ECIS 2009 Proceedings

European Conference on Information Systems (ECIS)

2009

# IT impacts on operation-level agility in service industries

Volker Hoyer
*University of St. Gallen*, volker.hoyer@unisg.ch

Katarina Stanoevska-Slabeva
*University of St. Gallen*, katarina.stanoevska@unisg.ch

Recommended Citation

Hoyer, Volker and Stanoevska-Slabeva, Katarina, "IT impacts on operation-level agility in service industries" (2009). *ECIS 2009 Proceedings*. 111.
http://aisel.aisnet.org/ecis2009/111

# TOWARDS A REFERENCE MODEL FOR GRASSROOTS ENTERPRISE MASHUP ENVIRONMENTS

Hoyer, Volker, University of St. Gallen (=**mcm***institute*) and SAP Research St. Gallen, Blumenbergplatz 9, 9000 St. Gallen, Switzerland, volker.hoyer@unisg.ch

Stanoevska-Slabeva, Katarina, University of St. Gallen (=**mcm***institute*), Blumenbergplatz 9, 9000 St. Gallen, Switzerland, katarina.stanoevska@unisg.ch

## Abstract

*A new kind of Web-based application, known as Enterprise Mashups, has gained momentum in the last years: Business users with no or limited programming skills are empowered to leverage in a collaborative manner user friendly building blocks and to combine and reuse existing Web-based resources within minutes to new value added applications in order to solve an individual and ad-hoc business problem. Current discussions of the Mashup paradigm in the scientific community are limited on technical aspects. The collaboration and the peer production management aspects of the Mashup development have received less attention yet. In this paper, we propose a reference model for Enterprise Mashups which provides a foundation to develop and to analyse grassroots Enterprise Mashup environments from a managerial and collaborative perspective. By following the design science research approach, we investigate existing reference models and leverage the St. Gallen Media Reference Model (MRM). The development of Enterprise Mashups is structured by market transaction phases similar to electronic markets. The user roles, the necessary processes and the resulting services are modelled according to the views of the MRM. By means of the SAP Research RoofTop Marketplace prototype we demonstrate the application of the designed reference model for grassroots Enterprise Mashups environments.*

*Keywords: Enterprise Mashups, St. Gallen Media Reference Model, Design Science, Electronic Markets.*

# 1 INTRODUCTION

## 1.1 Motivation and Problem Scope

The process of development of Web-based business applications follows usually the typical process of software development involving first assessment of user requirements followed by a long process of development and testing. The functionality of the resulting application is actually a compromise of user requirements, as not all user requirements can be considered. As a result, there is a long tail – a term first coined and popularized by Chris Anderson (2004) – of many specific and heterogeneous user requirements or dynamically changing user requirements that are not covered by the IT department (Carrier et al. 2008, Hoyer and Stanoevska-Slabeva 2008).

A possible solution for this problem could be a new development paradigm, known as Enterprise Mashups, which integrates the users from the business units characterized by no programming skills in the software development process (Cherbakov et al. 2007). At the core of the paradigm are two aspects: First, empowerment of the end user to cover ad hoc and long tail needs by reuse and combination of existing software artefacts; and second, broad involvement of users based on the peer production concept. According to Yochai Benkler, who coined the term peer production, *"it refers to production systems that depend on individual action that is self-selected and decentralized rather than hierarchically assigned"* (Benkler 2006). Thereby, the creative energy of large number of people is used to react flexible on continuous and dynamic changes of the business environment. Instead of long-winded software development processes, existing and new applications are enhanced with interfaces and provided as user friendly building blocks.

Existing research efforts focus mostly on technical aspects like the development of Mashup tools – i.e., IBM Mashup Center (formerly IBM QED Wiki, IBM Mashup Hub, IBM Damia), Intel Mash Maker, Microsoft Popfy, and Kapow Mashup Server, which enable easy integration of available components - or underlying technical concepts and principles - i.e., Maximilien et al. (2008), Yu et al. (2008), Hoyer et al. (2008). The discussion from a collaborative and peer production perspective is still missing in the scientific community discussing the implications, challenges, but also the potential benefits and limitations of the Mashup paradigm in the enterprise context. Important questions from these perspectives are: Who is involved in the Mashup development and what are the roles of the different players in particular of the software development department and the business units? What are the necessary processes to enable and support community building and collaboration?

The goal of this research paper is to fill this gap by designing a reference model which integrates the collaborative and community aspects and can serve as a framework to structure the development and analysis of Enterprise Mashup environments. The general research questions guiding this study are to model the roles and relationship of the interacting users as well as to interlink community and technical aspects within grassroots Enterprise Mashup environments.

## 1.2 Research Design: Design Science applied

For answering the research questions motivated in the previous section and characterized by a practical nature, engaged research is needed in order to provide rigorous solutions. Design science research aims at solving practical and theoretical problems by creating and evaluating IT artefacts indented to solve identified organizational problems (March and Smith, 1995, Hevner et al. 2004, Peffers et al. 2008). Artefacts represent the final result of a design process. They can be characterised as constructs, model, methods, or instantiations (March and Smith, 1995).

To come to rigorous and relevant research results, we draw upon on Peffers et al. (2008) to specify the subsequent phases of the design science research process applied:

- **Problem Identification and Motivation**. In the motivation, we specify the specific research problem, show the practical relevance and justify the value of a solution. Based on the problem scope, we derive the research questions guiding this paper.
- **Define the Objectives for a Solution.** In the second chapter, we infer the objectives of a solution from the problem definition and knowledge of the state of art. In particular, chapter two defines Enterprise Mashups, analyses existing reference models by means of a literature review and presents the St. Gallen Media Reference Model.
- **Design and Development.** In chapter three, we leverage the St. Gallen Media Reference Model (MRM). As observed by Legner (2008), Hoyer and Stanoevska (2008), and Carrier (2008) the peer production of Enterprise Mashups has many similarities to electronic markets: Available components are classified and offered by providers and potential consumers search for the most suitable ones and if required pay for the usage. Thus, the required support should enable the matching of supply and demand in a way similar to conventional market phases (knowledge, intention, contract/ design, and settlement). We apply these market phases to model the roles and the relationships between the interacting users according to the layers of the MRM.
- **Demonstration.** For demonstration of the designed artefact, we apply the reference model in order to develop the SAP Research Rooftop Marketplace. In particular, we structured the requirement and software design phases according to the reference model for Enterprise Mashups.

The results of each of the above activities are presented in the remaining parts of the paper. Finally, the last chapter closes with a brief summary, limitations of the conducted research and an outlook to further research.

# 2 REALTED WORK AND OBJECTIVES OF THE SOLUTION

## 2.1 Enterprise Mashups – Definition and Characteristics

In literature, the exact definition of Enterprise Mashups is open to debate. In this work, we refer to the following definition: *"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"* (Hoyer et al. 2008). By simplifying concepts of Service-Oriented Architecture (SOA) and by enhancing them with the Web 2.0 Philosophy of peer production, Enterprise Mashups focus generally on software integration on the user interface level instead of traditional application or data integration approaches (Daniel et al. 2008). In contrast to SOA that is characterized by high technical complexity of the relevant standards and requiring specialists' technical knowledge, the simplified Enterprise Mashups enable the integration of the end users with no or limited programming skills in the development process.

The relevant architectural components of the Enterprise Mashop paradigm are resources, widgets, and Mashups (Hoyer et al. 2008) and can be structured in an Enterprise Mashup Stack comprising three layers (see figure 1): **Resources** represent actual contents, data or application functionality that are the core building blocks of Mashups. They are encapsulated via well-defined public interfaces (Application Programming Interfaces; i.e., WSDL, RSS, Atom, etc.) allowing the loosely coupling of existing resources – a major quality stemming from the SOA paradigm. These resources are provided by enterprise systems or by external Web providers (i.e., Amazon, Google, etc.) and are created by traditional developers who are familiar with the technical development concepts. The layer above contains **widgets** which provide simple user interaction mechanism abstracting from the complexity of the underlying resources. For example a widget "Customer Data" might provide results for a predefined query requesting the data for all customers of a sales manager. The creation of these widgets can be done by consultants or key users in the business units who understand the business requirements and know basic development concepts. Finally, end users with no programming skills are able to combine and configure such visual widgets according to their individual needs, which results in a **Mashup**. For example, the sales manager wires the "Customer Data" with a map to show the location of the customers.

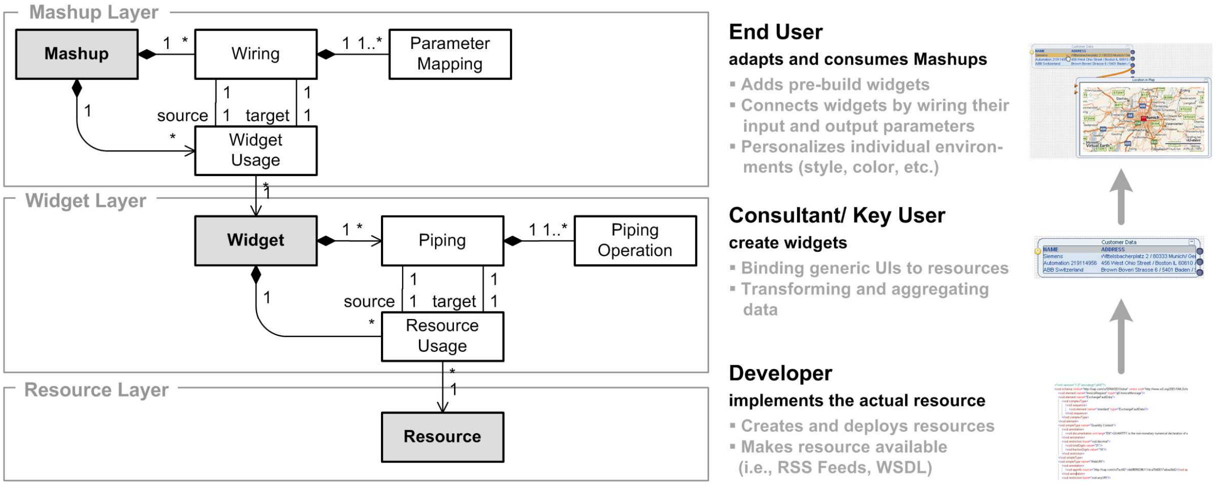

Mashup Layer
Mashup
1 *
Wiring
1 1..*
Parameter Mapping
1
1
1
source 1 target 1
*
Widget Usage
*
1
Widget Layer
Widget
1 *
Piping
1 1..*
Piping Operation
1
1
1
source 1 target 1
*
Resource Usage
*
1
Resource Layer
Resource
End User
adapts and consumes Mashups
▪ Adds pre-build widgets
▪ Connects widgets by wiring their input and output parameters
▪ Personalizes individual environments (style, color, etc.)
Consultant/ Key User
create widgets
▪ Binding generic UIs to resources
▪ Transforming and aggregating data
Developer
implements the actual resource
▪ Creates and deploys resources
▪ Makes resource available (i.e., RSS Feeds, WSDL)
Customer Data
NAME
ADDRESS
Siemens
Wittelsbacherplatz 2 / 80333 Munich/ Ge
Automation 219114956
456 West Ohio Street / Boston IL 60610
ABB Switzerland
Brown Boveri Strasse 6 / 5401 Baden /
Location in Map

*Figure 1.* *Enterprise Mashup Stack (Hoyer et al. 2008) - Meta Model and User Roles*

The first key driver of the Enterprise Mashup paradigm is the *lightweight composition style* by reusing existing building blocks in new ways. The Enterprise Mashups paradigm separates between the wiring and piping composition as depicted in Figure 1. The piping composition integrates heterogeneous resources defining composed processing data chains concatenating successive resources. Aggregation, transformation, filter, or sort operations adapt, mix, and manipulate the content of the underlying resources. The visual composition of input and output parameters of widgets on the Mashup layer is called wiring (i.e. the output parameter address of the customer widget is connected to the input parameter of the map widget). In addition to this lightweight composition style, the *mass collaboration principle* is the second key driver. The willingness of users to offer feedback to the Mashup creator, who may be unaware of problems or alternative uses, directly contributes to the adoption of the Mashup and can foster its ongoing improvement (Hoyer et al. 2008). Another important contribution of users is the inclusion of their Mashups in the available pool of components. The willingness of users to provide their Mashups for further reuse increases the number of available components.

## 2.2 Reference Modelling

Despite the popularity of the term reference modelling since the 1990s, there is still no single meaning connected to this term and it is used to designate different approaches. By analysing various definitions, Fettke and Loos (2007) identified the following three main characteristics of reference models: First, *best practices*. A reference model provides best practices for conducting business. Second, *universal applicability*. A reference model does not represent a particular enterprise or system, but a class of domains. Hence a reference model is valid for a class of domains. Third, *reusability*: They can be understood as blueprints for information systems development. Thus a reference model is a conceptual framework that could be reused in multitude of information system projects. Modeling guidelines (Becker et al. 1995) and evaluation criteria of reference models (Frank 2007) are discussed in the scientific community.

In general there are two approaches for creating reference models: either by observing many instances available in practice and extracting common elements into a reference model or by leveraging and adjustment of existing reference models. The first approach is suitable when a sufficient number of instances are available. The second approach might be a suitable option when the underlying phenomenon is not well researched yet, but similarities to other phenomena can be revealed. As

Enterprise Mashups are a recent development there are not proven good practices yet that can be applied for reference model extraction. Thus, the second approach was applied: existing reference models were checked if they fulfil the requirements of Enterprise Mashups environments.

By means of a literature review and by applying the classification framework of Braun and Esswein (2007), we analysed existing reference models that are relevant for the Enterprise Mashups paradigm. Gartner proposes a practitioner reference model that specifies the technical architecture components in Enterprise Mashup environments (Bradely 2007). A practitioner reference model of Forrester uses a similar layer structure like Gartner and the presented Enterprise Mashup Stack. In addition, a phase model is integrated specifying the inputs and information flow (Young 2008). First, the actual content provided by the IT department is provisioned for the Enterprise Mashup environment from both internal and external resources. Second, users from the business units use a so-called Mashup composer to arrange and combine content, as well as determine a visualization paradigm. Third, the mashable components are managed by a Mashup life-cycle manager and shared with others to use in new Mashups if desired. Even though both reference models provide first technical structures of Enterprise Mashup environments, a multi-view concept integrating the managerial perspective is missing. The existing reference models miss in particular support for the collaborative aspect of Enterprise Mashups development and do not provide sufficient support for the peer production process. In order to integrate the different aspects (community, processes, or technical), a multi-view reference model is necessary.

Another specific characteristic of Enterprise Mashup environments is their similarity with electronic markets. Enterprise Mashup environments need to provide besides support for easy integration of software artefacts also support for efficient management and matching of supply and demand for Mashup components. Legner (2007) describes the trading of Web Services according to market transaction phases. Carrier et al. (2008) put the discovery and sharing of mashable elements in the center of the development process to reuse existing assets in new combinations.

In summary, a comprehensive reference model for Enterprise Mashups is required that on the one hand considers technical requirements regarding the easy integration of mashable components and on the other hand support for matching of supply and demand for required Mashups based on the market paradigm. We will incorporate these findings by leveraging a scientific multi-view reference model (The St. Gallen Media Reference Model) that has its roots in electronic markets and that has already been applied succesfully for modeling electronic markets (Schmidt and Lindemann 1998), m-commerce communities (Stanoevska-Slabeva 2003), and cross-company electronic collaborations (Schroth and Schmid 2009).

### 2.3 St. Gallen Media Reference Model

The St. Gallen Media Reference Model (Schmidt and Lindemann 1998) provides a framework for specifying IT infrastructures. Under the term medium, we understand platforms based on information and communication technologies, i.e., communication spaces of *"social interaction which allow the participant to meet and which embed them in a common physical, logical, and socio-organizational structure"* (Schmidt 1997). The media reference consists of two dimensions: The horizontal dimension contains the four phases of a market transaction whereas the vertical dimension is built of four views.

The four view layers structures the different successive interaction goals of the participating agents. The *community view* describes the participating agents, their roles and the organizational structure defining the relationships among roles together with their obligations and rights. The *interaction view* refers to the relevant processes and is based upon the underlying services. The *service view* comprises all services in the four market phases that need to be available on the platform. The four services are: First, the *knowledge phase* is which information about offered services and knowledge and the media platform itself is acquired. Second, the *intention phase* in which agents signal their intentions in terms of offers and demand. Third, the *contract phase* where agents negotiate legal binding contracts and finally the *settlement phase*, in which agents act according to the negotiated contract using the platform's settlement services offered for this purpose. Examples of services in the service view are electronic product catalogs in the knowledge phase or payment services in the settlement market phase. Finally, the *infrastructure view* contains communication protocols and standards which comprise the groundwork for the implementation of services.

# 3 DESIGN: REFERENCE MODEL FOR ENTERPRISE MASHUPS

As elaborated in the previous chapter, we leverage and adjust the existing St. Gallen Media Reference Model due to its similarities to electronic markets and due to their successful application for managing communities. The driving force is the transfer of typical market transaction phases to the development of software artefacts to address the specific requirements of Enterprise Mashups. In addition to the two dimensions (views and phases), we introduce the architectural Enterprise Mashup Stack as a third dimension of the reference model for Enterprise Mashups as depicted in Figure 2.

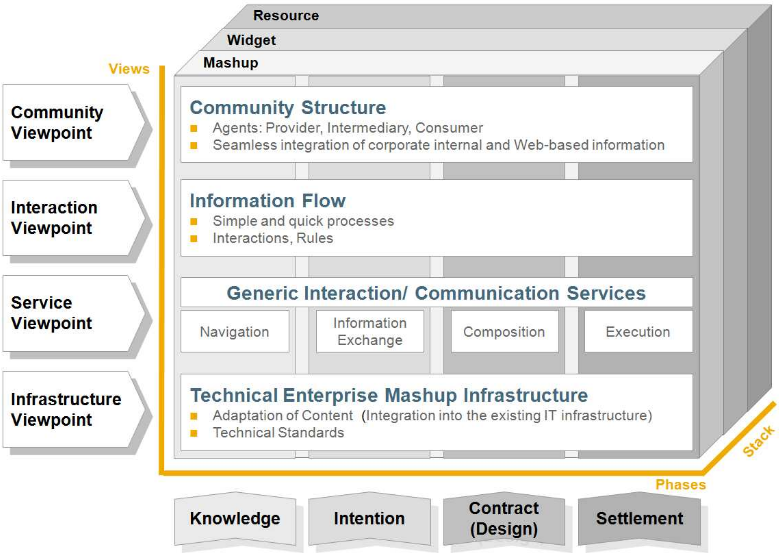

Resource
Widget
Mashup
Views
Community Viewpoint
Interaction Viewpoint
Service Viewpoint
Infrastructure Viewpoint
Community Structure
Agents: Provider, Intermediary, Consumer
Seamless integration of corporate internal and Web-based information
Information Flow
Simple and quick processes
Interactions, Rules
Generic Interaction/ Communication Services
Navigation
Information Exchange
Composition
Execution
Technical Enterprise Mashup Infrastructure
Adaptation of Content (Integration into the existing IT infrastructure)
Technical Standards
Stack
Phases
Knowledge
Intention
Contract (Design)
Settlement

*Figure 2. St. Gallen Media Reference Model for Enterprise Mashups.*

Starting with the *knowledge phase*, available mashable components (Mashup, widget, resource) are classified, rated and explained in different ways to the agents of the Enterprise Mashup environment. Concepts from the Web 2.0 philosophy, like rating, tagging, or recommending are integrated for browsing through the Enterprise Mashup medium. During the *intention phase*, the concrete offers are provided in a structured manner including the payment mode, the price as well as the delivery conditions. In the *contract (design) phase*, users select the right mashable component based on the provided information, configure it according to their preferences and combine it with other components. Finally, in the *settlement phase* the Enterprise Mashup is executed.

In the following, each view of the model is described and modelled by using the well-known conceptual modelling languages Unified Modeling Language (UML), Business Process Modelling Notation (BPMN) and Fundamental Modelling Concepts (FMC).

## 3.1 Community View

To describe the interacting and connected agents as well as their tasks and roles, we refer to the following interaction model well known in Service-Oriented Architectures (Papazoglou 2003) but also in electronic markets (Legner 2007): A *provider* develops and publishes a mashable component via an *intermediary*, where a *consumer* can find it and subsequently may compose and consume it. As depicted in Figure 4, the interaction between consumers and providers is always managed by the intermediary. The tasks of the three agent roles are described in the following:

- **Provider.** A provider implements and hosts a Mashup component (resource, widget, or Mashup) which encapsulates the actual content or knowledge. To promote their provided functionalities, the provider annotates the component with relevant information and publishes it to the intermediary through which the component description is published and made discoverable.
- **Intermediary**. An intermediary mediates between providers and consumers similar to electronic marketplaces (Legner 2007). In contrast to traditional SOA-based implementations like UDDI or ebXML, novel forms of intermediaries are currently about to emerge which improve navigation, transparency, and governance. They monitor continuously the parameters (such as availability or response latency) and provide performance metrics and other evaluation results which may be used by the consumers to select the right Mashup component.
- **Consumer.** Based on the available information of a mashable component, a consumer is able to retrieve and compose Mashup components according to his individual preferences. Consumers take also over the role of annotating Mashup components by tagging, recommending, or rating them. The consumers also contribute to the community base of widgets by providing their created and adapted Mashups back in the community pool.

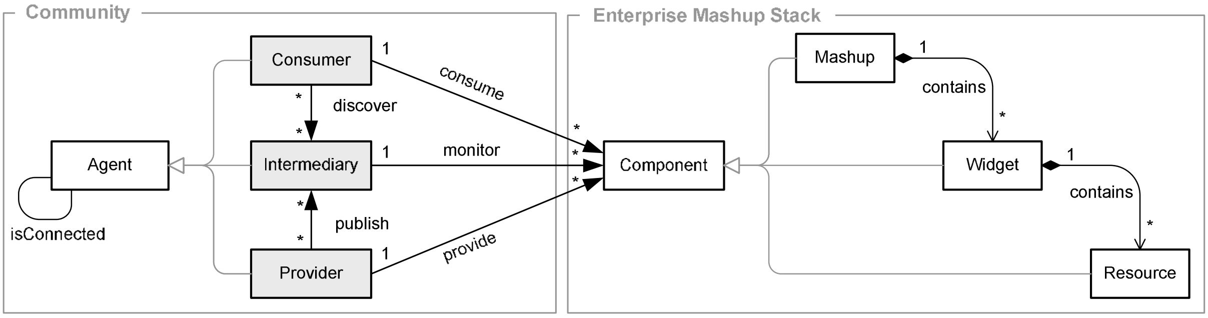


*Figure 3. Agents in the Enterprise Mashup Community.*

A critical success factor for Enterprise Mashups is a broad potential user group, familiar with the technology and willing to use it in their daily operational environment. Users from the business units are able to create their own component and provide it to the community (Hoyer and Stanoevska-Slabeva 2008, Carrier et al. 2008). In this sense, users can take over both agent roles (consumer and provider). For example, a sales manager can publish his created Mashup combing different widgets (i.e., the customer and map widget) to the community. One of his colleagues can reuse and consume the Mashup immediately without the involvement of the IT department. The creation of more complex widgets and the adaptation of existing backend services are in responsibility of the developer from the IT department (provider role). Additionally, the IT department takes over the intermediary role.

The Mashup components itself are located within the enterprise boundaries (i.e., my sales orders) or are sourced via external Web providers. The foundation of the richness of Enterprise Mashups applications is based on the seamless combination of corporate internal and external information sources. News feeds, a map or a credit cart checking Web Services can be combined ad-hoc with the internal customer data. The continuous growing numbers of available Web-based resources can be observed by the two intermediaries Programmableweb.com or Seedka.com.

### 3.2 Interaction View

Figure 4 depicts a simplified process in BPMN describing the interaction between the three agent roles as presented in the community view before. The detailled interaction process can be found in Hoyer and Stanoevska-Slabeva (2009). According to the findings of section two, the interaction process itself is characterized by permanent loops between the converging design and runtime phases. The need to adapt the operational environment ad-hoc leads to adding, removing, or replacing existing mashable components. In the following discussion of this paper, we focus on the Mashup layer.

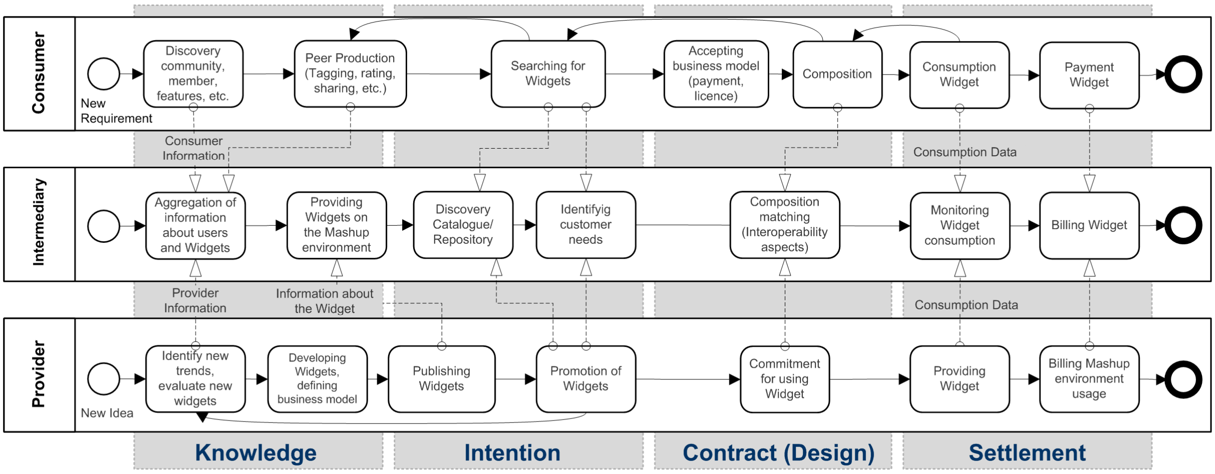

Consumer
Intermediary
Provider
New Requirement
Discovery community, member, features, etc.
Peer Production (Tagging, rating, sharing, etc.)
Searching for Widgets
Accepting business model (payment, licence)
Composition
Consumption Widget
Payment Widget
Consumer Information
Consumption Data
Aggregation of information about users and Widgets
Providing Widgets on the Mashup environment
Discovery Catalogue/ Repository
Identifyig customer needs
Composition matching (Interoperability aspects)
Monitoring Widget consumption
Billing Widget
Provider Information
Information about the Widget
Consumption Data
New Idea
Identify new trends, evaluate new widgets
Developing Widgets, defining business model
Publishing Widgets
Promotion of Widgets
Commitment for using Widget
Providing Widget
Billing Mashup environment usage
Knowledge
Intention
Contract (Design)
Settlement

*Figure 4. Interactions between the three Agent Roles (Mashup Layer).*

After registering to the Enterprise Mashup environment, consumers (i.e. the sales manager) are able to discover the community and member profiles. By means of examples in form of short videos, the benefits of the Enterprise Mashup environment are demonstrated to potential users. Only if a huge amount of users are convinced of using the environment, it will exploit its actual potential. By discovering the catalogue of mashable components (in this case widgets), consumers are able to select a widget based on extensive information provided by the provider, intermediary as well as the consumers. Reviews, recommendations, and ratings of colleagues help for selecting a component. In case the consumer accepts the underlying business model (costs, payment model, consumption licence, etc.) of a widget that is defined by the provider, he can compose the component with others by connecting the input and output parameters of the widgets. In contrast to the classical software development, the design of ad-hoc applications uses real resources and no demo systems. In this sense the consumption in the settlement phase differs only from the hidden configuration capability in contrast to the design phase. In case a new business situation comes up, the consumer shifts quickly to the design or intention perspective to adapt the individual operational environment.

## 3.3 Service View

Based on the described interaction process, we derived the required services for the process steps. According to the role of the intermediary mediating between consumers and providers, the IT department is responsible to act as a service intermediary (Hoyer and Stanoevska-Slabeva 2008). Because business users focus on solving daily business problems in the sales or accounting department and not on creating or adapting their operational environment, Enterprise Mashup platforms have to hide the complexity from the users. The figure below depicts the related services to implement the interaction process by using the Fundamental Modelling Concepts (FMC) notation. In contrast to the technical oriented UML notation, FMC focuses on human comprehension of complex systems[1].

The growing number of available mashupable components requires adequate discovery services for retrieval purposes. According to the user context (profile, preferences, social network it belongs to) relevant widgets are presented to the consumers who are able to select the relevant Mashup component. Sharing of information, experiences and knowledge within the community is a key driver for Enterprise Mashups. Besides the default semantic annotations (functional and non-functional

[1] http://www.fmc-modeling.org

qualities) defined by the provider, consumers are able to tag, recommend, or rate the mashable components. A catalogue manages all this information for widgets, Mashups, and users. The design of the widget components (wiring) is handled by the composition service that accesses the catalogue for the required information. As mentioned already before, good enough solutions lead to a converging design and run time. That implicates a direct integration between theses two services. From consumer perspective, no traditional deployment exists. They design their operational environment and execute it immediately. During the execution phase, the Enterprise Mashup environment monitors and protocols the usage of the mashable components. Based on the aggregated statistics resulting in popularity, availability or error rate information, consumers and providers get additional information to select or adapt a component or to publish a new one. Additionally, the accounting of the environment usage is calculated by this data. To administrate and monitor the performance of the Enterprise Mashup environment, the IT department needs adequate services to manage the running system.

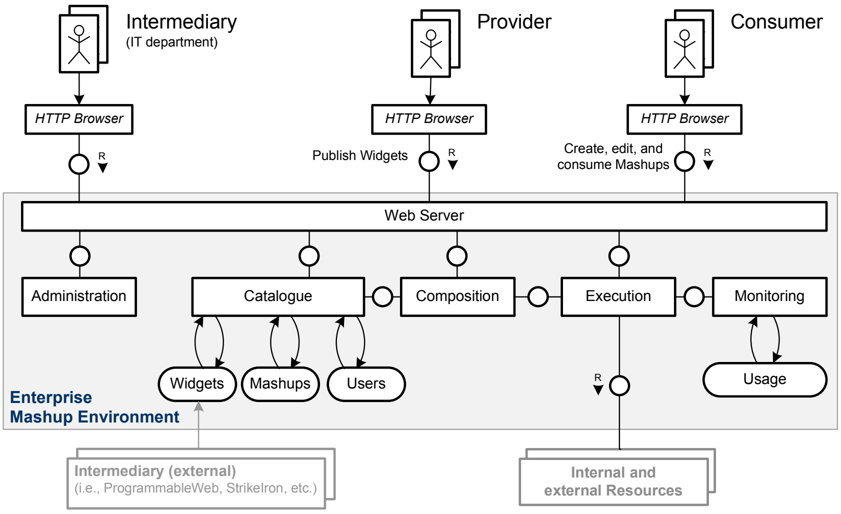


*Figure 5. Services of Enterprise Mashup Environments (FMC Notation).*

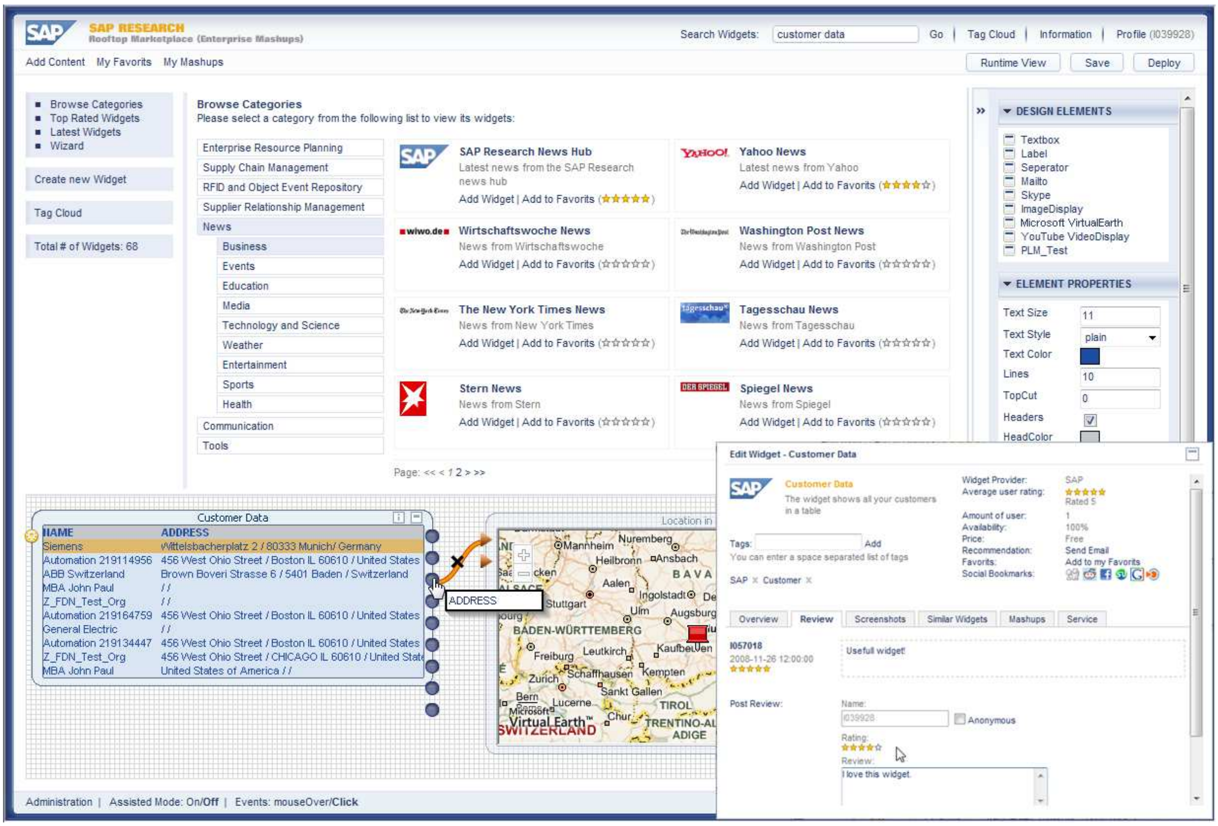
### 3.4 Infrastructure View

In contrast to existing applications (i.e., MS Excel or MS Access) created and managed by business units to address ad-hoc requirements, the technical infrastructure of Enterprise Mashups environments are managed and provided by the corporate IT department. Independent if the users from the business units act as consultant (widget layer) or as end user (Mashup layer), they are able to integrate easily their local resources encapsulated by Mashup components. Consumer-oriented Mashup environments like iGoogle, Netvibes, or Facebook consume mostly light weight Web-based resources sourced via RSS, ATOM, or JSON. Instead Enterprise Mashup environments integrate resources from legacy systems as well. Currently, the major vendors of enterprise systems like SAP, Oracle, or Microsoft enable their applications to service-oriented platforms which are based on established standards. For example, SAP provides more than 1500 so-called enterprise services described by WSDL. However, besides standardized Web Services, the implementation of first Mashup prototypes in Enterprise Mashup environments shows the relevance to integrate other resource types as well, i.e. XML files, data bases, or RPC.

Due to the open and Web-based character of Enterprise Mashups, wide accepted standards or protocols are required. This includes technical standards for the visual composition (piping and wiring) but also standardized accounting methods.

# 4 DEMONSTRATION – SAP RESEARCH ROOFTOP MARKETPLACE

This section is devoted to apply and demonstrate the design artefact by means of the SAP Research Rooftop Marketplace platform (Hoyer et al, 2009). It represents a prototype which allows the creation and adaptation of Enterprise Mashups according to the individual and heterogeneous needs. We used the designed reference model to develop the prototype by following the marketplace and collaborative character as elaborated in this paper. Thereby, the reference model structured the requirement analysis and the transfer to the technical specification of the platform.



Search Widgets: customer data

**Browse Categories**
Please select a category from the following list to view its widgets:

**SAP Research News Hub**
Latest news from the SAP Research news hub
Add Widget | Add to Favorits (★★★★★)

**Yahoo News**
Latest news from Yahoo
Add Widget | Add to Favorits (★★★★☆)

**Wirtschaftswoche News**
News from Wirtschaftswoche
Add Widget | Add to Favorits (☆☆☆☆☆)

**Washington Post News**
News from Washington Post
Add Widget | Add to Favorits (☆☆☆☆☆)

**The New York Times News**
News from New York Times
Add Widget | Add to Favorits (☆☆☆☆☆)

**Tagesschau News**
News from Tagesschau
Add Widget | Add to Favorits (☆☆☆☆☆)

**Stern News**
News from Stern
Add Widget | Add to Favorits (☆☆☆☆☆)

**Spiegel News**
News from Spiegel
Add Widget | Add to Favorits (☆☆☆☆☆)

| | |
|---|---|
| Text Size | 11 |
| Text Style | plain |
| Text Color | ■ |
| Lines | 10 |
| TopCut | 0 |
| Headers | ☑ |
| HeadColor | □ |

Customer Data

| NAME | ADDRESS |
|---|---|
| Siemens | Wittelsbacherplatz 2 / 80333 Munich/ Germany |
| Automation 219114956 | 456 West Ohio Street / Boston IL 60610 / United States |
| ABB Switzerland | Brown Boveri Strasse 6 / 5401 Baden / Switzerland |
| MBA John Paul | / / |
| Z_FDN_Test_Org | / / |
| Automation 219164759 | 456 West Ohio Street / Boston IL 60610 / United States |
| General Electric | / / |
| Automation 219134447 | 456 West Ohio Street / Boston IL 60610 / United States |
| Z_FDN_Test_Org | 456 West Ohio Street / CHICAGO IL 60610 / United State |
| MBA John Paul | United States of America / / |

**Edit Widget - Customer Data**

Customer Data
The widget shows all your customers in a table

Tags: Add
You can enter a space separated list of tags
SAP × Customer ×

| | |
|---|---|
| Widget Provider: | SAP |
| Average user rating: | ★★★★★<br>Rated 5 |
| Amount of user: | 1 |
| Availability: | 100% |
| Price: | Free |
| Recommendation: | Send Email |
| Favorits: | Add to my Favorits |
| Social Bookmarks: | |

I057018
2008-11-26 12:00:00
★★★★★
Usefull widget!

Post Review:
Name: I039928 □ Anonymous
Rating: ★★★★☆
Review: I love this widget.

*Figure 6. SAP Research Rooftop Marketplace.*

The SAP Research Rooftop Marketplace itself is a Web-based application based on AJAX (Asynchronous JavaScript and XML) and is internally available to all SAP employees without installation. In this sense, the platform is provided by SAP Research acting as an intermediary. By using a Single Sign On (SSO) login process, users (provider and consumer) are able to register to SAP Research Rooftop automatically. After defining the individual user profile, users are able to discover the Mashup community which provides information about Enterprise Mashups in general as well as features and demonstrations (videos) of the SAP Research Rooftop tool in specific. A catalogue allows the discovery of the provided Mashups and widgets. Besides the browsing of predefined categories, the user is able to select top rated, most popular or latest widgets. Each widget is annotated by extensive information provided by all three agent roles (see figure 6). The actual consumer is able to rate, to tag, or to recommend a widget. In case a user has created an Enterprise Mashup, he is able to send a link to his colleague who can consume immediately the component. On the other side, providers can specify a default description and the business model (i.e. the costs) of a widget. The SAP Research Rooftop Marketplace platform itself (intermediary) monitors the consumed widgets and Mashups continuously (i.e., popularity and availability) and provides the information to the consumer and provider.

The composition of widgets is depicted at the bottom of in the figure above. The customer data widget is wired with a map by connecting the input and output parameter (orange line) represented by the red bullets on the left (input) and the blue bullets right (output) side of the widget. The configuration and information of a widget can be easily accessed by clicking on the *i(nfo)* button of the widget. It allows that the consumer can directly contribute to the community by rating or tagging a widget without changing to a different view which covers the knowledge/ intention phase. According to the described marketplace and collaborative characteristic of Enterprise Mashups, the SAP Research Rooftop Marketplace platform integrates these two market phases. The figure above at the bottom right indicates the available information of the "Customer Data" widget and how a consumer can add easily a rating and comment to a widget.

Coming back to the composition environment, the real data of the "Customer Data" widgets are displayed already at the design time as depicted in the figure. If the user selects a customer (Siemens in Munich, Germany) in the widget, the address is updated in the map (in this case Microsoft Virtual Earth). So, there exist no separation between the design and runtime within the SAP Research Rooftop Marketplace prototype. By shifting to the runtime view, only the configuration capabilities like adding new widgets are logged.

# 5 CONCLUSION

The aim of this paper is the design of a reference model for grassroots Enterprise Mashups environments serving as foundation to develop and structure Enterprise Mashup environments. In order to achieve this, we follow the design science methodology. After defining the main terms related to Enterprise Mashups and reference models, we presented a designed reference model for Enterprise Mashups by leveraging the St. Gallen Media Reference Model. Thereby, we took advantage of the observed and identified similarities to electronic markets and collaborative characteristics. Instead of following the traditional software development phases (requirements, specification, development, testing and deployment), we propose the structure of market transactions. The roles, required services, and relationships between the interacting agents (provider, consumer and intermediary) were modelled. By means of the SAP Research Rooftop Marketplace, we demonstrate the application of the designed reference model for structuring the development of an Enterprise Mashup environment.

What is still missing is a broader application of the reference model for Enterprise Mashups in other areas and its further verification in an iterative design cycle ("Design as a search process") according to the design science methodology (Peffers et al. 2008, Hevner et al. 2004). Further research will deal with the application of the reference model to structure further Enterprise Mashup environments. In particular, we have applied the reference model in the frame of the EU funded FAST/ EzWeb project (http://fast.morfeo-project.eu) that covers the Mashup and widget layer.

The designed reference model for Enterprise Mashups provides furthermore only a first generic framework that helps to understand the organizational and managerial challenges of the Mashup paradigm in enterprise environments. Besides the structuring of requirements for the development of Enterprise Mashup platforms, the generic reference model has to be extended and operationalized with unhandled managerial aspects. The loosely coupled user-friendly building blocks both from internal and external IT systems require also a governance, quality, and security concept defining who is able to access a widget. The inclusion of these aspects would provide a more detailed and different models of the community and interaction view.

## References

Anderson, C. (2006). The Long Tail: How endless choice is creating unlimited demand. Random House Business Book, London.

Becker, J., Rosemann, B., Schuette, R. (1995). Guidelines for Modeling. In Wirtschaftsinformatik, 37(5), 435-445. [in German]

Benkler, Y. (2006). The Wealth of Networks. How Social Production Transforms Markets and Freedom. Yale University Press, New Haven and London.

Bradley, A. (2007). Reference Architecture for Enterprise Mashups. Gartner Research, ID Number G00151491.

Braun, R., Esswein, W. (2007). Classification of Reference Models. In: Decker, R., Lenz, H.-J. (Eds): Studies in Classification, Data Analysis, and Knowledge Management, Springer, Berlin, 401-408.

Carrier, N., Deutsch, T., Gruber, C., Heid, M., Jarret, L.L. (2008). The Business Case for Enterprise Mashups, IBM White Paper, August 2008.

Cherbakov, L., Bravery, A., Goodman, B., Pandya, A., and Bagget, J. (2007). Changing the corporate IT development model: Tapping the power of grassrots computing. IBM System Journals 46(4).

Daniel, F., Matera, M., Yu, J. Benatalla, B. Saint-Paul, R., and Casati, F. (2007). Understanding UI Integration. A Survey of Problems, Technologies, and Opportunities. IEEE Internet Computing, 11(3), 59-66.

Fettke, P., Loos, P: (2007). Perspectives on Reference Modeling. In Fettke, P. and Loos, P. (Eds.) Reference Modeling for Business Systems Analysis, Idea Group, pp. 1-20.

Frank, U. (2007). Evaluation of Reference Models. In Fettke, P. and Loos, P. (ed.) Reference Modeling for Business Systems Analysis, Idea Group, pp. 118-140.

Hevner, A.R., March, S.T., Park, J., and Ram, S. (2004). Design Science in Information Systems Research. MIS Quarterly, 28(1), 75-105.

Hoyer, V., Gilles, F., Fleischmann, K., Dreiling, A., Stanoevska-Slabeva, K. (2009). SAP Research RoofTop Marketplace, Proceedings of the 2nd Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009) in conjunction with the 18th International World Wide Web Conference (WWW2009), Madrid, Spain.

Hoyer, V. and Stanoevska-Slabeva, K. (2009). Generic Business Model Types for Enterprise Mashup Intermediaries, Proceedings of the 15th American Conference on Information Systems (AMCIS)

Hoyer, V. and Staneovska-Slabeva, K. (2008). The Changing Role of IT Departments in Enterprise Mashup Environments. Proceedings of the 2nd International on Web APIs and Service Mashups.

Hoyer, V., Stanoevska-Slabeva, K., Janner, T., and Schroth, C. (2008). Enterprise Mashups: Design Principles towards the Long Tail of User Needs. In IEEE International Conference on Service Computing (SCC'08). Volume 2, 601-602.

Legner, C. (2007). Is there a Market for Web Services? – An Analysis of Web Service Directories. In Proceedings of the 1st International Workshop on Web APIs and Services Mashups.

March S. and Smith G. (1995). Design and Natural Science Research on Information Technology. Decision Support Systems, 15(4), 251-266.

Maximilien, E.M., Ranabhu, A., Godmadam, K. (2008). An Online Platform for Web APIs and Service Mashups. IEEE Internet Computing, 12(5), 32-43.

Papazoglou, M.P. (2003). Service-Oriented Computing: Concepts, Characteristics and Directions. In Proceedings of the 4th International Conference on Web Information Systems Engineering.

Peffers, K., Tuunanen, T., Rothenberger, M., and Chatterjee, S. (2008). A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems, 24(3), 45-77.

Schmid, B. (1997). The Concept of Media. In Proceedings of the Fourth Research Symposium on Electronic Markets, 77-90.

Schmid, B. and Lindemann, M. (1998). Elements of a Reference Model for Electronic Markets. In Proceedings of the 31st Hawaii International Conference on Service Sciences (HICCS).

Schroth, C., Schmid, B. (2009). Reference Architecture for Cross-Company Electronic Collaboration. International Journal of e-Collaboration, 5(2), 75-91.

Stanoevska-Slabeva, K. (2003). Towards a Reference Model for M-Commerce. Proceedings of European Conference on Information Systems (ECIS'03).

Yu, J., Benatallah, B., Casati, F., Daniel, F. (2008). Understanding Mashup Development. IEEE Internet Computing, 12(5), 44-52.