resources. W3C Widgets have also become the object of mashup composition [29]. Widgets are the result of the (still ongoing) W3C effort to standardize the development of UI components, i.e., full-fledged mini-applications offering functionality and data coming from third party resources, that can be exploited within a Web page. The interest in such component-based technology acknowledges a potential for composition technologies.

What makes mashups different from plain Web service composition is their potential as an alternative solution that can help realize the dream of a programmable Web even by non-programmer users [30]. This is due to the emphasis that such composition technology poses on the *integration at the presentation layer*, focusing on the synchronization of service user interfaces to create rich interactive Web applications. UI integration has not been adequately investigated so far in more traditional fields of service and application integration [31]; instead, integration standards and technologies have been proposed that, however, cannot be mastered by non-technical users [32].

The mashup potential for technology targeting end users is still rarely exploited. So far research on mashups has focused on enabling technologies and standards, with little attention on easing the mashup development process– in many cases mashup creation still involves the manual programming of the service integration. However, there is a considerable body of research on mashup tools, the so-called mashup makers, which provide graphical user interfaces for combining mashup services [15,16,33–36]. With respect to manual programming, such platforms certainly alleviate the mashup composition tasks. However, they still require an understanding of the integration logic (e.g., data flow, parameter coupling, and composition operator programming). Some recent user-centric studies found that the current composition methods are still difficult to use by non-technical users [4]. According to the EUD vision, enabling a larger class of users to create their own applications requires intuitive abstractions, easy development tools and a high level of assistance. Some projects have focused on easing the creation of effective presentations on top of Web services, to provide a direct channel between the user and the service [37,38]. However, such approaches do not allow the composition of multiple services in an integrated application. In some cases, building a complete Web application equipped with a user interface requires the adoption of additional tools or technologies. In [39] the authors propose an easy-to-use environment that allows the users to create full-fledged Web applications by including, within a single Web page, portions of external HTML pages. The paradigm offers support mainly for *Web clipping* activities, i.e. for extracting content from Web pages, while it does not address heterogeneous resources, generally needed by users in several contexts, especially within working communities often referencing domain-specific service ecosystems.
Also the integration capabilities are limited to the definition of parameter cou-plings between input and output components, so that the content of such components is synchronized. This paradigm, however, might be still demanding for end users who, as confirmed by the user study in [4], do not think about "connecting" services.

There are also new composition practices and related execution environments that are emerging around the W3C Widgets. Dedicated execution platforms, the so-called *Widget portals* or *Widget containers* [40], allow users to group within a page miscellaneous pieces of information. Some works have also investigated the use of such technologies for the creation of personal workspaces in specific domains, such as education [41,42] and ERP systems [16]. However, while these platforms make the deployment and execution of widgets easy, they still do not enable the fusion of content and UIs within unified interactive spaces: the resulting Web pages just include collections of non-coordinated contents or functions made available by each single widget (that can possibly be deployed on the local machine), and the customization by end users of such pages consists in the selection of the widgets they are more interested in. iGoogle™ was an example. A few preliminary research works are addressing the integration aspects [43], but the literature still lacks convincing results. Some studies addressed the adoption of widget-based approaches to support end users of ERP systems to create small interactive applications for improving information management in the business administration field [16]. The studies revealed that EUD tools and methods are perceived as useful by end users, but this is still far from the adoption of EUD approaches in an integrated mashup platform.

Another relevant point to be considered is that all the composition platforms so far proposed have tried to be generic (domain-independent), to increase their validity across different domains. This is, however, a weakness rather than a strength: limiting the possibility to customize both the platform functionality and the composition language becomes a barrier for the adoption of such platforms by non-technical people [5].

4. Meta-design approach to PIS creation

Our approach for the lightweight composition of PISs is contextualized within a meta-design approach, based on the Software Shaping Workshop (SSW) model that some of the authors have developed [19,44]. This design model underlines the creation of software infrastructures that support EUD activities and knowledge co-creation by the different stakeholders involved in system design. All stakeholders of an interactive system, including end users, are "owners" of a part of the problem: software engineers know the technology, Human–Computer Interaction (HCI) experts know human factors, graphic designers know how to create an appealing graphical design, domain experts know the application domain and end users know their goals. Most of these stakeholders are non-professional developers. In order to contribute to system design by bringing their own expertise, all these figures need different software environments, specific to their culture and skills. The professional developers involved in traditional design actually become meta-designers, who create software environments, called Software Shaping Workshops (in short SSWs or workshops, intended as a virtual laboratory whose users *shape* software), through which the other stakeholders, acting at some point as designers, contribute to shaping software artifacts. They create and modify elements (objects, functions, user interface widgets) of the system of interest and exchange the results of their activities to converge to a common design and to allow end users to adapt the software to fit their specific needs. In a similar way, various communities of stakeholders are involved in the different phases of the PIS life cycle.

The adopted meta-design approach responds to some user needs observed in our case study in the Cultural Heritage domain. During the last year we have performed systematic contextual enquiries [45], observing the work of professional guides conducting visits in the archeological parks of the Apulia region, in Southern Italy. It emerged that guides could benefit from using PISs, in which they could organize multimedia material to be shown in different phases of the visit through different devices. For example, the guide could make an introduction to the visit by interacting with his/her PIS on a large display available in the hall of the museum associated to the park or in a room. The PIS on a tablet would be used during the tour in the park for providing additional information, such as photos, videos, 3D reconstructions of ancient monuments and references to related sites.

It also emerged that different stakeholders would be involved in the creation and exploitation of PISs. Professional guides of the park are the main end users. Other end users could be the visitors, if the guides decide to share their PIS with them. Professional guides may create and access their PISs using different software solutions running on different devices, as shown at the bottom level of Fig. 1. Our platform currently provides applications for desktop PCs, tablets and large multi-touch displays. These applications are created according to the meta-design approach adopted for the development and customization of the

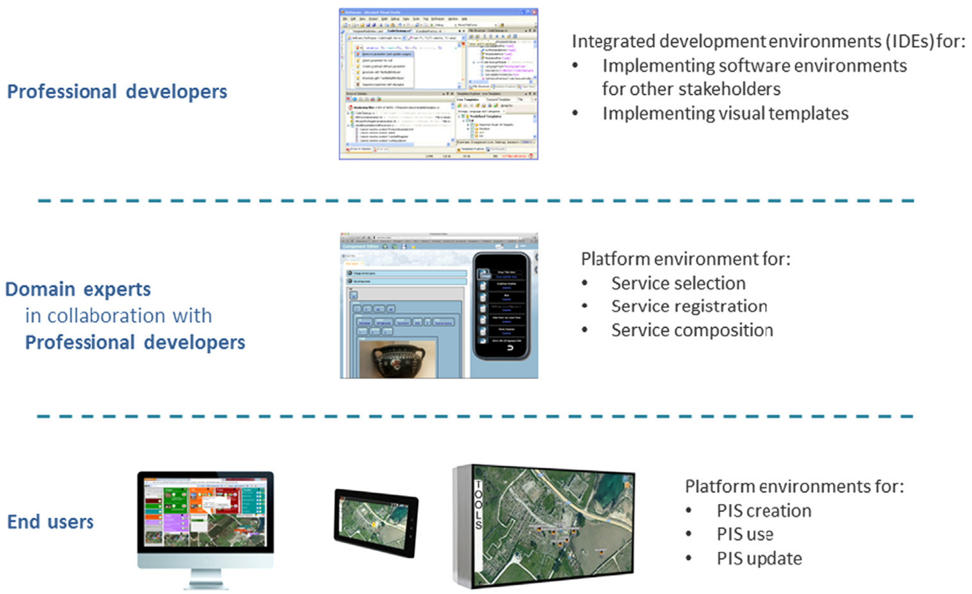

Fig. 1. Meta-design approach to PIS creation. The bottom layer outlines the composition environments for the end users (deployed on different devices) and the middle layer the composition environments for domain experts: the top layer instead highlights the role of professional developers that program the basic elements (service descriptors and visual templates) that the other stakeholders exploit during composition.

composition platform, which in fact offers different *design layers*; at each layer, either activities of meta-design are performed, or a mix of design and use activities, depending on the different stakeholders involved. Indeed, professional developers perform meta-design, since they create the software environments (SSWs) for all the other stakeholders involved in the design and implement and/or modify the software artifacts that require programming efforts (top level in Fig. 1). In order to facilitate the composition process by end users, in our preliminary studies we soon realized that the composition environments (bottom level in Fig. 1) had to be customized to their needs. This introduces another layer of activities to be performed by other stakeholders (middle level in Fig. 1). In our case study, professional developers and Cultural Heritage experts collaborated in meta-design activities to customize the general-purpose tools, by registering relevant services, implementing adequate visual templates and packaging resources for the end users, e.g., the professional guides. Such collaboration is essential for a successful customization. In fact, Cultural Heritage experts are familiar with the types of information the guides would retrieve, the manipulations they would perform and the most suitable visualizations (e.g., a map of the park). However, they do not know how to access the services on the Web that can provide specific information. On the other hand, professional developers, provided they are assisted by the Cultural Heritage experts, are able to set up the service access and also create proper visualizations using Web technologies (for example HTML and JavaScript) or specific languages for other devices (e.g., Java for Android).

The customization to a new application domain is usually performed once; it is possibly updated to satisfy specific needs emerging later, e.g., to register or to combine further services. Thus, the work of more stakeholders is required in order to create environments for non-technical users. This is true in various contexts, as discussed in [46,47]. Some stakeholders perform meta-design activities (e.g., professional developers, service management experts), even if they are non-technical people (e.g. domain experts), since they create environments and tools that allow others to be designers; end users usually perform a mix of design and use activities.

5. The platform for PIS composition and use

To support the meta-design scenario illustrated above, we have developed a platform prototype which extends a general-purpose mashup environment [7] to respond to the need for introducing customized elements (e.g., components, visual templates and composition mechanisms) that are adequate for specific communities of users. The general composition platform is characterized by a "lightweight" composition paradigm, in which direct manipulation of visual elements enables the creation of new resources without any need to program or adopt complicated design notations. As illustrated in Fig. 2, the basic

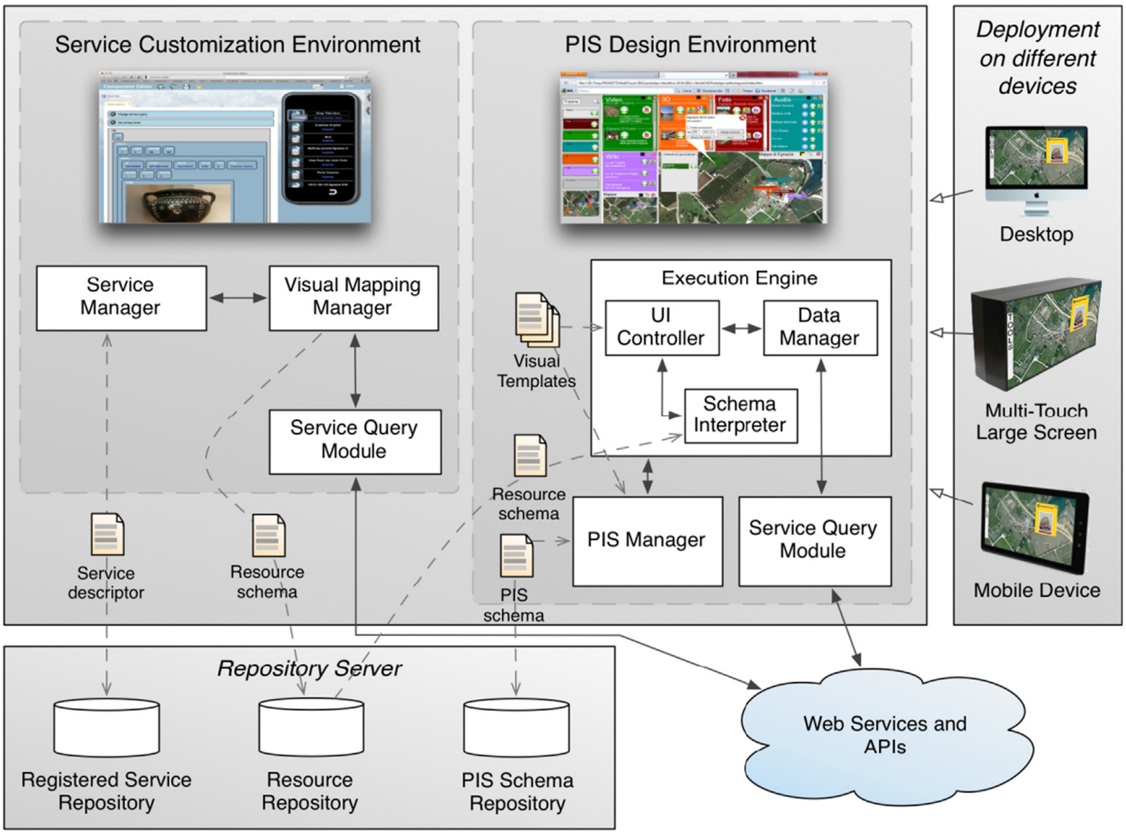

Fig. 2. Architecture of the platform for PIS composition and use.

resources are heterogeneous services, which can be remote Web services and APIs or locally managed data sources and offer contents that can be manipulated and integrated at different levels.

An important component of the platform is the *Service Customization Environment* (SCE), targeting both technology experts and domain experts. Working together and using the SCE module, these stakeholders select and package service-based components that are adequate for specific composition scenarios in a certain application domain and that the end users can fruitfully adopt to create their PISs. Such resources embed the logic, defined by the experts, for querying services, integrating the retrieved results and visualizing the resulting data set.

The descriptors of the resources created through the SCE module are made available to the *PIS Design Environment* (PIS-*DE*), also customized with respect to the characteristics of the target domain, through which end users are enabled to select the contents of interest, as made available by the resources created in the SCE module, and compose their PIS. The PIS schema can then be deployed on different devices.

A feature of our approach is that the two environments, although offering different visual mechanisms for content and service composition, are based on the same conceptual model for resource integration, as well as on similar model-driven mechanisms for the creation of platform-independent schemas that can be executed on multiple devices. In the sequel, we illustrate the two composition environments and describe the modeling abstractions that guide the composition and the execution of service-based artifacts.

5.1. *Service customization environment*

The SCE offers support for querying any kind of distributed resources (Web services or APIs) that return an XML-based or JSON result set. Services need to be registered in the platform by specifying basic properties to invoke the services, such as the service URI and the value of some search keys. Service registration is facilitated by visual forms that guide the user to insert the data needed, so that, even if it is usually performed by technology experts, domain experts or end users themselves could do it with some guidance or after some training.

Service registration produces *Service Descriptors* (see Fig. 2) that instrument a *Service Querying Module* with the service settings that are needed to execute service queries and retrieve an initial result set.

Fig. 3 shows a screenshot of the SCE user interface. It refers to a situation in which, starting from services already registered into the platform, an expert is now exploring the content such services provide. The *data panel* on the left shows the initial result set retrieved by querying, for example, the Flickr service using the

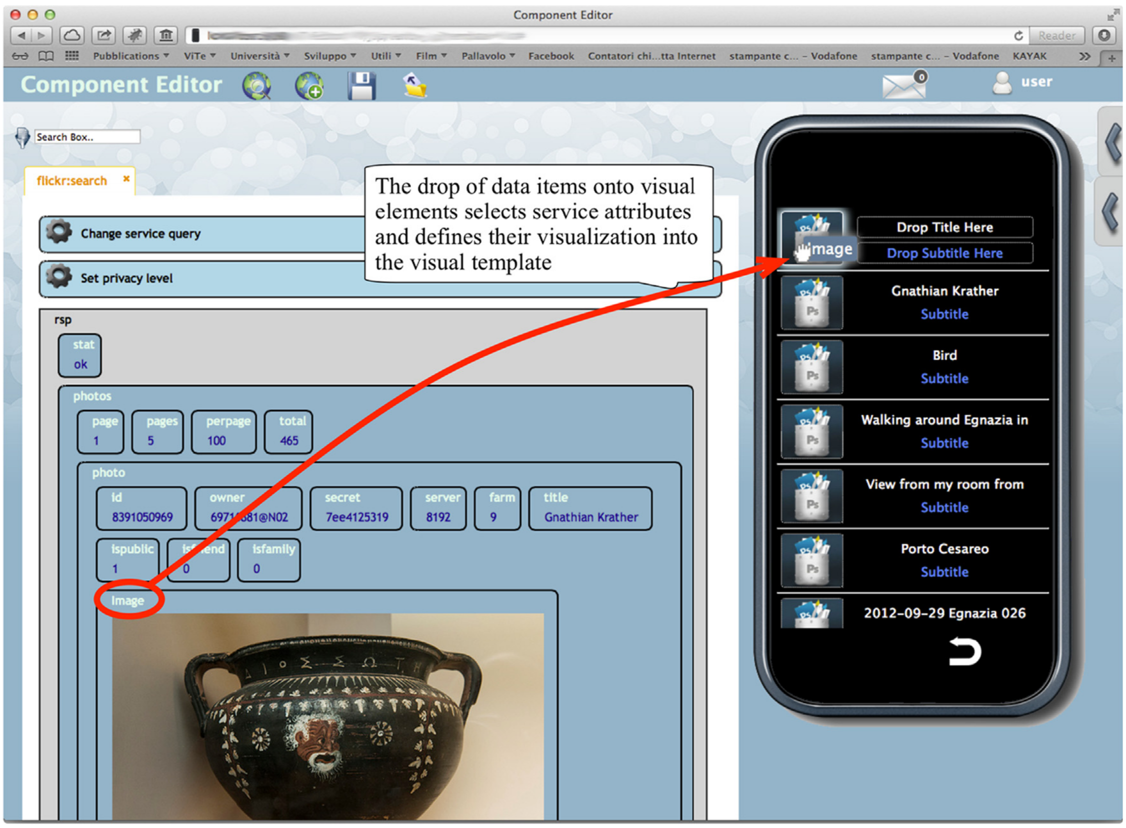

Fig. 3. The service customization environment.

keyword “Egnathia”. It shows that the returned result set is composed of 465 items, each item being characterized by a number of attributes, including also query meta-data, such as an identification name (id), the owner id, the privacy settings and the photo preview.

This visual representation in the data panel hides the technical representation of the service data, natively expressed through Web formats such as XML or JSON. It is meant to facilitate the task of accessing service data, still highlighting the schema of the queried source in terms of data attributes and also presenting examples of instances. Without this facility, the user should write code to invoke the service query and to parse the retrieved data.

The initially visualized result set can be further refined/filtered. The user can, for example, change the initial query by specifying different search keys. Data sets deriving from different services can also be integrated. In this case the result is the definition of *resource local schemas* that will be later used by the composition platform to query each single service and offer the same filtered data, as defined by the domain experts, to the end users composing their final PISs.

The domain expert has now to define how the content returned by this service will be visualized in a proper container visualization to be then adopted in the end user's PIS design environment. In our platform, *visual templates* play the role of container visualization. In the example of Fig. 3, the visual template chosen by the expert is a list-based template, i.e., a list of data items which is particularly effective for a visualization on devices with limited rendering capabilities, e.g., tablets or smart phones.

Using the selected visual template, the expert can *reduce* the initial service schema made of the set of attributes shown in the data panel, by projecting only the attributes of interest. As represented by the arrow depicted in Fig. 3, attribute projection is expressed by moving attributes from the data panel into some notable visual elements of the selected visual template, which we call *visual renderers*. The effect of this mapping is immediately shown in the visual panel: the visual template is filled in with the content items returned by the service for the defined attribute projection. The expert can iteratively modify the attribute choice. In the example of Fig. 3, the expert is selecting, from the original service schema, the “Image”, “Title” and “Description” attributes to be assigned to the image placeholder and the two text fields in the list-based visual template, respectively.

If the selection of attributes is done starting from the result sets of multiple services, the effect of the mapping actions is the definition of union and join queries, providing for *datamediation* operations [48]. In this case, the structure of the visual template and in particular the set of its visual renderers determines a *global* integration *schema* which the reductions of each single service are mapped to.

In general, each visual mapping action results in the definition of a *binding* between a visual renderer and projection or join queries. At low level, queries are then translated into expressions to navigate in the service result sets and, at run time, in executable queries and data fusion procedures if the integration of different result sets is required [48]. The translation of visual actions into corresponding modeling abstractions is operated by the *Visual Mapping Manager* module that, as the result of a service customization session, generates a *resource schema* that specifies:

- a set of *service schemas*, which specify the reduction of the original service schema based on which each single service has to be queried;
- in the case of multiple involved services, a *global integration schema*, defining how different reduced result sets have to be integrated;
- a *user interface schema*, specifying how the integrated data have to be displayed through visual templates in the PIS design environment.

It is worth noting that UI templates are adopted also in other approaches for the composition of service-based interactive applications, but from a different perspective. For example, in the mashup composition approach presented in [49], a so-called *service front end* is a form-based UI module that gives a representation of the technical interface of a Web service and provides the users with the list of parameters expected by the service. The user can specify values for such parameters, depending on the content needed. The resulting application is thus able at runtime to query the service and visualize the results in a tabular template. Our visual templates also offer support to query services, but through a paradigm that seamlessly allows the user to define integrated views over different services. The service front ends in [49] resemble our forms for service registration, which also allow users to define basic service queries. However, our visual templates introduce additional abstractions, which go beyond pure service querying, since they guide the users in a data integration process resulting into integrated visualizations.

5.2. PIS design environment (PIS-DE)

The resource schemas created through the SCE module, assembled within XML-based resource descriptors (see Section 5.3), are interpreted within the PIS design environment to provide end users with the possibility to access the resources previously created. The *PIS-DE* then allows end users, to further filter and compose the contents provided by the resources packaged by the experts. In particular, through PIS-DE the end users synchronize contents with “container” visualizations that can also be customized to their needs.

Fig. 4 shows a screenshot of the PIS-DE for the Cultural Heritage case study. The customization consisted, first of all, of packaging appropriate resources through the SCE module. These resources are made available in the PIS-DE through the so-called *resource windows*. For example in Fig. 4, there is a resource window with videos taken from YouTube, Vimeo, etc. (the green window in Fig. 4); there is also a resource window providing photos that were retrieved from FlickR, Instagram, Google Images, and even from other services. Another resource window provides audios from a local repository of MP3 recorded by the park

Fig. 4. PIS-DE running on a PC.

guides or by Cultural Heritage experts. The guide of the archeological park (a female) can select the resources she wants and, possibly, associate them to specific locations in the park map, which is the visual template chosen as a container visualization. The visual templates, together with the service-based resources ad hoc packaged by the service experts thus constitute elements that address domain specificity.

In the example of Fig. 4, by typing the word "Traiana" in a search box, the guide searches for contents in the available resource windows related to Trajan Way, an important Roman road in Egnathia. The resource windows are therefore synchronized to present content items accordingly, using the visual template defined by the experts during the service customization. For example, on the left of Fig. 4, the Wiki resource window shows the title and a short description of three content items: "La via Traiana", "La via Traiana in Puglia" and "Via Egnazia". The guide has also moved interesting content items into specific positions on the map in the main window. The call-out shown in Fig. 4 refers to a video about Egnathia that the guide is positioning at the top of the map.

In this composition scenario, even if it is at a different level of granularity, the users perform visual mapping actions that correspond to the definition of *bindings* between an element of a container visualization (e.g., a marker on the map characterized by geographical coordinates) and the identifier of a specific content instance to be visualized at that point. In other words, while in the SCE the visual mapping operates at the service schema level (*intensional level*), in the PIS-DE the visual mapping operates over a specific instance (*extensional level*). In fact, as highlighted by the field studies that we conducted in the first phase of our research [8,12], at this level it is important for the end users to dynamically query services, but it is also fundamental to be able to "save" the instances included in the PIS, in order to retrieve the same content across different executions.

The visual composition actions that the end users perform through the PIS-DE are captured and automatically translated into an XML-based, platform-independent model (PIS schema), which specifies the inclusion of contents extracted from the different resources and the binding among such content and the container visual template. The so-created composition model is immediately interpreted and executed on the device where the composition is taking place. Indeed, the PIS-DE also includes some modules for PIS execution that allow the users to immediately observe the effect of their composition actions. End users are therefore able to interactively define their PIS and also be aware of how the final application looks and behaves.

It is worth noting that, in any domain-specific customization of our generic composition platform, the composition paradigm has always been characterized by the intermixing between the design and execution of service-based artifacts: the users can define their compositions, immediately experience the effect of their composition actions and iteratively and interactively refine the resulting applications [7].[1] This choice is also due to the importance that we assign to UI-centric composition paradigms. Even though the final result the user wants to achieve is the integration of service-based resources, the UI becomes a design artefact that evolves according to the exploration and integration actions of the users. It is a *frontier artefact*, which allows the users to express their requirements on data, while also operating on the structure and the behavior of the final applications as actually perceived by them, i.e., observable on the application UI.

At the end of the PIS composition, the PIS schema can be saved in a remote *PIS schema repository*; end users can thus access their created schemas any time and deploy them on different devices. Indeed, the module for the execution of the PIS illustrated above can be run on different devices, e.g., as Web applications, or as native apps compliant with the technologies of the target devices. This is possible thanks to the model-driven logic behind our approach that generates schemas, complying with a Domain-Specific Language (DSL), that can then be interpreted and instantiated on different devices through lightweight execution engines implemented in the device-specific technology.

In the next section, we describe the main abstractions as defined by our DSL. In Section 5.4 we then illustrate how application schemas based on it are interpreted and instantiated on multiple devices.

5.3. The Domain-Specific Language

As we said, *visual templates* are a schematic representa-tion of the final user interface of component resources or of PISs, that act as *containers* for data that the users retrieve from services or resources. Currently, our platform offers visual templates based on lists, maps and charts. In our case study, the services offered to the end users have been customized using a list-based template (see Section 5.1), while the environment offered to the guide uses a *map template*, which exploits the visualization offered by the Google map service to geo-localize data extracted from other resources. In fact, visual templates can be supported by Web services (e.g., maps and charts are supplied by the Google public APIs), or can be ad hoc developed, as happened for the list-based template used in the examples discussed in this paper.

Independently of the adopted concrete visualization that is constructed at runtime, a visual template can be considered a set of *visual renderers*, which represent placeholders for the data selected by the users, that are independent of any specific template. In the “concrete” final UI, visual renderers are then used to render the data associated to them, depending on the chosen template (i.e., a map, a list, or a chart). At execution time, visual renderers are also sources of events (e.g., the selection of a data item), which then trigger the execution of queries on the underlying service and the display of the retrieved data through the visual renderers themselves.

Fig. 5 reports a fragment of the XML-based composition schema that is automatically created when the experts compose resources in the SCE module. The schema fragment refers to a query to the YouTube API, whose results are displayed in a list.

For each data source involved into the composition, the parameters used to query the source at runtime are specified. The visual renderers are then expressed. For each visual renderer, a *data binding* specifying the content to be visualized is expressed by the pair ⟨*data source, path*⟩, representing the source from which data are extracted and the path expression to extract data from the source result set. Such bindings can also associate multiple queries issued to different services to the same visual element.

In line with model-driven approaches (for example those illustrated in [50–54]), we rely on a set of platform-independent abstractions that then drive the instantiation of a running application thanks to model-to-code transformations. Contrary to other model-based approaches to mashup composition, we adopt a UI-centric composition model that provides a way to specify how data should be fused into abstract (i.e., not necessarily bound to a given presentation style) visual elements. In this respect, although we give emphasis to UI elements, our model covers all the essential elements of a composite application: from service access to content integration and content presentation through (in principle) any presentation style on any target device. On the other hand, other model-based composition approaches reflect more the service-based nature of the composite applications, for example, highlighting the definition of event-driven, publish-subscribe listeners to achieve service orchestration [15] or focusing on data mediation operations needed to integrate contents [55].

The three different layers involved in such a model also recall the typical layers of the MVC pattern: a *presentation layer*, expressed in terms of the involved visual renderers of the adopted visual template, represents the View through which the final user can interact with the application. A *service data layer*, expressed through the *source* attribute specified for each visual renderer, provides the *Model*. A *binding layer*, expressed in terms of queries associated to each visual renderer, determines the behavior of the *Controller*, in charge of acting on the Model (e.g., by querying services also applying data fusion techniques [48]) and propagating the achieved results on the View. The MVC pattern is commonly adopted to foster a separation of concerns that can make the design of user interfaces effective. Such a separation of concerns, in our approach, is, however, exploited to enhance flexibility from the end user perspective, letting end users select interesting data (i.e., acting on the Model) and the required integration model (i.e., acting on the Controller) through visual templates (the View) where the Controller logic is triggered. It also has an interesting side effect from the domain specificity perspective, which is related to the

[1] In the case of device limitations, or depending on the specific situation of use, the two phases can be of course detached. For example, for compositions to be executed on smart phones, it would be convenient to execute the composition design on a different device without screen limitations.

```xml
<sources>
 <category name="Videos" icon="...">
 <source name="Youtube" icon="...">
  <query name="GetVideos" label="Get Videos" type="POST"
    url="https://gdata.youtube.com/feeds/api/videos ?">
  <params>
   <param name="q" label="query" type="input">egnathia</param>
   <param name="orderby" label="order by" type="input">published</param>
   <param name="v" label="version" type="input">2</param>
  </params>
  </query>
 </source>
 </category>
</sources>

<visual-mapping template="List" type="video">
 <vr name="title" label="Title" type="text">
  <data source="Youtube" path_type="xpath" path="feed/entry/title" />
 </vr>
 <vr name="thumbnail" label="Thumbnail" type="image">
  <data source="Youtube" path_type="xpath"
       path="feed/entry/media:group/media:thumbnail@url" />
 </vr>
 <vr name="description" label="Description" type="text">
  <data source="Youtube" path_type="xpath"
        path="feed/entry/media:group/media:description" />
 </vr>
 <vr name="url" label="URL" type="text">
  <data source="Youtube" path_type="xpath"
       path="feed/entry/link[rel=self]@href" />
 </vr>
</visual-mapping >
```

Fig. 5. An excerpt of an XML-based composition schema generated by the service customization environment.

possibility of customizing each single layer on which users are entitled to operate without influencing the others.

5.4. *Execution engine*

The execution of the created PIS schema requires a dedicated *execution engine* (EE), i.e., a set of modules able to interpret the composition schema and dynamically instantiate the corresponding application on different target devices. Therefore, going back to our scenario, during the visit the guides can interact with their PISs using an application on a multi-touch display, which supports the briefing phase before the tour. They can also use a tablet application during the tour in the park to show multimedia contents. The execution engine also allows the user to extend or modify the composition. If during the visit the guide needs further contents, she queries the resources by typing a keyword in a search box. The retrieved contents can possibly be added to the PIS, whose model is updated accordingly.

The logic adopted for the instantiation and execution of resources and PISs, as well as for extending the PIS, is the same for all the possible classes of devices. What is different is the technology through which such a logic is implemented and especially the rules to adapt the rendering of the visual templates to the device capabilities. As reported in Fig. 2, a *Schema Interpreter* is in charge of parsing the schema of a PIS. It then invokes the *UI Controller* that, based on the adopted visual template, dynamically generates the user interface. The UI Controller also invokes the *DataManager* module, which in turn queries the involved services based on the specification in the model of the user-defined bindings. The service responses (e.g., represented in JSON) are cached locally on the device. The *UI Controller* finally manages the display of the retrieved data through the visual elements also specified in the bindings.

So far, we have developed EEs for Web browsers, for a multi-touch platform and for Android mobile devices, but, in principle, EEs for any kind of device can be developed. Indeed, the choice of having local EE allows us to exploit efficiently the capabilities of the native technology of each class of device, without forcing the user to access the composed PIS through a Web browser also on devices with a small screen and without requiring the adoption, at code level, of sophisticated rules (e.g., media queries [56] or semantic annotation [57]) to adapt at runtime the presentation of the same Web page on different client devices. The logic to adapt the UI to the different devices is embedded in each EE implementation (one for each class of device). The user downloads the EE once from the platform repository and then uses it for interpreting and executing any PIS schema. Such "local" EEs also offer the additional advantage of exploiting services that are local to the device. This is especially valuable for mobile devices, where, for example, the GPS module can support context-awareness.

Other works adapt the presentations of Web content on multiple devices (see [58] as an example) by adopting HTML templates that, according to *responsive design* mechanisms [59], encapsulate layout rules, for example, to scale the font size or consider different screen sizes. We encapsulated within the EE logic the rules for the display of visual templates on different devices. We also try to go beyond the pure management of layout rendering, since the different EE implementations provide dedicated execution environments where also the applications logic (including the functionality for content access) can be adapted to the different devices. For example, our current implementation of the EE for smart phones uses pagination techniques to limit and distribute along different requests the size of the downloaded result set, given the bandwidth limitations these devices are subjected to and also the limited screen size.

6. Evaluation of the visual composition paradigm

The composition platform has been designed according to a user-centered approach, which prescribes iterative development and prototype evaluation [60]. We performed formative evaluations by triangulating different methods, including studies for collecting feedback in a real context of use. Two usability tests were carried out to evaluate the visual paradigm adopted in the SCE to select and package new resources. Specifically, Section 6.1 illustrates the results of a usability study involving 10 students of Computer Engineering of the Politecnico of Milan. The visual paradigm was chosen so that even non-technical people, i.e., domain experts, could perform customization activities after some short training; thus, a new study was carried out involving 16 people with no technology skills. Both studies analyzed the perceived ease of use of the visual paradigm and the users' satisfaction, without any reference to a specific application domain. As we will see, the scenarios presented in the test address daily life activities that people, in general, are familiar with.

6.1. Usability study involving technical people

6.1.1. Participants and design

A total of 10 people, aged 26–29 years old, were selected at the Politecnico di Milano among the students in the third year of the Computer Engineering curriculum; they had experience in programming and some exposure to Web-service management. The participants were tested one at a time in a quiet research laboratory of the Politecnico di Milano.

6.1.2. Procedure

The test was composed of two sessions: training and test. During the training, one of the two researchers involved gave a 10 min demonstration to introduce the participants to the SCE and its basic composition functions.

In the test session, each participant was given two scenarios, communicated by written instructions. Scenario 1 was simpler and required the participant to perform tasks very similar to those shown during the demonstration. For Scenario 2, the participant had to perform a greater number of steps, manipulating more services.

Scenario 1. In order to organize your free time, you want to keep an eye on the events available in the city of Milano. By exploiting the information available on the Web thanks to the services "Eventful" and "Upcoming", create a map of events. Every event should be accompanied by title, description and address where it will be held.

Scenario 2. You are organizing a trip to Milan. You decide to create a map that displays hotels and metro stations. For each hotel, you would see name and category. For each hotel, you want to see its name and category. For each metro station, you need its name and the routes passing through that station. The map also shows the position and some photos of the Cathedral retrieved from the "FlickR" service.

One HCI researcher observed the interaction and took notes of the difficulties encountered by the participants and of their comments. After the completion of each scenario, the participants were asked to use their composition to check if it satisfied the requirements implied in the scenario. At the end of the test, the participants filled in a questionnaire with closed questions, plus an open question for final comments, in order to evaluate both perceived ease of use of the SCE and user satisfaction.

6.1.3. Results

The *perceived ease of use* was analyzed through quantitative and qualitative data. The quantitative data were collected through four questions in the post-questionnaire; the participants were asked to judge whether they found it easy to identify and include services in the composition and to perform the visual mapping for defining the service query and the visualization of data. The participants rated the perceived ease of use of the SCE on a 7-point scale (1 = very negative/7 = very positive). On average, the ease of use received a positive mark (mean = 5.6, std err = 0.92).

Qualitative data sources to analyze the perceived ease of use were the observers' notes, which included the spontaneous comments the participants made during the test, and the answers to the open question of the questionnaire. The observation indicated that the participants were able to complete both scenarios without difficulties. The average times were 3 min 30 s and 4 min 36 s to carry out Scenario 1 and Scenario 2, respectively. In general, the participants perceived the system easy to use; they all remarked that the adopted visual representations made service management easier than the manual programming they were used to, and also facilitated the interpretation of the result sets which would otherwise be represented, for example, in XML or JSON.

The *perceived satisfaction of use* with the SCE was assessed using a semantic differential scale that required users to judge the method on 12 items (e.g., clear, useful, simple, reliable). The participants could modulate their judgment on each item through a 7-point scale (1=very negative/7=very positive). A satisfaction index, computed as the mean value of the scores across all the 12 items, was 5.2 (std err=0.94). This result was confirmed by the explicit rating participants gave to assess their global satisfaction with respect to the CSE on a 10-point scale (1=very negative/10=very positive). The global satisfaction was high (mean=7.7, std err=0.8). The last two questions asked the participants to judge their performance as composers and to indicate the percentage of requirements, implied in the scenarios, they believed to have satisfied with their composition, based on the observation of the PIS they were able to generate. This metric can be considered as a proxy of confidence [61]. On average, the participants stated they were able to cover 94% of the requirements specified by the two experimental scenarios. They rated their performance as composers with a 4-point scale (1=very negative/4=very positive), and felt satisfied (mean=3.2, std err=0.6).

6.2. *Usability study involving non-technical people*

The second usability test described in this section followed the design and the procedure of the study previously described, since we wanted to verify if actions for retrieving and composing services can possibly be performed even by non-technical people after short training.

6.2.1. *Participants*

Sixteen people with different cultural backgrounds (e.g., students of various disciplines, employees, retired people, etc.) and aged 23–70 years old participated in the study. All of them were familiar with the Internet and the mobile world. They had never carried out activities that required technical competences, such as software programming, design activities, networks set-up, and creation of software applications based on the mashup of Web services.

6.2.2. *Results*

With reference to the *perceived ease of use*, the analysis of the quantitative data revealed that participants' lack of technology skills did not influence their perception on the ease of use of the tool. Similarly to the results of the previous study, on average, the ease of use received a positive mark (mean=5.4, std dev=1.2). Qualitative data showed that only some of them complained about the visualization used to show the service result set in the data panel, which created some difficulties to identify the relevant attributes. However, the observation indicated that the participants were able to complete both scenarios without particular difficulties. The average times were 3 min 30 s and 5 min 20 s to carry out Scenario 1 and Scenario 2, respectively. It is important to highlight that the time spent by the participants of the two studies to carry out the two scenarios was very similar. This suggested that the visual paradigm implemented in the CSE allowed its users, independently of their technology skills, to create their applications in a reasonably short time.

The *perceived satisfaction of use* was assessed as in the previous study. Both the satisfaction index (mean=5.3, std dev=1.1) and the global satisfaction (mean=7.6, std dev=0.8) were high. On average, the participants were very satisfied with their performance related to the coverage of the requirements specified by the two experimental scenarios; they believed that they covered 89% of the requirements implied in the experimental scenarios. They also were satisfied with their performance as composers (mean=3.1, std dev=0.6).

Based on the results of the two usability studies performed we can conclude that the SCE allows its users, with or without technical skills, to easily perform composition activities with satisfaction.

7. Evaluation of PIS creation and use in the CH domain

In the early phases of design and development of the case study, which addresses the creation of PISs to enrich the visit experience at Cultural Heritage sites, a participatory design team was set up. It also included professional guides with a long experience in conducting visits to archeological parks and museums, who regularly create multimedia presentations for the lectures that they give in schools and cultural clubs. The guides were observed while leading several groups of visitors and interviewed to iteratively define and refine user requirements. Some of them had also been involved in the lab evaluations that we had initially conducted to assess the usefulness of our PIS design environment on different devices [12].

A study was conducted in November 2012 at the archeological park of Egnathia to assess the use of a PIS in a real setting. The study was intended as a formative evaluation to obtain insights from the use of current prototypes in the field, highlighting the pros and cons and obtaining insights on the overall approach and on how to improve and extend the platform. Thus, only two guides were involved. We also observed the behavior of the visitors, even though this was not the main goal of our study. The prototypes used in the study were implemented on desktop PCs, large interactive displays and tablets.

7.1. *Participants and design*

The study involved two professional guides, named Achille (male) and Conny (female), and 28 visitors. Both guides formally agreed to participate in the study by signing an explicit consensus. Each guide accompanied a group of 14 visitors in the visit of the Egnathia park. The visitors were people who had booked a visit to the park. They were heterogeneous as regards age (from 21 to 50 years old, plus an 8-year old child), gender and cultural background. They were all Italian but one, a lady from Portugal who currently lives in a nearby city and speaks

Italian very well. These visitors were randomly divided into two groups.

7.2. *Procedure*

The study took place on two days and consisted of two main sessions: (1) PIS composition and (2) park visit. The *PIS composition* session occurred on November 7th, 2012 in the guides' office. The two guides were given a 1-h demonstration of the desktop application, accessible through a PC, to be used to compose the PIS. After this, according to the co-discovery exploration technique [62], the two guides were invited to create together a PIS for visiting the archeological park of Monte Sannace, in the Apulia region. In this way, the guides had the possibility to become familiar with the application. Then, they were asked to create their PIS to be used for the visit of the Egnathia archeological park. The guides individually created their PIS by positioning on an interactive map of the park all the multimedia contents they would like to show to visitors. At the end of the PIS composition session, the guides participated in a design workshop together with a platform designer and two HCI researchers, in order to discuss impressions, problems and possible modifications of the composition mechanisms and the overall system.

On November 17th, the *park visit* session was performed at the archeological park of Egnathia. This session consisted of two different phases: (1) the briefing phase at the beginning of the visit, in which the guide accessed his/her PIS through a large multi-touch display (46-in.) placed at the entrance of the indoor park museum (Fig. 6a) and (2) the tour phase, in which the guide accessed his/her PIS on a tablet (7-in.) during the tour through the remains in the park (Fig. 6b).

First of all, the visitors were informed that they were participating in a slightly different visit with respect to the traditional one, since multimedia materials available on different technological devices were going to be tested. Thus, some pictures would be taken but none would be published in which people's faces could be recognized. They all agreed to participate in the visit.

In the briefing phase, the guide interacted with his/her PIS on the multi-touch display to introduce visitors to the history of Egnathia and to what they were going to see in the park. After this, the tour phase began. During the tour, guides could use their PIS on the tablet to satisfy situational needs, e.g., to show visitors more details about the remains and to answer visitors' questions better by showing specific multimedia contents. In both phases, guides could search for new contents from services and, possibly, update their PIS. The visit session lasted approximately one hour and half.

The two visits occurred in the morning. In the afternoon, another design workshop was carried out involving the same participants as the first one, focusing on the pros and cons of PIS use. Each group of visitors participated in a focus group at the end of the visit, where their impressions on the overall visit experience were discussed.

7.3. *Data collection*

In order to analyze the guides' experience in composing their PIS and using it, data were gathered through naturalistic observation of the guides during: (1) the PIS composition on the desktop PC, (2) the interaction with the created PIS running on the multi-touch display during the briefing phase, and (3) the interaction with the tablet during the tour phase. These data were complemented by the guides' comments gathered during the design workshops after the PIS composition and at the end of the Egnathia park visit.

In the PIS composition session, two HCI experts observed the two guides creating the PIS together: one took notes on paper and the other videotaped all interactions. At the end of the session, they participated in the design workshops together with the guides and the platform designer. The workshop was audio-taped.

Six HCI experts (three experts for each group) followed the park visit session, videotaping and taking notes of the main events. At the end of the visit, the three experts moderated the focus group with the visitors' group they had observed. The focus groups were audio-taped.

a b

Fig. 6. The guides interacting with their PIS: (a) during the briefing phase using the multi-touch display and (b) during the tour using the tablet.

Moreover, the two guides participated in a design workshop, with the same modality and participants of the previous workshop.

The set of notes collected by the experts in the two sessions was substantially extended by video- and audio-analysis. Two researchers transcribed the videos and the audios and independently double-checked 65% of the material. If the inter-rater agreement was less than 70%, the researchers discussed the differences and reached an agreement. Final reliability was high (agreement over 90%).

7.4. *Results*

The results are presented in three different parts, depending on the phase they are referring to: composition phase, briefing phase and tour phase.

7.4.1. *Composition phase*

In this phase, the guides were observed while composing their PIS for visiting the archeological park of Egnathia, using the desktop application. In general, the usability problems they experienced were not so serious to get them stuck; they were in fact able to continue the PIS composition without the help of the HCI experts. Both guides appeared disoriented by the few contents returned by some of the searches they had performed; they tried to refine the search by typing different keywords and, finally, added the most appropriate multimedia materials they retrieved.

At the end of this phase, the two guides participated in the design workshop. As an overall impression, they said they appreciated the ease of use of the application, in particular the possibility to quickly put the retrieved content on the park map. They were rather satisfied by the PIS they had created and they were confident that it would be a valuable support during the visit. Achille jokingly said to Conny: *When this system will be released, I'll call you the day before a visit to ask for suggestions about what to include in my application.*

7.4.2. *Briefing phase*

The briefing aimed at both introducing visitors to the history of Egnathia and providing some preliminary information about the park. The briefing time, during which the guides used the multi-touch display, lasted much longer than in traditional visits, where the briefing to introduce visitors to the archeological park is about 5 min at most. Conny used the multi-touch display for about one third of Achille's time (see Table 1). During his interaction with the multi-touch display, Achille experienced three interaction difficulties due to some technological limitations (Table 1): (1) a temporary loss of Internet connection; (2) he was not able to close the pop-up window by touching the "X" icon, which was located near the display border (our multi-touch device is not very sensitive along its borders); and (3) in the few situations he had to use the virtual keyboard displayed on the screen, due to low precision of the device in correctly detecting the pressed key. Conny had only one problem during her interaction with the display related to the use of the virtual keyboard. However, both guides were able to autonomously manage such difficulties.

Table 1
Use time and interaction difficulties with the multi-touch display.

Variable	Achille	Conny
Use time	28 m 35 s	11 m 45 s
Interaction difficulties	3	1

Table 2
Number of performed searches and modifications of the PIS with the multi-touch display.

Activities	Achille	Conny
Searching new content	4	1
Modifying the PIS	3	0

Both guides appeared quite relaxed in using the multi-touch display. They illustrated the multimedia contents they had previously inserted in their PISs. They were able to search new content without difficulties related to the search functionality. Specifically, Table 2 shows the number of searches and PIS modifications Achille and Conny performed. Achille carried out 4 searches, and only 1 out of 4 was not successful because the retrieved contents did not satisfy his needs. In three cases, Achille believed the retrieved content should have been inserted in his PIS and thus he modified it. Conny performed only 1 search and she did not update her PIS.

It is worth noticing that, when the search for new content required more than 2 min, visitors appeared to be distracted and started chatting among themselves and looking around. Also, in the focus group, some visitors remarked that the position of the multi-touch display generated some problems since, when the guide was interacting with the display, he partially covered it and visitors had to move their heads or their bodies, since they were curious to see all the steps of the interaction.

During the briefing phase, all visitors appeared very interested in the contents illustrated by the guides on the multi-touch display: they asked their guide questions, commented on images among themselves and in general appeared engaged and stimulated by the material shown. This was confirmed in the focus group, in which visitors explicitly expressed their positive opinion about the briefing phase. Nobody complained about this longer phase; on the contrary, they all said: *It's worth it*!. They also said that they would have liked a debriefing phase at the end of the visit, i.e., a phase in which to deepen some topics and possibly look again at the multimedia resources on the multi-touch display, in order to comment with the guide those aspects that had captured their attention during the visit.

7.4.3. *Tour phase*

In the tour phase, the two guides accompanied the visitor group through the remains in the outdoor park. The guides were free to use their tablet as well as the panels located in the park to present the park remains better. Achille and Conny used tablet and panels in different ways (see Table 3). Conny was more prone to the use of such tools; in fact she used the tablet nine times and the panels nine times. In total, she spent 7 min and 53 s commenting the contents available on the tablet, and 3 min and 58 s commenting images on the panels. Achille used such tools very little: he used the tablet once for 1 min and a panel once for 10 s. It is evident that both guides were stimulated to talk more about the contents on the tablet than those on the panels.

Both guides performed searches through the PIS on the tablet. In 3 out of the 9 times in which Conny interacted with the tablet, she performed a search. Only one search did not provide results of her interest. The only time Achille used the tablet was to perform a search. Specifically, the Portuguese visitor said that she loved history and she had visited some Roman archeological parks in Portugal. Achille was very intrigued and started to make searches to understand similarities and differences between the Portuguese sites and Egnathia. From the retrieved images, similarities between the two archeological sites were evident.

During the tour phase, both Achille and Conny did not want to modify the PIS (see Table 4) since they thought that this would require time and, consequently, distract the visitors.

In the design workshop after the park visit, the guides reported that, during the many visits they have performed in their career, very often visitors interrupt them to integrate the guide's presentation with their own knowledge, e.g., history teachers report something they studied, archeologists mention something about a recent discovery in another site, etc. They said that, after the visit, they like to study and analyze more in depth the information given by such visitors, and therefore generally search on the Web, or request material from their colleagues by e-mail or by phone. This modus operandi allows them to enrich their knowledge in preparation for successive visits. The guides clearly remarked that the PIS would improve greatly the acquisition and storage of new knowledge.

Both guides also mentioned that, during the tour, searches requiring more than 2 min interrupted the narrative and distracted visitors. Even though they did not feel uncomfortable during this waiting time, they would have preferred to collect more material without the delay due to the Internet connection, i.e., from local repositories.

Achille and Conny said that they would have liked to use the tablet more, since they appreciated its support in making useful material available. Conny explained that she had used her PIS so little during the tour because she had inserted in it many images that were available on the panels in the park. Thus, she had preferred to show such images on the panels since the tablet was too small for a group of 14 people.

The tablet size emerged as an issue also in the focus groups with visitors, who said that they preferred to look at images on the panels rather than flocking together around the guide to see them on the tablet. They also complained about the brightness of the tablet screen, compromised by external factors, such as sunlight.

Table 3
Frequency and use time of tools.

Tools	Variable	Achille	Conny
Tablet	Frequency	1	9
	Time	1 m	7 m 53 s
Panels	Frequency	1	9
	Time	10 s	3 m 58 s

Table 4
Number of performed searches and modifications of the PIS with the tablet.

Activities	Achille	Conny
Searching new content	1	3
Modifying the PIS	0	0

7.5. Discussion

The objective of this study was to assess the value of the PIS, accessible from different devices, in enhancing visits to archeological parks. To this aim, we analyzed the experience of the two categories of actors involved in the visit: the guide and the visitors. The guide had a double role: (1) designer, i.e., s/he created her/his PIS and (2) end user, i.e., s/he used the PIS during the visit to illustrate the park remains better to visitors. The visitors participated in a visit which was enhanced by the availability of different types of multimedia materials and, thanks to the possibility given by the PIS to search new content, their curiosity might be better satisfied than in a traditional visit.

Composing the PIS with a desktop application did not create particular problems for the guides. They appreciated the support of the composition platform in organizing the material for the visit. However, the guides complained about the scarce material they were able to find when searching the services available in the platform. This is a problem common to all service-based applications, which have to rely either on content made available by a third-party or on user-generated content. To limit this problem, more sensible services should be added to the platform; they can be further third-party services, if any responding to the user needs exist, but they can also be local and ad hoc created collections of contents, maintained by domain experts. Also, given that the services used for the study in the Egnathia park are Web 2.0 resources, the guides could publish online their own material (e.g., videos, pictures, Wikipedia pages) that can be easily accessed through the composition environment. This of course requires a more intensive use of the system by the guides, since they have to realize which material is missing and consequently enrich their public online collections.

Both guides and visitors appreciated very much the briefing phase with the support of the multi-touch display, which appeared to be very valuable in that phase of the visit. The display allows the guides to present much more multimedia materials related to park elements, which enrich their spoken presentation greatly. The visitors' satisfaction is confirmed by their request for a debriefing phase at the end of the visit. As pointed out in several studies, e.g., [63,64], a debriefing phase would be very useful, since it provides the opportunity to deepen and elaborate the information received during the visit, in order to consolidate the acquired knowledge. During the debriefing phase, points of interest that, due to time constraints, were not possible to visit could be quickly illustrated. For example, in the specific case of Egnathia, the necropolis is far from the main city and often it is not visited. In a debriefing phase, the guide could present it by showing pictures or videos; later, people could visit it by themselves if they want to.

The study results showed a general appreciation of use of the multi-touch display in the context of the visit. However, a difficulty was generated by the position of the multi-touch display. It was positioned on a support 110 cm high. For this reason, some visitors could not see the whole display. In future installations, it would be better to use a higher support (at least 150 cm), placing it on a platform at least 50 cm high, which the guide will get on. In this way, the display would be more easily visible to all visitors. However, the fact that the visitors moved to see the display is a symptom of their interest in looking at the material showed by the guide's PIS.

A negative aspect of the use of the PIS on the multi-touch display was the waiting time during a search for new content. This was in part due to the time for typing the search keywords and in part to the low connection speed. However, the search through the PIS design environment is limited to the services registered in the platform (i.e., Flickr, YouTube and Wikipedia). Thus, the search for very specific material can often be unsuccessful, and this might easily bother guides and visitors. As already mentioned, the problem can be reduced by adding further contents to the platform, especially ad hoc collections of contents. Anyhow, it is evident that a search for completely unknown content should be avoided; the search is not the main task to be performed when using the PIS.

Before the study, we expected a larger use of the PIS on the tablet during the tour phase, since it could show images of monuments and other elements of interest, helping visitors to reconstruct the original appearance of such elements and figure out how life used to be in ancient times. Actually, Achille did not show any multimedia content and used the tablet only for one search of a new content. The video analysis revealed that, in a specific situation, Achille exclaimed: *It is a pity that I do not have a picture to show you*! However, he did not consider the possibility of using the tablet to search for the picture. Since in the design workshops he was clearly enthusiastic about the technological tools used, it seems that he would need more time to appropriate these tools. This also holds for Conny. She used the tablet more times but she inserted in her PIS primarily pictures that were also reproduced on the panels in the park rather than additional material that could complement what is already available. It was evident that visitors preferred to look at the images on the panels rather than on the small screen of the tablet, whose visibility is compromised by the sunlight. To overcome this problem, we are going to implement the possibility to visualize some contents of the guide's PIS on the visitors' smart phones.

Moreover, we are also developing new solutions to allow the users to share the PIS and allow others to reuse it. We are in particular investigating to what extent a collaboration paradigm would improve the usefulness of PISs for supporting the cooperation among different stakeholders. The need for collaboration to co-create and share PISs has emerged as a desirable feature in all the user-based evaluation sessions we have performed so far. Based on these new requirements, we have already defined some extensions of our composition platform to enable PIS annotation and PIS co-creation [65].

8. Conclusion and future work

This paper has illustrated how, capitalizing on the synergy between service composition technologies and EUD approaches based on a meta-design paradigm, we have designed and implemented a platform that allows end users, who are not necessarily experts of technologies, to compose Personal Information Spaces (PISs) that satisfy their situational needs and that can be pervasively executed on different devices. Starting from a general-purpose platform for mashup composition, we have identified how some modeling abstractions, which guide the integration of contents within container visualizations, enable different stakeholders to package ad hoc resources and to co-create PISs through intuitive visual paradigms. In particular, end users are facilitated in their composition activities by the availability in the platform of domain-specific resources and user interface templates, which guide their activities by providing basic visual elements that they can easily manipulate. Thanks to the adoption of platform-independent modeling abstractions, the structure of the composed applications is specified in automatically generated schemas that can be deployed on multiple devices, thus promoting the pervasive fruition of the created PISs and also facilitating their sharing [65]. Such a separation of concerns, together with the possibility to extend the platform with ad hoc visual templates and the ease of packaging ad hoc content resources, facilitates the platform customization to specific application domains.

We are currently refining the platform prototype by designing an environment for supporting the WYSIWYG creation of visual templates, which currently have to be manually coded in HTML and JavaScript. Also, since during the field studies in the archeological park we realized that it is difficult for the visitors to access the PIS content when it is displayed on the guide's tablet or on the multi-touch screen, we are defining new mechanisms to "distribute" the execution of the PIS (or even of portions of it) in parallel on different devices. This feature would enhance a lot the PIS sharing in collaborative contexts, and also facilitate information flows between different actors. For example, the guide could chose to publish on the visitors' devices some content, and in turn visitors could enrich the guide's PIS with additional material. This feature, which we observed in our case study in the Cultural Heritage domain, has a general validity in different work communities. Thus, if supported, it may increase the usefulness of our method, and more in general of composition platforms, in several application domains.

Acknowledgments

This work was partially supported by the Italian Ministry of University and Research (MIUR) under the VINCENTE grant and by the Italian Ministry of Economic Development under the LOGIN grant. We are extremely grateful to the director and the personnel of the Egnathia Archeological Park, and to the professional guides Achille Chillà and Conny Blasi who took part in our field study. We also thank organizers and participants of the EUD4Services 2012 workshop for the interesting discussions from which we obtained some useful suggestions, the students that helped us to implement the prototypes, the Editors and the reviewers of the Special Issue for their valuable comments. Finally, we thank Lynn Rudd for her help in revising the English text.

References

[1] G. Alonso, F. Casati, H. Kuno, V. Machiraju, Web Services: Concepts, Architectures and Applications, Springer, Berlin/Heidelberg, 2004, http://dx.doi.org/10.1007/978-3-662-10876-5_9.
[2] F. Daniel, M. Matera, M. Weiss, Next in Mashup Development: User-Created Apps on the Web, IT Professional 13 (2011) 22–29.
[3] N. Zang, M.B. Rosson, What's in a mashup? And why? Studying the perceptions of web-active end users, in: Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing, 2008 (VL/HCC 2008), Herrsching am Ammersee, Germany, 2008, pp. 31–38, http://doi.ieeecomputersociety.org/10.1109/VLHCC.2008. 4639055.
[4] A. Namoun, T. Nestler, A. De Angeli, Conceptual and usability issues in the composable web of software services, in: F. Daniel, F.M. Facca (Eds.), Current Trends in Web Engineering, LNCS, vol. 6385, Springer, Berlin/Heidelberg, 2010, pp. 396–407.
[5] F. Casati, How end-user development will save composition technologies from their continuing failures, in: M.F. Costabile, Y. Dittrich, G. Fischer, A. Piccinno (Eds.), End-User Development, LNCS, vol. 6654, Springer, Berlin/Heidelberg, 2011, pp. 4–6. http://dx.doi.org/10.1007/978-3-642-21530-8_2.
[6] M. Latzina, J. Beringer, Transformative user experience: beyond packaged design, Interactions 19 (2012) 30–33.
[7] C. Cappiello, F. Daniel, M. Matera, M. Picozzi, M. Weiss, Enabling end user development through mashups: requirements, abstractions and innovation toolkits, in: M.F. Costabile, Y. Dittrich, G. Fischer, A. Piccinno (Eds.), End-User Development, LNCS, vol. 6654, Springer, Berlin/Heidelberg, 2011, pp. 9–24.
[8] C. Cappiello, M. Matera, M. Picozzi, G. Sprega, D. Barbagallo, C. Francalanci, DashMash: a mashup environment for end user development, in: S. Auer, O. Díaz, G.A. Papadopoulos (Eds.), Web Engineering, LNCS, vol. 6757, Springer, Berlin/Heidelberg, 2011, pp. 152–166.
[9] A. Sutcliffe, N. Mehandjiev, End-user development (introduction to special issue), Commun. ACM 47 (2004) 31–32. http://dx.doi.org/10.1007/1-4020-5386-X_9.
[10] H. Lieberman, F. Paternò, V. Wulf (Eds.), 9, Springer, Dordrecht, The Netherlands, 2006. http://dx.doi.org/10.1007/1-4020-5386-X_9.
[11] M.F. Costabile, Y. Dittrich, G. Fischer, A. Piccinno (Eds.), Springer-Verlag, Berlin/Heidelberg, 2011. http://dx.doi.org/10.1007/978-3-642-21530-8.
[12] C. Ardito, M.F. Costabile, G. Desolda, M. Matera, A. Piccinno, M. Picozzi, Composition of situational interactive spaces by end users: a case for cultural heritage, in: Proceedings of the 7th Nordic Conference on Human–Computer Interaction: Making Sense Through Design (NordiCHI), ACM, Copenhagen, Denmark, 2012, pp. 79–88, http://dx.doi.org/10.1145/2399016.2399029.
[13] H. Werner, M. Latzina, M. Brade, Symbik – a new medium for collaborative knowledge-intensive work, in: Proceedings of the International Conference on Education, Informatics, and Cybernetics (icEIC), International Institute of Informatics and Systemics (IIIS), Orlando, FL, 2011, http://www.iiis.org/CDs2011/CD2011IDI/ICEIC_2011/PapersPdf/EI924DD.pdf.
[14] Y. Jin, B. Benatallah, F. Casati, F. Daniel, Understanding mashup development, IEEE Internet Comput. 12 (2008) 44–52.
[15] F. Daniel, F. Casati, B. Benatallah, M.-C. Shan, Hosted universal composition: models, languages and infrastructure in mashArt, in: A.H.F. Laender, S. Castano, U. Dayal, F. Casati, J.P.M de Oliveira (Eds.), Conceptual Modeling – ER 2009, LNCS, vol. 5829, Springer, Berlin/Heidelberg, 2009, pp. 428–443.
[16] M. Spahn, V. Wulf, End-user development for individualized information management: analysis of problem domains and solution approaches, in: J. Filipe, J. Cordeiro (Eds.), Enterprise Information Systems, vol. 24, Springer, Berlin/Heidelberg, 2009, pp. 843–857.
[17] A. Dix, Designing for appropriation, in: Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI…But Not As We Know It – Volume 2, British Computer Society, University of Lancaster, United Kingdom, 2007, pp. 27–30.
[18] G. Fischer, E. Giaccardi, Y. Ye, A. Sutcliffe, N. Mehandjiev, Meta-design: a manifesto for end-user development, Commun. ACM 47 (2004) 33–37, http://www.scopus.com/record/display.url?eid.
[19] M.F. Costabile, D. Fogli, P. Mussio, A. Piccinno, Visual interactive systems for end-user development: a model-based design methodology, IEEE Trans. Syst. Man Cybern. A 37 (2007) 1029–1046.
[20] M.F. Costabile, P. Mussio, L. Parasiliti Provenza, A. Piccinno, Supporting end users to be co-designers of their tools, in: V. Pipek, M. B. Rosson, B. de Ruyter, V. Wulf (Eds.), End-User Development, LNCS, vol. 5435, Springer, Berlin/Heidelberg, 2009, pp. 70–85.
[21] C. Ardito, P. Buono, M.F. Costabile, R. Lanzilotti, A. Piccinno, End users as co-designers of their own tools and products, J. Vis. Lang. Comput. 23 (2012) 78–90.
[22] G. Fischer, Understanding, fostering, and supporting cultures of participation, Interactions 18 (2011) 42–53.
[23] H. Jenkins, Confronting the Challenges of Participatory Culture: Media Education for the 21st Century, MIT Press, Cambridge, MA, 2009.
[24] D. Díez, A. Mørch, A. Piccinno, S. Valtolina, Cultures of participation in the digital age: empowering end users to improve their quality of life, in: Y. Dittrich, M. Burnett, A. Mørch, D. Redmiles (Eds.), End-User Development, LNCS, vol. 7897, Springer, Berlin/Heidelberg, 2013, pp. 304–309, ISBN: 0262513625.
[25] J. Porter, Designing for the Social Web, New Riders Press, Thousand Oaks, CA, 2008, ISBN: 0321534921.
[26] C. Tang, S. Carpendale, An observational study on information flow during nurses' shift change, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI), ACM, San Jose, California, USA, 2007, pp. 219–228, http://dx.doi.org/10.1145/1240624. 1240661.
[27] L. Bannon, S. Bødker, Constructing common information spaces, in: Proceedings of the 5th Conference on European Conference on Computer-Supported Cooperative Work (CSCW), Kluwer Academic Publishers, Lancaster, UK, 1997, pp. 81–96, ISBN: 0-7923-4638-6.
[28] A. Boden, C. Dörner, S. Draxler, V. Pipek, G. Stevens, V. Wulf, Tangible and screen-based interfaces for End-user Workflow Modeling, Software, IEEE, PP (2013) 1. http://dx.doi.org/10.1109/MS.2013.71.
[29] M. Cáceres, Packaged web apps (widgets) – packaging and XML configuration, 2nd ed., W3C Recommendation, 2012 ⟨http://www.w3.org/TR/widgets/⟩ (accessed 15.02.13).
[30] E.M. Maximilien, H. Wilkinson, N. Desai, S. Tai, A Domain-Specific, Language for web APIs and services mashups, in: B. Krämer, K.-J. Lin, P. Narasimhan (Eds.), Service-Oriented Computing – ICSOC 2007, LNCS, vol. 4749, Springer, Berlin/Heidelberg, 2007, pp. 13–26.
[31] F. Daniel, J. Yu, B. Benatallah, F. Casati, M. Matera, R. Saint-Paul, Understanding UI Integration: A Survey of Problems, Technologies, and Opportunities, 2007, 59–66.
[32] A. Ro, L.S.-Y. Xia, H.-Y. Paik, C.H. Chon, Bill organiser portal: a case study on end-user composition, in: Proceedings of the International Workshops on Web Information Systems Engineering, Springer-Verlag, Auckland, New Zealand, 2008, pp. 152–161, http://dx.doi. org/10.1007/978-3-540-85200-1_18.
[33] R. Ennals, E. Brewer, M. Garofalakis, M. Shadle, P. Gandhi, Intel mash maker: join the web, SIGMOD Rec. 36 (2007) 27–33.
[34] T. Copeland, Presenting archaeology to the public, in: T. Merriman (Ed.), Public Archaeology, Routledge, London, UK, 2004, pp. 132–144.
[35] Yahoo! Inc., YahooPipes, 2007 ⟨http://pipes.yahoo.com/pipes/⟩ (accessed 22.02.13).
[36] J. Wong, J.I. Hong, Making mashups with marmite: towards end-user programming for the web, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, San Jose, California, USA, 2007, pp. 1435–1444, http://dx.doi.org/10.1145/1240624.1240842.
[37] R. Krummenacher, B. Norton, E. Simperl, C. Pedrinaci, SOA4All: enabling web-scale service economies, in: Proceedings of the 2009 IEEE International Conference on Semantic Computing, IEEE Computer Society, Berkeley, CA, USA, 2009, pp. 535–542, http://dx.doi.org/10.1109/icsc.2009.46.

[38] J. Spillner, M. Feldmann, I. Braun, T. Springer, A. Schill, Ad-hoc usage of web services with dynvoker, in: Proceedings of the 1st European Conference on Towards a Service-Based Internet, Springer-Verlag, Madrid, Spain, 2008, pp. 208–219, http://dx.doi.org/10.1007/978-3-540-89897-9_18.

[39] G. Ghiani, F. Paternò, L.D. Spano, Creating Mashups by Direct Manipulation of Existing Web Applications, in: M.F. Costabile, Y. Dittrich, G. Fischer, A. Piccinno (Eds.), End-User Development, LNCS, vol. 6654, Springer, Berlin/Heidelberg, 2011, pp. 42–52.

[40] D. Griffiths, M.W. Johnson, K. Popat, P. Sharples, S. Wilson, The wookie widget server: a case study of piecemeal integration of tools and services, J. Univer. Comput. Sci. 18 (2012) 1432–1453.

[41] D. Griffiths, M.W. Johnson, K. Popat, P. Sharples, S. Wilson, The educational affordances of widgets and application stores, J. Univer. Comput. Sci. 18 (2012) 2252–2273.

[42] A. Soylu, F. Mödritscher, F. Wild, P. De Causmaecker, P. Desmet, Mashups by orchestration and widget-based personal environ-ments: key challenges, solution strategies, and an application, Program: Electronic Library and Information Systems, vol. 46, , 2012, 383–428.

[43] S. Wilson, F. Daniel, U. Jugel, S. Soi, Orchestrated user interface mashups using W3C widgets, in: A. Harth, N. Koch (Eds.), Current Trends in Web Engineering, LNCS, vol. 7059, Springer, Berlin/Heidelberg, 2012, pp. 49–61.

[44] M.F. Costabile, D. Fogli, P. Mussio, A. Piccinno, End-user development: the software shaping workshop approach, in: H. Lieberman, F. Paternò, V. Wulf (Eds.), End User Development, vol. 9, Springer, Dordrecht, The Netherlands, 2006, pp. 183–205.

[45] H. Beyer, K. Holtzblatt, Contextual Design: Defining Customer-Centered Systems, Morgan Kaufmann, San Francisco, 1998.

[46] F. Cabitza, D. Fogli, A. Piccinno, "Each to His Own": Distinguishing Activities, Roles and Artifacts in EUD Practices, in: L. Caporarello, B. Di Martino, M. Martinez (Eds.) Smart organizations and smart artifacts - fostering interactions between people, technologies, and processes, LNISO, Springer, Berlin/Heidelberg, in print.

[47] D. Fogli, A. Piccinno, Co-evolution of end-user developers and systems in multi-tiered proxy design problems, in: Y. Dittrich, M. Burnett, A. Mørch, D. Redmiles (Eds.), End-User Development, LNCS, vol. 7897, Springer, Berlin/Heidelberg, 2013, pp. 153–168.

[48] C. Cappiello, M. Matera, M. Picozzi, End-user development of mobile mashups, in: A. Marcus (Ed.), Design, User Experience, and Usability. Web, Mobile, and Product Design, LNCS, vol. 8015, Springer, Berlin/Heidelberg, 2013, pp. 641–650.

[49] T. Nestler, A. Namoun, A. Schill, End-user development of service-based interactive web applications at the presentation layer, in: Proceedings of the 3rd ACM SIGCHI Symposium on Engineering Interactive Computing Systems, ACM, Pisa, Italy, 2011, pp. 197–206, http://dx.doi.org/10.1145/1996461.1996521.

[50] S. Trewin, G. Zimmermann, G. Vanderheiden, Abstract representations as a basis for usable user interfaces, Interact. Comput. 16 (2004) 477–506.

[51] J.M.C. Fonseca, W3C Model-Based UI XG Final Report, W3C Incubator Group, 2010, ⟨http://www.w3.org/2005/Incubator/model-based-ui/XGR-mbui-20100504/⟩ (accessed 25.02.13).

[52] F. Paternò, C. Santoro, L.D. Spano, Model-based design of multi-device interactive applications based on web services, in: 12th IFIP TC 13 International Conference on Human–Computer Interaction: Part I, Springer-Verlag, Berlin/Heidelberg, 2009, pp. 892–905, http://dx.doi.org/10.1007/978-3-642-03655-2_98.

[53] F. Paternò, C. Santoro, L.D. Spano, MARIA: A universal, declarative, multiple abstraction-level language for service-oriented applications in ubiquitous environments, ACM Trans. Comput. Hum. Interact. 16 (2009) 1–30.

[54] M.F. Costabile, D. Fogli, A. Marcante, P. Mussio, L. Parasiliti Provenza, A. Piccinno, Designing customized and tailorable visual interactive systems, IJSEKE 18 (2008) 305–325.

[55] Open Mashup Alliance (OMA), OMA EMML Documentation, 2013 ⟨http://www.openmashup.org/omadocs/v1.0/⟩ (accessed 25.11.13).

[56] E.J. Etemad, Cascading Style Sheets (CSS) Snapshot 2010, W3C Working Group Note, 2011 ⟨http://www.w3.org/TR/CSS/⟩ (accessed 25.11.13).

[57] Y. Yesilada, R. Stevens, S. Harper, C. Goble, Evaluating DANTE: semantic transcoding for visually disabled users, ACM Trans. Comput. Hum. Interact. 14 (2007) 14.

[58] M. Nebeling, F. Matulic, L. Streit, M.C. Norrie, Adaptive layout template for effective web content presentation in large-screen contexts, in: Proceedings of the 11th ACM Symposium on Document Engineering, ACM, Mountain View, CA, USA, 2011, pp. 219–228, http://dx.doi.org/10.1145/2034691.2034737.

[59] E. Marcotte, Responsive web design, A Book Apart, 2011, ISBN: 978-0984442577.

[60] ISO/IEC 9241-210, Ergonomics of Human–System Interaction—Part 210: Human-centred Design for Interactive Systems, 2010.

[61] K. Hornbæk, Current practice in measuring usability: challenges to usability studies and research, Int. J. Hum. Comput. Stud. 64 (2006) 79–102.

[62] J.A.M. Kemp, v.T. Gelderen, Co-discovery exploration: an informal method for iterative design of consumer products, in: P.W. Jordan, B. Thomas, B.A. Weerdmeester, I.L. McClelland (Eds.), Usability Evaluation in Industry, Taylor and Francis, London, 1996, pp. 139–146.

[63] C. Ardito, M.F. Costabile, A. De Angeli, R. Lanzilotti, Enriching exploration of archaeological parks with mobile technology, ACM Trans. Comput. Hum. Interact. 19 (2012) 1–30.

[64] C. Ardito, R. Lanzilotti, M.F. Costabile, G. Desolda, Integrating traditional learning and games on large displays: an experimental study, Educ. Technol. Soc. 16 (2013) 44–56.

[65] C. Ardito, P. Bottoni, M.F. Costabile, G. Desolda, M. Matera, A. Piccinno, M. Picozzi, Enabling end users to create, annotate and share personal information spaces, in: Y. Dittrich, M. Burnett, A. Mørch, D. Redmiles (Eds.), End-User Development, LNCS, vol. 7897, Springer, Berlin/Heidelberg, 2013, pp. 40–55.