

# Process-oriented Enterprise Mashups

Paul de Vrieze\*, Lai Xu\*, Athman Bouguettaya<sup>†</sup>, Jian Yang<sup>‡</sup> and Jinjun Chen<sup>§</sup>

*\*SAP Research Switzerland*

*St. Gallen, Switzerland*

*Email: {pdevrieze, laixu}@acm.org*

*<sup>†</sup>CSIRO ICT Centre*

*Canberra, ACT, Australia*

*Email: athman.bouguettaya@csiro.au*

*<sup>‡</sup>Department of Computing*

*Macquarie University*

*Sydney, NSW, Australia*

*Email: jian@comp.mq.edu.au*

*<sup>§</sup>Faculty of Information & Communication Technologies*

*Swinburne University of Technology*

*Melbourne, Victoria 3122, Australia*

*Email: jchen@swin.edu.au*

## Abstract

*Mashups, a new web 2.0 technology provide the ability for easy creation of web-based applications by end-users. The uses of the mashups are often consumer related. In this paper we explore how mashups can be used in the enterprise area and hat the criteria for enterprise mashups are. We provide categories for the classification of enterprise mashups, and based upon a motivating example we go further in depth on business process enterprise mashups.*

## 1. Introduction

Being not only useful for integrating legacy systems and permitting the separation of concerns by founding the basic building blocks of functionality, service-oriented paradigm provides a new way of designing distributed system that encompasses its own design principles, design pattern catalogs, pattern languages, a distinct architectural model, and a set of associated technologies and frameworks [1]. Service-orientation provides a way of thinking of business process management (BPM) in terms of computational infrastructures, services, service-based development and outcomes of those services.

The service-oriented architecture (SOA) paradigm specifies a set of loosely coupled services whose interfaces are published, discovered and invoked over the Internet along with their interactions. It promises a more flexible IT infrastructure that is able to react to business changes more quickly than before. SOA aims to enhance efficiency, agility, flexibility and productivity by positioning services as the primary atomic functional elements. Despite of its benefits, very few companies actually offer their (internal)

services to others at the moment. However, this situation is changing. Companies such as Amazon, Google, Yahoo and eBay are exposing their IT services over the Web. Their customers can integrate these services in order to build new applications in a “mashup” fashion.

Web services have been proven very popular in industry. It has become a layer of abstraction over legacy hardware and software platforms. A Web service is an interface that describes a collection of operations that are network-accessible through standardized Web protocols, and whose features are described using a standard XML-based language [2].

Widgets are small, simple applications that allow the user to perform common tasks or monitor important information. It allows the user to monitor or interact with an application without needing to launch the full application. Widgets are designed to use XML and a scripting language (ie. javascript) to enable developers to build small, simple, downloadable applications that can monitor information about the user’s computer, or Internet-based items of interest like the weather, stock prices, etc. Widgets typically run on a run-time engine, such as the Yahoo! Widget Engine, Microsoft’s Windows Vista gadget engine, or dashboard widgets of Apple Macintosh.

Web 2.0 has gained increasing interest and attention, and led to development and evolution of Web culture communities and hosted services. Web 2.0 technologies have further simplified user contributions and released the power of cooperation with efforts. Web 2.0 has popularized concepts such as mashups and syndication though technologies such as RSS, Atom. They provide new means and possibilities for business networking and business flexibility. Web 2.0 facilitates rich user experience and user participation.

User participation implies openness, freedom and collective intelligence. Web 2.0 also supports dynamic content and metadata. Web standards and scalability are further provided by Web 2.0 [3].

On-line mashups nowadays have evidently captured interest and continue to grow in popularity. They have the capability to combine data/information with an application to create and produce conclusive and useful output. The term “mashup” implies easy, fast integration, frequently made possible by access to open APIs and data sources to produce results data owners had no idea could be produced. Mashups are data aggregation applications, combining data from different sources to create valuable information. Mashup users use, loosely defined “Web 2.0” techniques to effect these results. Mashup applications can be built using different widgets, open APIs, web services, and data sources.

We have seen many examples of lightweight integration (mashups, gadgets and pipes), from the time-honored stock ticker, to e-commerce sites that combine UPS or FedEx tracking data with an order history to present a single view of order status. Inside the enterprise, portal server vendors, including IBM and Plumtree, have long offered users graphical tools to integrate data sources, resulting in simple, personalized Web applications.

There are also mashups using Web traffic and other analytics data to create visual reports for customers. Or mashups to give employees access to the repair, order, and service history for any part, integrating a half-dozen data sources. In the case of real-estate agents mashups can also be used to integrate client data from the CRM system within agents’ e-mail, allowing them to view reports of current property matches from e-mail alerts. Although attractive for lightweight, rapidly developed applications, current mashups also have limitations such as a very strong focus on data integration. At least, mashups currently make sense for 80 percent of noncritical IT processes and logic, no company would rely on mashups for critical information.

From all above examples of mashups, we can see that today’s mashups are primarily focused on data integration instead of creating processes. Enterprise mashups on the other hand have different usage and types. We see four types of enterprise mashups:

- 1) Business process mashups, which coordinate different process orchestration events.

Business process mashups allow a user to automate his tasks. When a user needs to perform a task this often involves getting information from one place, aggregating, filtering, and shaping it, and then sending the result to a different place, and then again until some time when the process is finished.

When performing a purchase this for example ends up into: requesting quotes from a number of suppliers. When each quote comes in, verify that it satisfies

price and product conditions. When not enough good quotes are received after a period of time, request more quotes from different suppliers. When enough quotes are received, providing the three best quotes to the decision maker, who selects one. Order the product from the selected supplier, and finally when the invoice arrives, verify the costs with the quote, and pay the supplier.

Many of these steps can be automated relatively simple, such that those elements which do not require human intervention by the user can be done automatically, or, if wanted, with the click of a single button. A business process mashup allows the user to specify this execution himself, and it will be executed as if he performed the tasks manually. This approach is different from traditional workflow management approaches in that the mashup is focussed on a single user/role, and the actions are performed like they would if they were done manually. These process mashups further live at a lower level in the process, more focussed on concrete tasks than abstract tasks that must be decomposed into other, more detailed tasks.

- 2) Front-end mashups, which are typically done by technical leads in lines of business and include information embedded in the page and visualization of related data sources. Some examples of front-end mashups are ones that overlay sales data on a map and pair inventory data with customer orders.

Front-end mashups are primarily visualisation tools. They perform business intelligence tasks by retrieving data from various sources, processing this, and finally visualising the results. These mashups allow users to build their own tools for monitoring the business process, or to get precise insight in particular parts of the business. As the need for information changes fast, the specific value is often not clear, and the information can often be gotten in different ways (such as spreadsheets) as well, the roundtrip to a development department is often a too big barrier for visualisation tools to be developed when not done by the end-user.

- 3) User interface mashups. These mashups function as user interfaces for data entry. Many of the tasks in a business process actually require the entry, editing, or approval of data. When the type of the data is known, appropriate entry controls can easily be generated, where the user creates the widget to place the controls in a convenient way, or to select alternative controls that have added value. An example of such a control would be a control that would show the picture of products. In a mashup for customer support, it might be valuable to match a product number to this picture to allow the representative a clearer understanding of

the product which has the issue.

- 4) Knowledge mashups. Knowledge mashups are mashups that are used to open up the knowledge that exists within an enterprise. A continuous problem within large enterprises is the management of knowledge. People in one department often don't know what members of other departments do. Knowledge mashups incorporate social features that allow different users to collaborate on opening up knowledge sources.

When thinking about knowledge mashup, think about tools that identify what people with similar roles access within the various knowledge bases within the company. Or maybe what contacts from related departments look at. Other applications of these mashups would help with data synchronization between various parts of the enterprise information system landscape.

In this paper, we focus on the second type of enterprise mashups – business process mashups. These business process mashups can coordinate different process orchestrations and support process-oriented enterprise situations. By lowering the barriers to business process management, they have a high potential to be of significant business value for small and mid-size enterprises.

- Relations between mashup and Web service composition.

Mashups have a broader range of functionality than web service compositions. Composing services is only a part of what mashups do. For example service compositions do not have an independent instantiation. They are initiated by calling the composed service. Mashups on the other hand, especially business process mashups, can have active parts that can monitor the environment or be notified of events.

Mashups, being broader in scope than web service compositions, actually differ most in the area of approach. A mashup is the result of fast development on a small scale. In the unix world, scripts are used for various tasks within the system, and for administration. These tasks can be performed by complete programs, but for many tasks, scripts are more suitable and convenient. Mashups can be compared with scripts, where web service compositions are programs. The capabilities (looking only at web service mashups) are similar, but the strengths of the approaches are in different areas.

Currently there are already many interesting Web 2.0 services and applications, but only in the data integration level [4]. The idea for a business process mashup is to provide process integration to these services and applications.

- Relation between business process mashup and traditional workflow systems.

The relation between business process mashups and

traditional workflow systems is also clear. Traditional workflow systems provide user a stable system more reflect to routine. Business process mashup can be complement of a traditional workflow system. Because of agile ability, a business process mashup can reflect to change better, changes of user requirements, changes of process related resources, etc. It should be cheaper and easier to use. Because of the use of Internet as the application basis, it may also includes accessibility problems, security problems and so on. Business process mashups are on the very beginning of their journey, and have a long way to go.

## 1.1. Related work

Current mashups, such as Google Mashup Editor [5] and Yahoo Pipes [6], are mainly data-driven Web applications which mainly focus on seamless end user programming without requiring any programming experience from the end users [7], [4]. Swasup provides a domain-specific language (DSL) for creating mashups [8]. The DSL is implement in Ruby on Rails and provides concepts that let users efficiently creat mashup: multiprotocol service and data access, mediation support, and a means for generating a UI for the resulting mashup [9]. F.Curbera and his colleagues present another mashup tool Bite, a lightweight composition support acting as a foundation for implementing workflows that can integrate UI on top of it [10] [11]. The business process mashup we propose is even more traditional like workflow systems, but much more agile.

In this paper we will discuss what further needs to be added to extend the current data mashups to become business process mashups. We first look at a motivating example for process-oriented enterprise mashups, its models and available Web services, feeds, widgets, gadgets, pipes, mashups, and APIs. We analyze the characteristics of process-oriented enterprise mashups in Section 3. Key issues of process-oriented enterprise mashups will be discussed in Section 4. The paper ends with a conclusive summary.

## 2. Motivating example: international moving services

Building a virtual enterprise (VE) for international moving services (IMS) presents some possibilities of building enterprise services from existing Web services, widgets, feeds, and other APIs. An international moving service aims to facilitate international relocations in various ways. These services go beyond moving items, but can include things such as visa applications and assistance in finding a new residence. In brief, the goal of international moving services starts with helping customers to find moving companies and request quotes. A very brief, incomplete and abstract

description of the various services offered by an international moving service would include:

- Find Moving Companies: compare the services of international movers by requesting free quotes for the customer, provide moving tips, and information documents needed for international moving such as official government Customs, visa and immigration, health, weather, etc.
- Travel Arrangements: find cheapest tickets and/or car renting in both places of departure and destination if needed.
- Temporary Stay Arrangement: find hotels or holiday/service apartments in both departure and destination if needed.
- Home Search: pre-select properties according to client requirements such as proximity to a childcare center and provide neighborhood guide which contains information on doctors, shopping, schools, leisure activities etc.
- School/childcare Search: provide explanation of the local education system, options including public, private and international schools, provide information on pre-school options including nurseries, toddler groups and other childcare facilities, provide list of possible schools, childcare or other facilities relating to the home search are.
- Settling-in Services: advise on banking systems, provide information on insurance of health, home, car etc., and advise on importing a car into the destination if applicable.
- Leaving Assistance: arrange property hand-back or sale, close utility accounts and arrange final bills, and manage property if client leaves before end of tenancy.

Being a VE-IMS, the payment, CRM, and bookkeeping functions that should be included for being a normal business are sourced from third parties. However, we only concentrate here on the VE's core business processes. General business related processes are not discussed here.

Because of the various requirements from its customers, the services provided by a VE-IMS are dependent on the particular situation of its customers. Different customers require a different process. This process is supported by a special-purpose piece of software, which we call an enterprise service mashup, with particular services, processes or activities. In addition to added capability, new service mashup can modify, enhance, customize or extend an existing service mashup, or include and combine parts or components (or both) from multiple existing service mashups.

A very preliminary service of the VE-IMS is to help customers to find a moving company for shipping their household effects to the new place of residence. First, the VE-IMS will request free quotes from moving companies according to the customer's place of departure and desti-

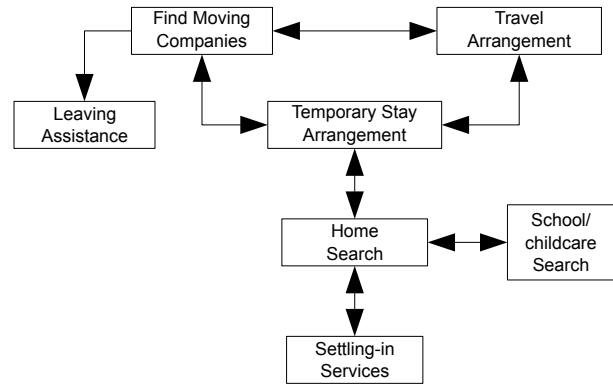


Figure 1. Meta Model for the VE-IMS Example

nation, if needed, arrange for visits, and provide a list of competent movers with their quotes.

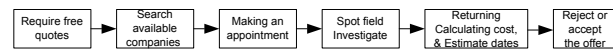


Figure 2. Model for the Scenario 1

Another, extended, service the VE-IMS may provide is finding an international mover and arranging temporary places of residence at both the locations of departure and destination, based on the dates of moving, travel, and the arrival of the household effects. The temporary place of residence at the destination should be close to a certain address such as the customer's working place. Customer can also ask for travel arrangements to be made. The time to fly and time of staying the temporary place should be worked out to minimize the total costs. Further, the customer may want the VE-IMS to find an available childcare place for the children of the customer as soon as possible, then find a rental home close by around the time that the household effects arrive.

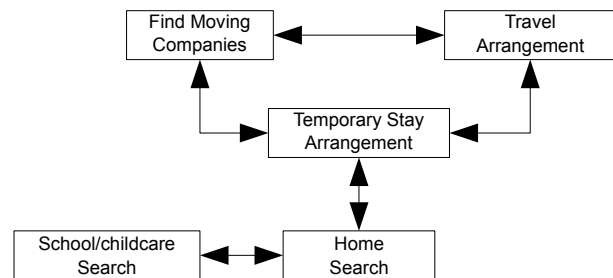


Figure 3. Model for the Scenario 2

### 2.1. Available resources supporting the example

We have found following available Web services, feeds, widgets, and mashups from web sites like syndic8.com and programmableweb.com.

A list of available feeds, Web services, widgets, gadgets, mashups that can be used as components in implementing the example:

- 1) Childcare Position Offered in France.
- 2) Childcare Position Wanted in France
- 3) Dental Plan Comparison: Compare Dental Plans Side-By-Side, Dental News, And Dental Coverage Information
- 4) Cheap Flights Special Deals: Europe low cost, the worldwide cheap flights search engine. Price comparison on low cost flights in real time.
- 5) Hotels & Accomodations Special Deals: Find hotels anywhere in the world using our worldwide hotel search engine. Accomodations price comparison in real time.
- 6) Home Value Calculator: Uses Zillow data to calculate the value of single-family homes in the U.S. Small widget suitable for placing on your Google home page.
- 7) My Camps Facebook APP: Facebook application for searching, rating, reviewing, and sharing summer camps. Put your favorite camps on the map and meet other friends who share the same experiences.
- 8) Real Estate Daily Widget: Real estate facts, terms, people, websites, history, and events in a Google Gadget. Each day of the year contains a reference about the real estate and housing industries.
- 9) Easy One Loan and Home Values: Mashup of Zillow and Yahoo Maps as supplement to online mortgage service.
- 10) Child Care Finder. Find babysitters, nannies, and other care options visually with Google Maps.
- 11) Peekacity API Users Google Streetview for neighborhood amenities Peekacity is used primarily by real estate agents as a service to their customers (currently in Chicago and Dallas/Fort Worth). The addition of Street View enables customers to view street level photos of properties, and virtually drive up and down the streets in the neighborhood.
- 12) Moving Companies Feeds from 123movers.
- 13) Moving Tips Feeds from 123movers.

Table 1. Services v.s. Available Web Services

Find Moving Service:	12, 13
Travel Arrangement:	4
Temporary Stay Arrangement:	5,7,8,11
Home Search:	6,8,9,11
School/childcare Search:	1,2,10,11
Settling in Service:	3,11
Leaving Assistance:	6

We provide information of which services can be implemented by which available Web services, feeds, widgets, and mashups in Table 1. It shows that the core business of VE-IMS can be fulfilled by existing resources manually in the certain areas, i.e. up to a user's requirements. The

list of available Web services, feeds, widgets, and mashups do not include universal school or child care information, home searching information even information of universal cheap flight tickets. Therefore, we do need a mechanism which support to look for related to users required resources such as Web services, feeds, and widgets and to compose them automatically. To be able to finding related resources, we need semantic technologies. We can thus annotate Web services, feeds, widgets and so on in advance and compose them automatically. In section 4, we will discuss it further.

In this example case, it will be expensive and difficult to build a traditional workflow system to support the business process. It means that we have to know all information in advance or providing an interface to add information in the traditional workflow solution. The dependences are various, such as finding a home close to the best school/ the available child care, find the good schools close to the home address. It would however be handy for using a business process mashup solution, specially, if automatically invocation of needed feeds, Web services, etc. and execution processes are supported. The different processes of VE-IMS can be implemented by different process instances. Users (i.e. owner of VE-IMS) may only need to edit the certain processes to be able to meet all requirements from new customers.

### 3. Characteristics of process-oriented enterprise mashups

Process-oriented mashups in the enterprise can help to solve both business and IT challenges, especially for small, medium and virtual enterprises that have less resources to create traditional BPM solutions. Businesses seek greater agility, greater configurability, cross-platform and need to respond faster to an increasing pace of business. A process-oriented enterprise mashup offers the next step in technology to aid business people to find best deals over the Web. Process-oriented enterprise mashups should also be end-user friendly. The creation of process-oriented applications should be based on reusability and adaptability. It should not require high IT skills from the end users to create or customize mashups.

Process-oriented enterprise mashups help to solve these challenges through self-service application development, enabling to move to the next level of innovation, speed, and agility by allowing users to combine and remix different sets of data in new ways. In this way, process-oriented enterprise mashups can provide insight into corporate data that was simply not possible before. Process-oriented enterprise mashups have thus following characteristics:

- Support enterprise users
- Fit within enterprise information system context (be supportable, scalable etc.)

- Support for “upgrading” the mashup to usage by colleagues with similar issues. Including versioning, transfer of ownership etc.
- Simple, flexible applications that solve daily problems
- Can be created in minutes, hours, or days (given heavy reliance on reuse and lightweight integration techniques).
- Can be “situational” in nature, easily customized to meet the unique needs of an individual or situation.
- Mashing of functionality from different sources supports new insights.
- Often supports self-service application development.
- Can help make Service-Oriented-Architecture (SOA) more business-relevant and visible, increasing reuse of services and widgets.

#### 4. Key issues of process-oriented enterprise mashup

For business use it is not sufficient to suffice with primarily data integration with external and enterprise systems. It is much more important to determine how to compose Web services, widgets, feeds and mashups into a service. However, Web services, widgets and feeds are too “small” from a business standpoint, their granularity is too fine. Only when they are elevated to the level of business processes, will they be of high value for business users.

Being able allow business people to self-serve using process-oriented enterprise mashups, many issues need to be solved first.

A lightweight business process modeling tool allows the users of process-oriented enterprise mashups to specify their requirements easier. Lightweight is a relative concept. What do we compare with? Lightweight in which dimension? One of the criteria of being lightweight is end-user friendliness. Who are actually our end users? What kinds of business or IT/technical skills we can expect from them. User friendliness is not easy. It is very hard to provide a definite answer. we do believe that the end-users may not be able to create a complex process model, but they should read a reasonably simple process model. Therefore, the end users are not typically paid to do the job of a professional modeler or programmer, and any programming or modeling efforts that they perform tend to be basic and only applied to the extent that they solve the business problem at hand.

A pattern is an abstraction from a concrete form which keeps recurring in specific non-arbitrary context [12]. The use of patterns is a proven practice in the context of programming, as evidenced by the impact made by the design patterns of Gamma et al. [13]. Process-oriented enterprise mashups need to provide process modeling patterns, modeling fragments /process modeling templates and even completed process models of typical cases to our end users. The end users can reuse, edit or add process models

using patterns and/or templates. In using process patterns, templates, it is very important to compare and specially to point out the differences of similar models (process patterns and templates). Because the end users are not experts in process modeling, the process model needs to be verified before invoking Web services. Process verification should be supported. In this way, the end users can run their processes easier while reasonable execution results can be expected.

Besides, current standards for describing Web services use syntactic (XML-based) notations such as WSDL. As these descriptions do not provide machine understandable service semantics, automation of Web service discovery, composition and invocation are thus not possible. Using semantic technologies such as SA-WSDL [14], OWL-S [15], WSMO [16], and WSMO-Lite [17] to describe Web services instead, Web services discovery, contracting, mediation, composition, and invocation can be performed automatically. Computers use machine-processable descriptions of Web services to reason and invoke according user specified requirements. Within Web 2.0, there are many services like mashups, gadgets, pipes which do not use standard Web services technology to describe their interface, communication or enactment, but work by merit of interpreting string. These entities should also be annotated.

Semantics facilitate the management of categories of process templates, Web services, Web 2.0 services and other resource as whole. It aids users to discover, select and finally automate services.

##### 4.1. Difficulties in adoption

From a technical standpoint enterprise mashups are still an unmaturing technology. Besides the challenge of semantic heterogeneity, the technological basis for enterprise mashup is mature though.

A big technical challenge to mashups is software evolution. Mashups are built flexibly and fast. Where mashups are further predominantly built up of external components, mashups are highly sensitive to change of these components. External components can come from the mashup platform, from within the enterprise, and from outside providers on contract or without contract. Especially components from outside providers can easily change, and require change to the mashups.

When exposing enterprise information systems, it is important that proper security and authorization is in place. Mashups act in the name of their owners, but these owners should not need to log in to each individual web service. This means that single sign on technologies need to be supported by the mashup platform.

Mashups represent an interesting challenge in the area of IT Governance. End-user programming means that the control over the application functionality moves to the end users, away from specialised departments. This may create

political problems as well as genuine management problems. For example, when an enterprise mashup becomes business critical, who then is responsible for its functioning? The “IT department” has no influence or knowledge of the design. The “Developer” is not schooled in program design and can not be expected to be held responsible for flaws. This problem however, is not altogether different from mission critical spreadsheets. It is a problem that must be understood and managed, not an unmanageable problem. A solution could be for example to transfer mashups that become mission critical to specialised departments, or to have a more permanent solution written.

Another management issue in the mashup area is the issue of duplicated work. When two people work on similar mashups part of the effort is wasted in duplication. On the other hand, coordination costs are avoided. Given the current difficulties to have custom solutions created, it seems that the benefits gained outweigh the costs of duplication. Especially when good sharing mechanisms are used that enable easy retrieval of existing mashups.

These possible issues show the limits of mashups. Mashups can not take the place of more elaborate information systems. The role of mashups is more in addition to existing information systems, providing automation to tasks that now have to be performed manually, as traditional information system development for the tasks is not economically viable.

## 5. Conclusions

Process composition in a business process mashup has turned out to be the potential useful technology for SOA-based automation and integration of business operations. It proved an agile approach to adapt fast pace business environment. This paper has presented a new concept of a business process mashup through a description of a motivating example and a view of potential resources for implementing the example using the business process mashup. We have analyzed the characteristics of process-oriented enterprise mashups and discussed the key issues of process-oriented enterprise mashup development.

To fully exploit the potential of Web-centric compositions, we are starting new work in several areas. We are exploring a lightweight business process modeler. We are providing most popular process templates which are based workflow patterns [18]. We are aiming that users can pick up and to run. Further we are also interested in annotating the resources using WSMO-lite and Micro-WSMO.

## References

[1] E. Thomas, *Principles of Service Design*. The Prentice Hall Service-Oriented Computing Series, 2007.

[2] F. Curbera, W. Nagy, and S. Weerawana, “Web service: Why and how?” in *The OOPSLA 2001, Workshop on Object-Oriented Services*, 2001.

[3] T. O’Reilly, “What is web 2.0: Design patterns and business models for the next generation of software,” *Communications & Strategies*, no. 1, p. 17, 2007.

[4] A. V. Riabov, E. Boillet, M. D. Feblowitz, Z. Liu, and A. Ranganathan, “Wishful search: interactive composition of data mashups,” in *WWW ’08: Proceeding of the 17th international conference on World Wide Web*. New York, NY, USA: ACM, 2008, pp. 775–784.

[5] “Google mashup editor,” <http://editor.googlemashups.com>.

[6] “Yahoo pipes,” <http://pipes.yahoo.com>.

[7] D. Benslimane, S. Dustdar, and A. Sheth, “Services mashups: The new generation of web applications,” *IEEE Internet Computing*, vol. 12, no. 5, pp. 13–15, 2008.

[8] E. M. Maximilien, A. Ranabahu, and K. Gomadam, “An online platform for web apis and service mashups,” *IEEE Internet Computing*, vol. 12, no. 5, pp. 32–43, 2008.

[9] E. M. Maximilien, H. Wilkinson, N. Desai, and S. Tai, “A domain-specific language for web apis and services mashups,” in *ICSOC ’07: Proceedings of the 5th international conference on Service-Oriented Computing*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 13–26.

[10] F. Curbera, M. Duftler, R. Khalaf, and D. Lovell, “Bite: Workflow Composition for the Web,” in *ICSOC ’07: Proceedings of the 5th international conference on Service-Oriented Computing*, vol. 4749. Berlin, Heidelberg: Springer-Verlag, 2007, p. 94.

[11] F. Rosenberg, F. Curbera, M. J. Duftler, and R. Khalaf, “Composing restful services and collaborative workflows: A lightweight approach,” *IEEE Internet Computing*, vol. 12, no. 5, pp. 24–31, 2008.

[12] D. Riehle and H. Zullighoven, “Understanding and using patterns in software development,” *Theory and Practice of Object Systems*, vol. 2, no. 1, pp. 3–13, 1996.

[13] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns Elements of Reusable Object-Oriented Software*. Addison-Wesley Publishing Company, Reading, MA, USA, 1995.

[14] W3C, “Semantic annotations for wsdL and xml schema,” <http://www.w3.org/TR/sawSDL/>, Aug. 2007.

[15] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, and K. Sycara, “Owl-s: Semantic markup for web services,” <http://www.daml.org/services/owl-s/1.1/overview/>, 2004.

[16] D. Fensel, H. Lausen, A. Polleres, J. de Bruijn, M. Stollberg, D. Roman, and J. Domingue, *Enabling Semantic Web Services: The Web Service Modeling Ontology*. Springer, 2006.

- [17] T. Vitvar, J. Kopecký, J. Viskova, and D. Fensel, "Wsmo-lite annotations for web services," in *5th European Semantic Web Conference 2008*, 2008.
- [18] W. van der Aalst, A. ter Hofstede, B. Kiepuszewski, and A. Barros, "Workflow Patterns," *Distributed and Parallel Databases*, vol. 14, no. 1, pp. 5–51, 2003.