Design Principles of Enterprise Mashups

Volker Hoyer¹ ² and Katarina Stanoevska-Slabeva²

¹ SAP Research CEC St. Gallen, Switzerland
² University of St. Gallen, =mcm institute, Switzerland
1. Motivation and Challenges
2. What are Enterprise Mashups?
3. Emerging Design Principles
4. Cases Study: FAST Platform
5. Conclusion and Outlook
“Yes we can!”
Empowering the Non-Technical Business Users

New approaches/ tools aim at enabling the mass of end-users to create enterprise-level applications according to their individual needs

- **“Enterprise Mashup[s] ... unleash a tremendous amount of value”** (Forrester)
- **Top 10 Strategic Technologies for 2009** (Gartner)
- **$700 Million Market** by 2013 (Forrester)
Motivation and Challenges

Business Situation
- Long tail of user needs (end-users are increasingly requiring individualized applications meeting exactly their daily needs)
- Situation application (ad hoc application) are needed

Shortcomings of Service Oriented Architectures
- Active end-user integration is missing (high complexity of the relevant standards like SOAP, WSDL, UDDI, BPEL, etc.)
- No flexibility to address ad hoc user requirements
- Innovation power of user communities is not used

Enterprise Mashups as a new Paradigm for Situational Applications
  → Emerging of new design principles
1. Motivation and Challenges
2. What are Enterprise Mashups?
3. Emerging Design Principles
4. Cases Study: FAST Platform
5. Conclusion and Outlook
“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.”
## Enterprise Mashups
### Enterprise Mashup versus SOA

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Traditional IT Development (Service-Oriented Architecture)</th>
<th>Enterprise Mashups</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scope</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time-to-Value</td>
<td>Many weeks, months, or even years</td>
<td>Minutes, hours or days</td>
</tr>
<tr>
<td>Lifespan</td>
<td>Long lived application</td>
<td>Variable, often short application</td>
</tr>
<tr>
<td>Application type</td>
<td>Strategic, standard applications</td>
<td>Backlog of simple, tactical, and individual applications (“long tail”)</td>
</tr>
<tr>
<td>Functional Requirements</td>
<td>Defined by limited number of users, IT needs to freeze requirements to move to development</td>
<td>As requirements change, Enterprise Mashups usually changes to accommodate business changes</td>
</tr>
<tr>
<td>Non Functional Requirements</td>
<td>Resources allocated to address concerns for performance, availability and security, robust solutions</td>
<td>Little or no focus on scalability, maintainability, availability, etc.</td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Development Phases</td>
<td>Well defined and scheduled (requirement, specification, implementation, testing, deployment)</td>
<td>Ad hoc or good enough solution to address and immediate need</td>
</tr>
<tr>
<td>Governance</td>
<td>Formal, centralized</td>
<td>Decentralized, community-driven</td>
</tr>
<tr>
<td>Evolution</td>
<td>Top-down, centrally</td>
<td>Organic</td>
</tr>
<tr>
<td><strong>Users</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application builders</td>
<td>IT department or external experts (developer skills)</td>
<td>Line of business, individuals or groups (limited or no programming skills)</td>
</tr>
<tr>
<td>Target users</td>
<td>Large groups</td>
<td>Small teams or even individuals</td>
</tr>
</tbody>
</table>
1. Motivation and Challenges
2. What are Enterprise Mashups?
3. **Emerging Design Principles**
4. Cases Study: FAST Platform
5. Conclusion and Outlook
Design Principles
Enterprise Mashup Stack (Architecture)

Enterprise Mashup Stack

**Mashup**
- Mashup scenario creation
- Consuming and personalizing individual environment (e.g., changing background color, hiding buttons, etc.)

**Widgets**
- Pre-built widgets from the catalogue
- Connecting widgets by wiring in-/out-ports
- Binding generic UIs to resources/services
- Transforming and aggregating data

**Resources**
- Web Services, RSS/Atom Feed, XML file, etc.
- Creates and deploys services - makes resources available (e.g., RSS Feeds, SAP Enterprise Services)

User Roles

- **End-Users** run Mashup scenario
  - Consume Mashup scenario
  - Personalize individual environment (e.g., can change the background color, hide buttons, etc.)

- **Key users** create Mashup Scenario
  - Add pre-build widgets (from the catalogue)
  - Connecting widgets by wiring their in-/out-ports

- **Consultants** create Widgets
  - Binding generic UIs to resources/services
  - Transforming and aggregating data

- **Developers** implement the services
  - Creates and deploys services - makes resources available (e.g., RSS Feeds, SAP Enterprise Services)
New services provided by emerging intermediaries
- Monitor continuously parameters (availability, response latency)
- Provide performance metrics and other evaluation results
- Adequate description of resources for retrieval by the actual end-users (consumer)

Integration of consumers
- Collaborative tagging of resources resulting in an organic (bottom-up) taxonomy
- User rating based on popularity and relevance
- Recommending of resources
Characteristics

- Co-innovation and co-production of enterprise applications through the actual end-users
- End-users self-organize their individual workspace ("quick and dirty")
- Willingness of sharing of experiences with thousands or even millions on the Web
- Foster the continuously improvement of resources
Design Principles
Lightweight Resource Composition (Piping vs Wiring)

Mashup

Wiring

Widgets

Widget 1
Pipe 1
Pipe 2
Pipe n

Widget 2
Pipe 1
Pipe 2
Pipe n

Widget 3
Pipe 1
Pipe 2
Pipe n

Resources

Piping

Piping Operators
- Aggregation
- Merging
- Filter
- ...

© SAP 2009 / Design Principles of Enterprise Mashups / Page 13
Characteristics
- Even if users are more tolerant of bugs and poor performance, a stable Enterprise Mashup environment is required
- Building individual applications *quick and dirty* with internal and external components

New Development Process
- Agile development model focusing on the actual working application
- Operative phase is explicit part (Converging design and run time)
- Disciplined build and development process
1. Motivation and Challenges
2. What are Enterprise Mashups?
3. Emerging Design Principles
4. Cases Study: FAST Platform
5. Conclusion and Outlook
Case Study
FAST Platform (Screenshot)
Agenda

1. Motivation and Challenges
2. What are Enterprise Mashups?
3. Emerging Design Principles
4. Cases Study: FAST Platform
5. Conclusion and Outlook
Conclusion and Outlook

**New Emerging Design Principles**
- Enterprise Mashup Stack
- Emerging Intermediaries
- Mass Collaboration
- Lightweight Resource Composition
- Perpetual Beta Development Model

**Prototypical Implementations in the EU funded Project**

http://fast.morfeo-project.eu
Thank you!

Volker.Hoyer@sap.com, Katarina.Stanoevska@unisg.ch