MARKET TRENDS OF USER-CENTRIC ENTERPRISE MASHUPS

Alexander Fesak  Gieljan Everaert  Mark Musters  Jasper Straver  Tim Stuart
Tilburg University, TiSEM, P.O. Box 90153, 5000 LE, Tilburg, The Netherlands

ABSTRACT
Numerous Web services and APIs available online have triggered the creation of mashups. They manage to provide added value by timely and flexible reuse and combination of existing data or software components to satisfy ad hoc needs. Most mashups strive for end-user friendliness and certain types of them are increasingly adopted by enterprises. In this paper, based on a review of recent literature and some enterprise mashup tools, we identify the current trends in the domain of user-centric enterprise mashups. Then we discuss the implications and challenges related to the enterprise mashup adoption.

Keywords
Enterprise Mashups, End-User Development, Web APIs, User-centric SOA, Process Mashups.

INTRODUCTION
Over the past 20 years the introduction of transaction systems has supported considerable business process automation efforts. It is argued that the next wave in corporate technology adoption will be related to productivity improvements due to broad collaboration and a high degree of participation (Hoyer, 2011). Thus, business users will “compose their individual applications based on user friendly building blocks” as opposed to consuming finished and largely unchangeable applications. This new demand for creating ad hoc applications in order to react to unexpected needs has urged a new development paradigm, known as enterprise mashups (Hoyer, 2011).

Being more “versatile and complex”, recent mashup tools tend to integrate at application and UI level, in contrast to early mashups that employed integration and composition mainly at data level (Roy, 2010). Enterprise mashups “leverage and combine concepts from service-oriented architecture and peer production in order to empower business users” (Hoyer, 2011).

Enterprise mashups are also claimed to have “enormous potential to allow more rapid and much less expensive development of applications by emphasizing assembly over development, economies of scale by enabling high levels of reuse, and the consequent ability to rapidly get software solutions with the right data in the right place at the right time” (Hoyer & Fischer, 2008).

BACKGROUND
According to Yu et al. (2008) Web mashups are “Web applications generated by combining content, presentation, or application functionality from disparate Web sources”. Mashups strive to combine those sources to create useful new applications or services. Hoyer & Fischer (2008) summarize that an Enterprise Mashup is “a Web-based resource that combines existing resources, be it content, data or application functionality, from more than one resource in enterprise environments by empowering the actual end-users to create and adapt individual information centric and situational applications.” Unlike consumer mashups, enterprise environments implicate additional requirements like security, quality, availability, and integration with existing back-end systems (Hoyer & Fischer, 2008). Still, in most business cases mashups are currently viewed as a flexible tool for developing non-mission-critical Web applications. Such applications (e.g., enterprise dashboards and other features enabled by services like IBM QEDWiki, ARIS MashZone, Yahoo! Pipes) are usually created to improve decision making or satisfy business collaboration needs, but often are used only for short periods of time while the need exists (Bianchini, De Antonellis, & Melchiori, 2012).
Mashup applications usually reuse and combine Feeds, JavaScript APIs, and Web services allowed by Web APIs. And such combination may be carried out across different integration layers (i.e., data, logic and UI level) (Yu et al., 2008). Web API is “a development in Web services” that uses communications based on simpler representational state transfer (REST) (Benslimane, Dustdar, & Sheth, 2008). REST Web services (often called ‘RESTful Web services’ or ‘RESTful web APIs’) generally represent code based API integration and do not require message based integration using XML Web service protocols (SOAP and WSDL utilized by SOA) to support light-weight services. (Benslimane et al., 2008; Yu et al., 2008).

Most existing enterprise mashups usually choose REST Web services (Pautasso, Zimmermann, & Leymann, 2008) over XML Web services to integrate enterprise functionality. However, existing large enterprise systems (e.g. SAP’s eSOA platform) mostly use complex XML Web services to develop and integrate applications (Roy, 2010). At the same time, value promising implications of the end user in the life cycle of software development represent an increasingly important issue (Benhaddi, Baina, & Abdelwahed, 2012). For instance, SOA often “suffers from not being user-centric due to the neglect of the creative potential of the end user, not involved in the life cycle of the SOA software” (Benhaddi et al., 2012).

General trend for the next few years will be to make systems easy to develop, and to create new environments that allow non-technical users to develop applications (Lieberman, Paternò, Klann, & Wulf, 2006). Lieberman et al. (2006) define End-User Development (EUD) “as a set of methods, techniques, and tools that allow users of software systems, who are acting as non-professional software developers, at some point to create, modify or extend a software artefact”. The scope of this research is limited to Enterprise Mashups aiming for distinct EUD capabilities.

As Roy (2010) points out, developers usually deal with the technical integration of Web services and the interpretation of their meaning. They try to predict appropriate parameters and operations of a potential future use case. Still, the actual knowledge and meaning is kept with the business users who inherently have extensive practical comprehension of the business processes and respective applications. “The challenge is to support and facilitate the developer’s effort to dynamically perform most of the integration process while leaving decisions about the meaning to end-users” who “may be assisted in their daily tasks by empowering them to intuitively create ad-hoc applications, which solve an immediate and specific business problem in less time” (Roy, 2010).

RECENT DEVELOPMENTS AND TRENDS

Ubiquitous Cloud Computing and Social Networking

Unsurprisingly, enterprise mashups increasingly use the capabilities of cloud computing and tend to integrate the growing number of applications distributed via software as a service (SaaS) delivery model (Hoyer, 2011; Siebeck et al., 2009). Services like IFTTT and Zapier focus on combining such numerous cloud-based services. Social networking platforms also play an expanding role as both a source of data or functionality to be mashed up and a collaborative tool to devise and refine another enterprise mashup (Bianchini, De Antonellis, et al., 2012; Lachenmaier, Ott, & Koch, 2012). Services like Yammer, MangoApps or Chatter may represent some first steps in the latter direction. Regarding social network analysis and data mining, one of the recent design research works presents a person-centric multi-user mashup middleware solution for aggregation and filtering of data from social services (Lachenmaier et al., 2012). It is also claimed to offer “a universal entry point in combination with unified data access for different client devices and can be used in various application scenarios with regard to individually specified service levels”.

Convergence of Enterprise Mashups and SOA

Although enterprise mashup tools based on RESTful Web services, RSS or other lightweight resources may help to fill in the gap between the end-user and the software development, these tools have a number of disadvantages (Benhaddi et al., 2012). They do not take into account enterprise-class and complex services supported by SOA (Roy, 2010), as well as do not allow abstraction of resources and use of complex parameters. Proper integration at the business process level is also not supported. Finally, unlike e.g. BPEL, they are rather fragile and unstable (Benhaddi et al., 2012).

Therefore, many enterprise mashups providers tend to add respective SOA features to their offerings (e.g. IBM WebSphere sMash). Considerable benefits are expected from making the SOA user-centric and combining it with the Mashup stack which allows visual mixing of web resources (Benhaddi et al., 2012). For instance, Benhaddi et al. (2012) propose an approach for making SOA more user-centric by utilizing the Mashup stack (Hoyer, Stanoevska-Slabeva, Janner, & Schroth, 2008).
Establishment of Process Mashups

Some enterprise mashups have started striving for integration at not only the data and presentation but also on the business process layer. They may model and apply the control flow in order to achieve the coordination or automation of tasks (Xie, De Vrieze, & Xu, 2009). Further, when such a mashup is able to support concurrent work of multiple users, incorporate multiple pages, and provide workflow support (including control and data flow with regard to human tasks), it may be defined as a process mashup (Daniel, Koschmider, Nestler, Roy, & Namoun, 2010). The development of process mashups is considered as one of the important trends within enterprise mashups and implies multi-user, multi-page and workflow-supported mashups (Daniel et al., 2010). The combination of all these characteristics “allows a controlled interwoven composition of the three basic ingredients data, functionality and UI that is currently not at all supported” (Daniel et al., 2010).

Although there seem to be few tools on the market that support the full definition of a process mashup (e.g. MarcoFlow editor satisfies it partially), recent research provides conceptual support for collaborative tasks in such mashups. For example, a respective extension of Business Process Model and Notation (BPMN) has been proposed to enhance process mashups (Torres, Pérez, Koschmider, & Daniel, 2011). It has been defined within the context of OOWS4BP, a model driven engineering approach to deal with the development of business process-driven web applications. Another tendency towards workflow-supported mashups is the increasing degree of Web API automation and near real-time reciprocity supported by services like Zapier, IFTTT or Wappwolf. For example, the authors of this paper utilized Zapier to integrate Google Drive/Docs with Basecamp project management service for the group work on the Seminar Information Management project.

API Aggregation

Web API aggregation is a general trend underlying the development infrastructure of enterprise and consumer mashups as well as other applications (Trischler, Kleinfeld, & Steglich, 2011). As the growing number of different scattered APIs increase development costs, some services try to aggregate those multiple APIs into a single interface or stack for developers to use. API aggregation services like Singly or Adigami standardize both interfaces and objects across potentially very disparate platforms. A recent academic paper has also proposed Web API selection patterns that could assist a mashup designer in selecting and aggregating Web APIs (Bianchini, Antonellis, & Melchiori, 2012).

Leveraging Telco Services

While the opportunities for voice search and other audio based online interaction are expanding (e.g. Apple’s Siri), Web services allowed by Web APIs are viewed as a channel bridging a mobile and web user’s voice or video with information he or she needs in a particular situation (Chudnovskyy, Gebhardt, Weinhold, & Gaedke, 2011; Gebhardt et al., 2012). The term ‘telco services’ represents numerous emerging telephony and messaging services providing new capabilities regarding voice, video and data transfer (Gebhardt et al., 2012). In terms of usage, the developer is abstracted from the complexity and protocol mismatch of operator networks as telco services are invoked with the help of provider-specific Web APIs (Chudnovskyy, Nestler, et al., 2012).

However, the integration of telco services into Web applications often remains challenging despite several existing initiatives aiming at standardization of cross-provider service interfaces (e.g. OneAPI GSMA initiative) (Chudnovskyy, Nestler, et al., 2012). Chudnovskyy, Weinhold et al. (2012) have presented a reference architecture for integration of telco services into enterprise mashups that supports execution and orchestration of business processes.

Other Recent Developments

Reflecting emerging trends in enterprise mashup composition, one of the recent research works is claimed to have introduced a Web API creation methodology in order to create Web APIs for enterprise web applications (even not initially intended to be reusable) much easier than by extending the source code (Tanaka, Kume, & Matsuo, 2011). Liu et al. (2012) also argue that mashups, mostly developed with a predefined set of services and components, still need sizeable programming work to be extended and struggle to quickly find an appropriate alternative and use it to substitute a suddenly unavailable service. Hence, they introduce a new approach “to enable mashups to select and invoke semantic Web services on the fly” (Liu et al., 2012). Another example of a recent development in the enterprise mashup domain is the approach to so-call distributed UI orchestration proposed by Daniel et al. (2012). This approach constitutes a component-based development technique for the workflow management and service composition, and a model for UI components that requires extending the expressive power of a standard service composition language, such as BPEL. The benefit of the proposed approach may be backed by more expressive models at design time supporting resource abstraction, improved separation of concerns, and robust code generator that do not require a new runtime system or language (Daniel et al., 2012).
DISCUSSION

The reviewed trends and recent developments within user-centric enterprise mashups may also reveal a number of challenges. As mentioned by Benhaddi et al. (2012), mashup adoption by enterprises requires handling related security and availability risks. The user support, quality of service, governance, and repository management issues should also be addressed properly. Clearly defined enterprise governance regarding the use of new mashup technologies and strategy aware adoption of a ‘user-centric mashuped SOA’ (Benhaddi et al., 2012) might be helpful here.

As the data integration is the central element of enterprise mashups, a noticeable challenge may be the separation between the data integration and the application logic. Grass roots processes of mashup composition may lead to a disordered mixture of data and application logic reminding a kind of problems traditional enterprise applications faced a few decades ago (Jhingran, 2006). The architectures for building layered Mashup applications should be ‘rigorously defined’ (Benhaddi et al., 2012) to keep the clear combination of data and application integration logic.

Another challenge for process mashups is additional requirements and constraints imposed by the intrinsically complex nature of the business tasks to be supported. For instance, an engine that is able to handle long running process logic for multiple concurrent process instances and users as well as adequate identification/authentication tools may be needed. Thus, process mashups may be positioned at the boundary between mashups and traditional application development (Daniel et al., 2010). On the one hand, more capable mashup platforms and execution environments can offer more expressive and stable mashup applications supporting different user roles or groups. On the other hand, “the required set of skills and knowledge to use them effectively is counter-productive to the mashup idea” (Daniel et al., 2010). Therefore, we agree with Daniel et al. (2010) who argue that process mashups will not replace core business process management systems, but still will be adopted by many enterprises satisfied with their expanding capabilities.

Successful implementation of enterprise mashups may also demand novel approaches to requirements engineering (RE). As primarily small and situational applications, mashups utilize visual composition tools and loosely coupled widgets to facilitate the rapid implementation of changing requirements. At the same time, methods and techniques of traditional RE are not flexible enough because of isolated development phases, inadequate tool support, and different models for segregated user groups (Tietz, Rümpel, Liebing, & Meißner, 2012). Business experts and mashup creators often face “time-consuming and error prone component discovery and composition” as they are insufficiently equipped for expressing and formalizing their requirements (Tietz et al., 2012). Overall, this issue remains open for future research.

Using social networking elements as a collaboration enabler for user-centric enterprise mashups may become a widely expanding feature. In our opinion, it may be especially relevant for process mashups where interaction logic and mutual agreements define the requirements and eventual success of the mashup. Another proposition could be that SMEs with their relatively modest needs for complexity, information security and technical performance may benefit from user-centric enterprise mashups the most.

CONCLUSION

We have presented a review of current trends in the domain of user-centric enterprise mashups. The key trends identified comprise the growing use of cloud computing and social networking mashable services, convergence of enterprise mashups and SOA, establishment of process mashups, Web API aggregation, utilization of telco services and others.

Possible management implications, challenges, and opportunities related to enterprise mashup adoption discussed include security and availability risks, user support, governance issues, combination of data and application integration logic, enterprise-class functionality and performance requirements disrupting the mashup idea, lack of adequate requirements engineering techniques, and social media enabled mashup collaboration.

REFERENCES


Short Papers for Seminar IM 2012-2013
B. Van de Walle
Interoperability in conjunction with itAIS, (2012).