



UNIVERSITY
OF TRENTO - Italy

Information Engineering
and Computer Science Department

Web Mashups: the Web as Integration Platform for End Users

Florian Daniel

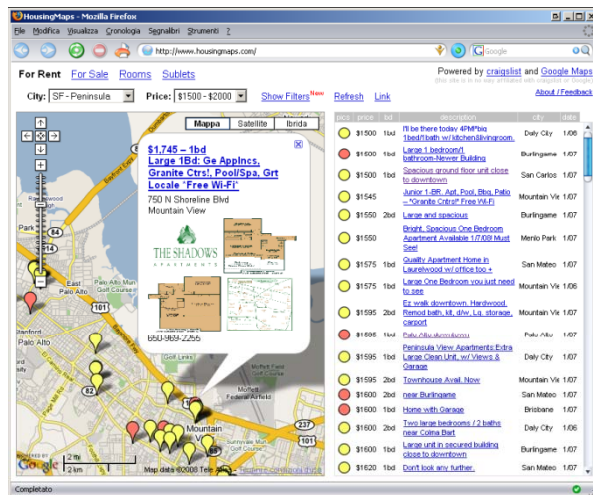
What are we talking about?

- **Mashup** – possible definitions
 - “...a mashup is a web application that combines **data** from more than one source into a single integrated tool...” [wikipedia.com]
 - “...you can integrate two or more [...] **Web APIs** to create something new and unique, known as a mashup...” [*]
 - “...a mashup is a web application that is developed by composing **data, application logic, and/or user interfaces** originating from disparate web sources...” [me]
- Similar terms: service mashups, data mashups

* http://www.ibm.com/developerworks/webservices/library/ws-soa-mashups/index.html?S_TACT=105AGX04&S_CMP=EDU

Let's see an example

- The HousingMaps application (<http://www.housingmaps.com>) composed of:
 - Google Maps (<http://maps.google.com>)
 - Craigslist (<http://www.craigslist.com>)



HousingMaps - Mozilla Firefox

File Modifica Visualizza Cronologia Segnalibri Strumenti ?

http://www.housingmaps.com/

Più visitati Own Private News University mashArt Compas Master CSS Journals Confs Various Google Maps Google Docs

For Rent For Sale Rooms Sublets

City: SF - Peninsula Price: \$1500 - \$2000 Show Filters New Refresh Link

Powered by Craigslist and Google Maps (affiliated with craigslist or Google)

Own application logic/UI

GoogleMaps

\$1,890 - 2bd

Lovely Duplex Unit in California Avenue Neighborhood -

Park Blvd & College Ave
Palo Alto

650-248-6605 / [email](#)

Craigslist

price	bd	title	location	date
\$1957	2bd	Get Pees Living!	San Mateo	1/14
\$1700	2bd	Move-In Special*Remodeled 2BR*	Palo Alto	1/14
\$1599	2bd	Newly Remodeled, spacious, Garage parking, available now!	Redwood Cr	1/14
\$1695	2bd	2 Bedroom Special Today! Available Now!	Redwood Cr	1/14
\$1600	2bd	Nice Duplex -	Redwood Cr	1/14
\$1600	1bd	Beautiful, Remodeled Apt, Near Downtown - On El Camino # -	Menlo Park	1/14
\$2000	3bd	3br 1bath + spare room & huge yard & driveway -	Palo Alto	1/14
\$1800		Newly remodeled one bedrooms, w/ washer & dryer -	Palo Alto	1/14
\$1595	2bd	2br/2ba 1001 Continentals Way Chateau D'Orn Apartments -	Belmont	1/14
\$1950	3bd	3br/2ba 2211 Hastings Drive Carlmont Heights Apartments -	Belmont	1/14
\$1595	2bd	2 BR/1BA Apt. Close to Downtown Ssf -	South San F	1/14
\$1950	2bd	Adorable and Cozy Redwood City Duplex -	Redwood Cr	1/14
\$1750	1bd	Beautiful One Bed Loft Condo In San Mateo Hills WView	San Mateo	1/14
\$1698	2bd	Townhouse w/ Fireplace (826-C) -	South San F	1/14
\$2000	2bd	2.5 Bath (457 Sierra Vista # 11) -	Mountain Vie	1/14
\$1800	2bd	2 BR House in Downtown San Mateo, 2 blocks from Train, shops, movies -	San Mateo	1/14

Map data ©2009 Tele Atlas - Termini e condizioni d'uso

Completato

We'll have a look at

1. The mashup phenomenon
2. Most prominent technologies
3. The tool-assisted development
4. Universal composition approach
5. The mashup development process
6. Open research issues and conclusion

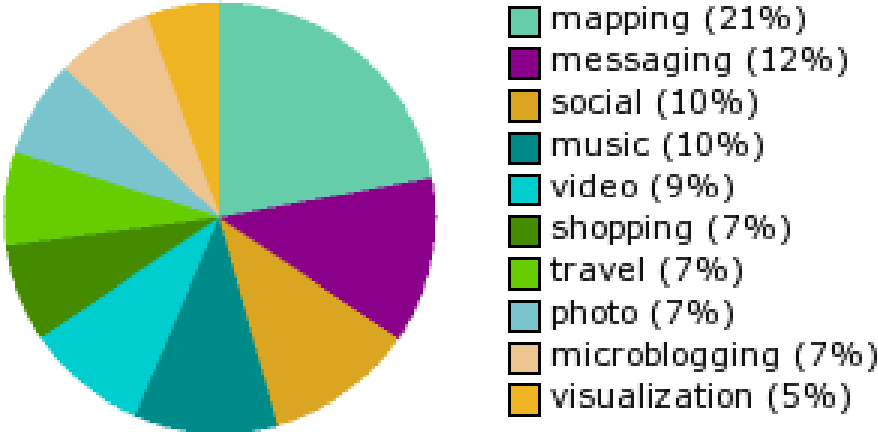
About the phenomenon

- Young **integrating** practice using the Web as platform
- Highly **user-driven**
 - Oftentimes the actual providers of content/functionality are not even aware of being “wrapped”
 - Google Maps example: initially skilled users hacked the AJAX code of the application
- Strong **evolution**: from hacking to first systematic development approaches in a few years
- **programmableweb.com**
 - Most comprehensive collection of **mashups** and **web APIs** available on the Web

of disobedient users who
develop own applications
on the Web!

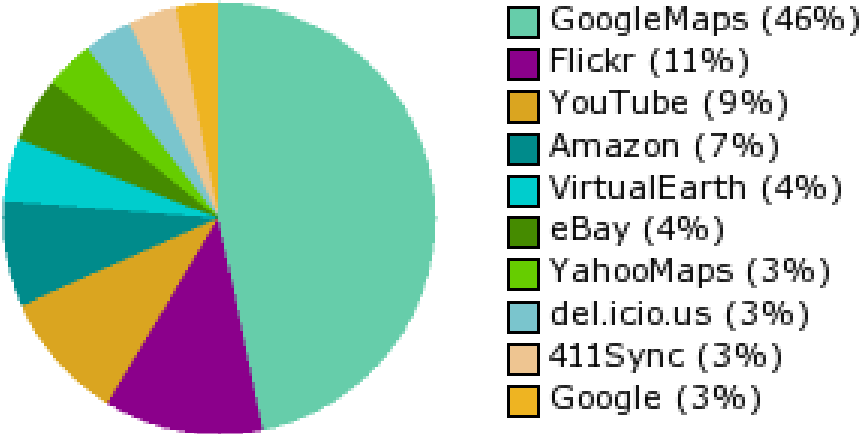
programmableweb.com

- Most popular categories of **mashups**



ProgrammableWeb.com 01/14/09

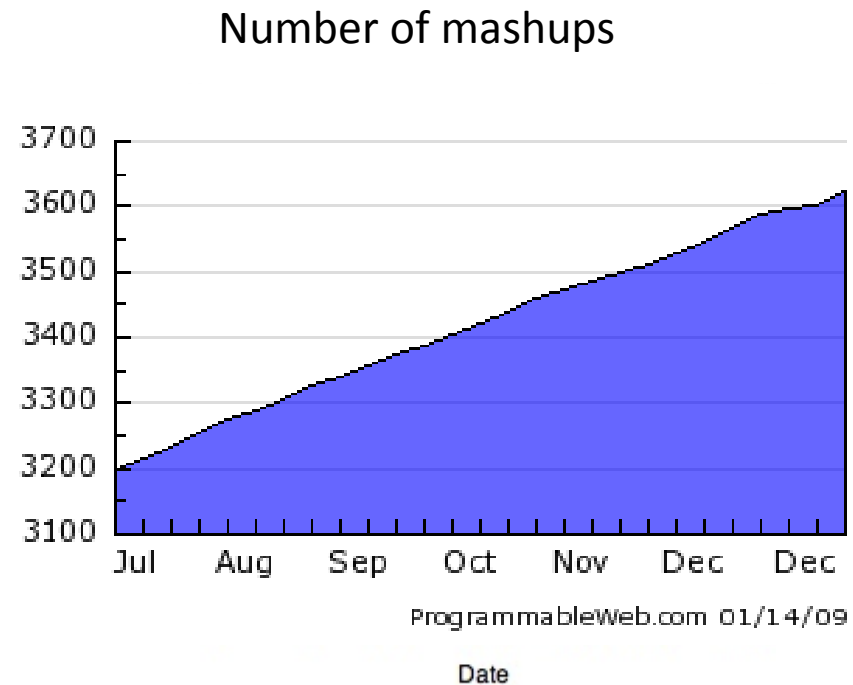
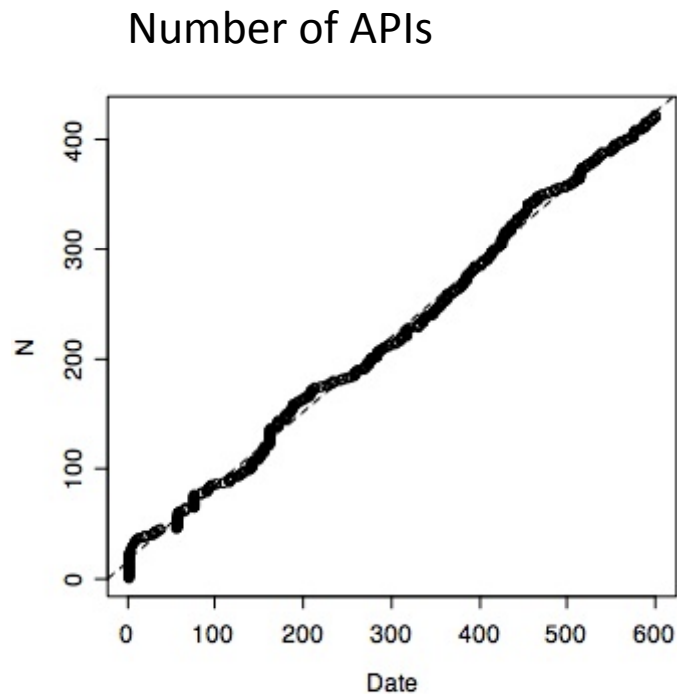
- Most popular web **APIs**



ProgrammableWeb.com 01/14/09

A constant growth

- Constant growth rate since programmableweb.com went online (over 600 days) [presentation by Michael Weiss, Carleton University]



Web engineering and mashups

- **Manual** development: static (plain HTML) and dynamic (CGI, PHP, JSP,...) pages coded via simple text editors
- **Tool-assisted** development: web-specific development tools (e.g., Dreamweaver) augment productivity by automating web-specific concerns
- **Model-driven** development: graphical modeling tools enable developers to reason at a high level of abstraction, to “draw” an application, and to automatically generate its code
- **Mashups**: trend toward the user-driven development of web applications in Web 2.0

Web 2.0 as enabler

- **Web 2.0?** Again, there are lots of different (and sometimes diverging) definitions:
 - “Web 2.0 is a term describing the trend in use of World Wide Web technology and web design that aims to enhance creativity, information sharing, and, most notably, collaboration among users...” [wikipedia.com]
 - “Web 2.0 is best described as a core set of patterns that are observable in applications that share the Web 2.0 label. These patterns are services, simplicity, and community...” [*]
 - “I’m still not sure Web 2.0 is an appropriate term...” [me]

* http://www.ibm.com/developerworks/webservices/library/ws-soa-mashups/index.html?S_TACT=105AGX04&S_CMP=EDU

Web 2.0 as enabler

- I would rather say that over the last years we have been witnessing **two main trends** on the Web:
 - User participation in the **content creation** process
 - User participation in the **development** process
- Which are enabled or fostered by:
 - **Communities, social networks, blogs**
(e.g., Wikipedia, Flickr,...)
 - **Simplicity of usage**: intuitive, interactive applications
 - **Simplicity of development**: novel and standardized web technologies

Technologies

- Here the most prominent ones:
 - XML
 - RSS and Atom
 - DHTML = HTML + DOM + JavaScript
 - AJAX
 - Flash, JavaFX, Silverlight
 - PHP, Ruby,...
 - SOAP/RESTful web services

XML

- XML = eXtensible Markup Language
 - XML is actually a **meta-language**, that is, a language with which we can define other languages
- Ingredients: tags (elements), attributes, name spaces, DTDs, XML schema definitions,...
- Why important?
 - Technology-independent (e.g., independent of programming language, platform)
 - >> Highly **portable**
 - >> **Enabler** of the most recent communication protocols

RSS and Atom

- Two **applications of XML** (i.e., languages) for the syndication and exchange of contents over the Web (**feeds**)
- Very popular in blogs, newspapers, social web sites,...
- Actually very simple (often even static) web resources with **references** to detailed contents
 - Example: latest news by the New York Times
 - Dynamic behavior achieved through dedicated readers

DHTML

```
<html>
  <head>
    <title>DHTML Test</title>
    <script type="text/javascript">
      function changeColor() {
        document.getElementById("div1").style.color = "blue"; }
    </script>
  </head>

  <body>
    <div id="div1" style="color:red;" onClick="changeColor();">
      If you click me, I change my color :-)
    </div>
  </body>
</html>
```

AJAX

- AJAX = Asynchronous JavaScript And XML
- Ingredients of AJAX applications:
 - DHTML
 - JavaScript
 - **XMLHttpRequest** object
- Enables **asynchronous** communications between the JavaScript logic running inside the browser and the web server >> **interactive** HTML applications (RIA)
- Shortcoming:
 - JavaScript code is hard to **debug**
 - JavaScript **sandbox** mechanism

Flash, JavaFX, Silverlight

- Adobe Flash, JavaFX, and Microsoft Silverlight follow an alternative approach to the development of RIA applications:
 - Full-fledged, light-weight application development **platforms** that run inside the client browser
 - Require the installation of a dedicated **browser plug-in**
 - Are based on simple **scripting** languages (JS-like)
 - Enable the development of highly interactive and multimedia user interfaces featuring **desktop-like** user experiences

PHP, Ruby,...

- Popular, simple **server-side scripting** languages
- **Free** plug-ins for most web servers
- Availability of powerful free code libraries (e.g., for the management of HTML templates: PHPTAL)
- Availability of advanced web development frameworks (e.g., Ruby on Rails with full support for MVC-based web applications)

SOAP/WSDL web services

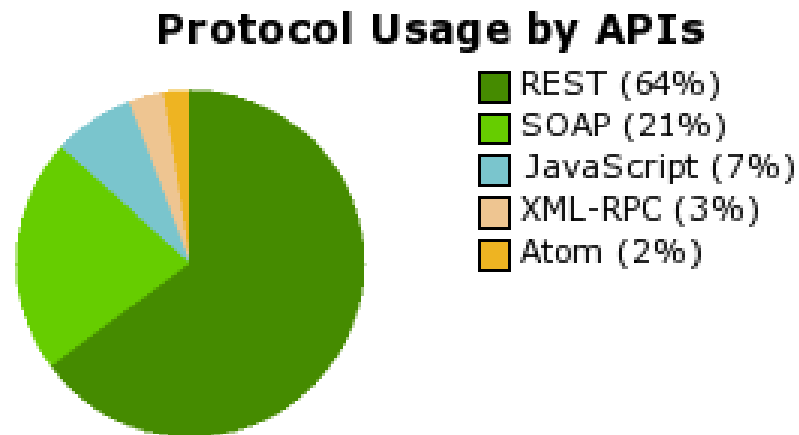
- **Programmatic interfaces** accessible over the Web
 - **WSDL** = Web Service Description Language
 - Abstract service description language (tech-agnostic)
 - **SOAP** = Simple Object Access Protocol
 - XML message exchange protocol
- **SOA** = Service-Oriented Architecture
 - Producer, consumer, registry (virtual marketplaces)
- Complex advanced features: security, reliability, transactions, addressing,...
- Orchestration and choreography

RESTful web services

- A new **architectural style** of developing web services
 - Operations explicitly use **HTTP methods** (GET, POST, PUT, DELETE)
 - Services are **stateless** (no session data at the server side)
 - Access via directory structure-like **URIs**
 - XML or JSON over **HTTP**
- Benefits/shortcomings:
 - Simplicity and immediacy
 - No big overhead for composing and parsing messages
 - No support for advanced features (e.g., security, reliable messaging, transactions)

“Protocol” usage by APIs

- From programmableweb.com:

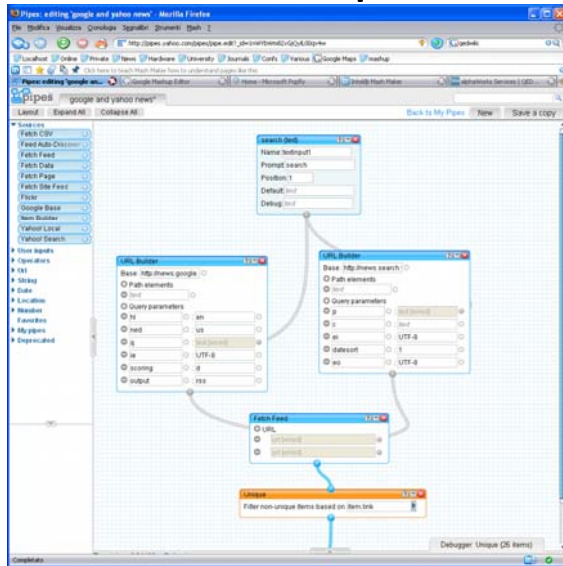


ProgrammableWeb.com 01/14/09

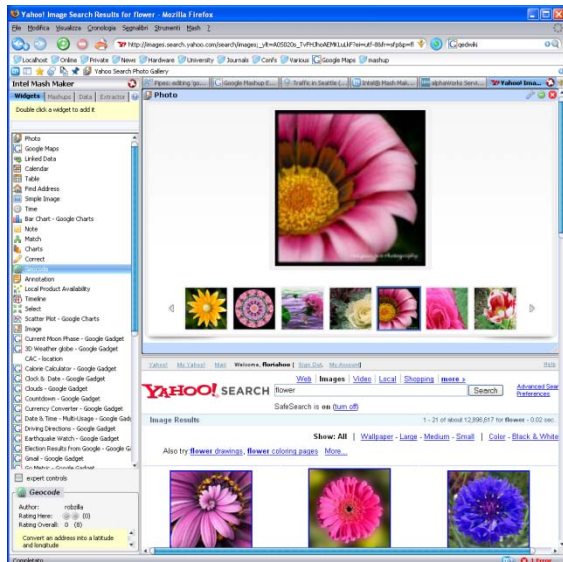
Mashup tools

- Mashup tools (or mashup makers) aim to
 - **Simplify** the overall development process
 - Provide easy-to-use **development** instruments
 - Provide dedicated **execution** environments
 - Enable even the **unskilled** user to mash up own applications

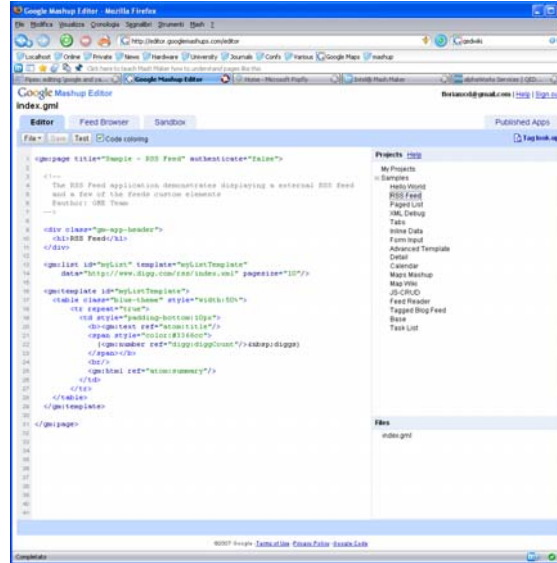
Yahoo Pipes



Intel Mash Maker

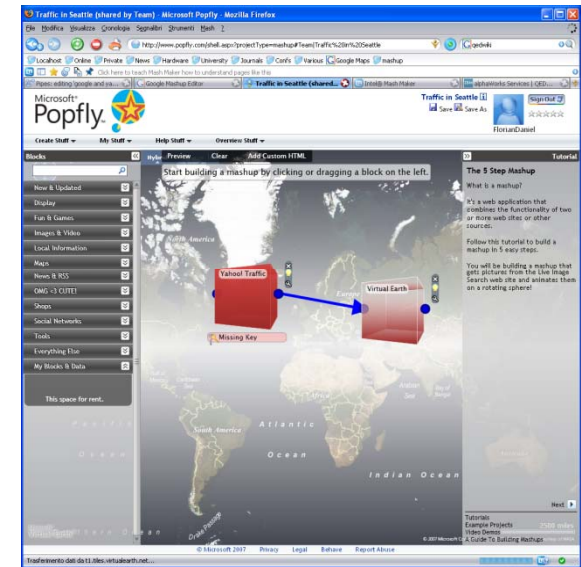


Google Mashup Editor

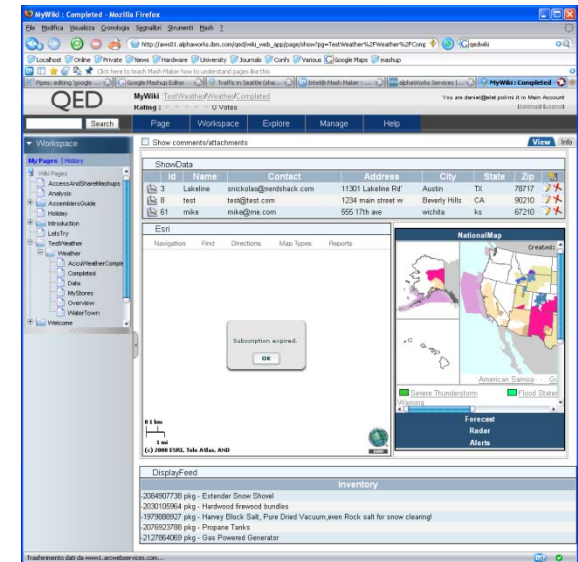


- <http://pipes.yahoo.com>
- <http://editor.googlemashups.com>
- <http://www.popfly.com>
- <http://mashmaker.intel.com>
- <http://services.alphaworks.ibm.com/gedwiki>

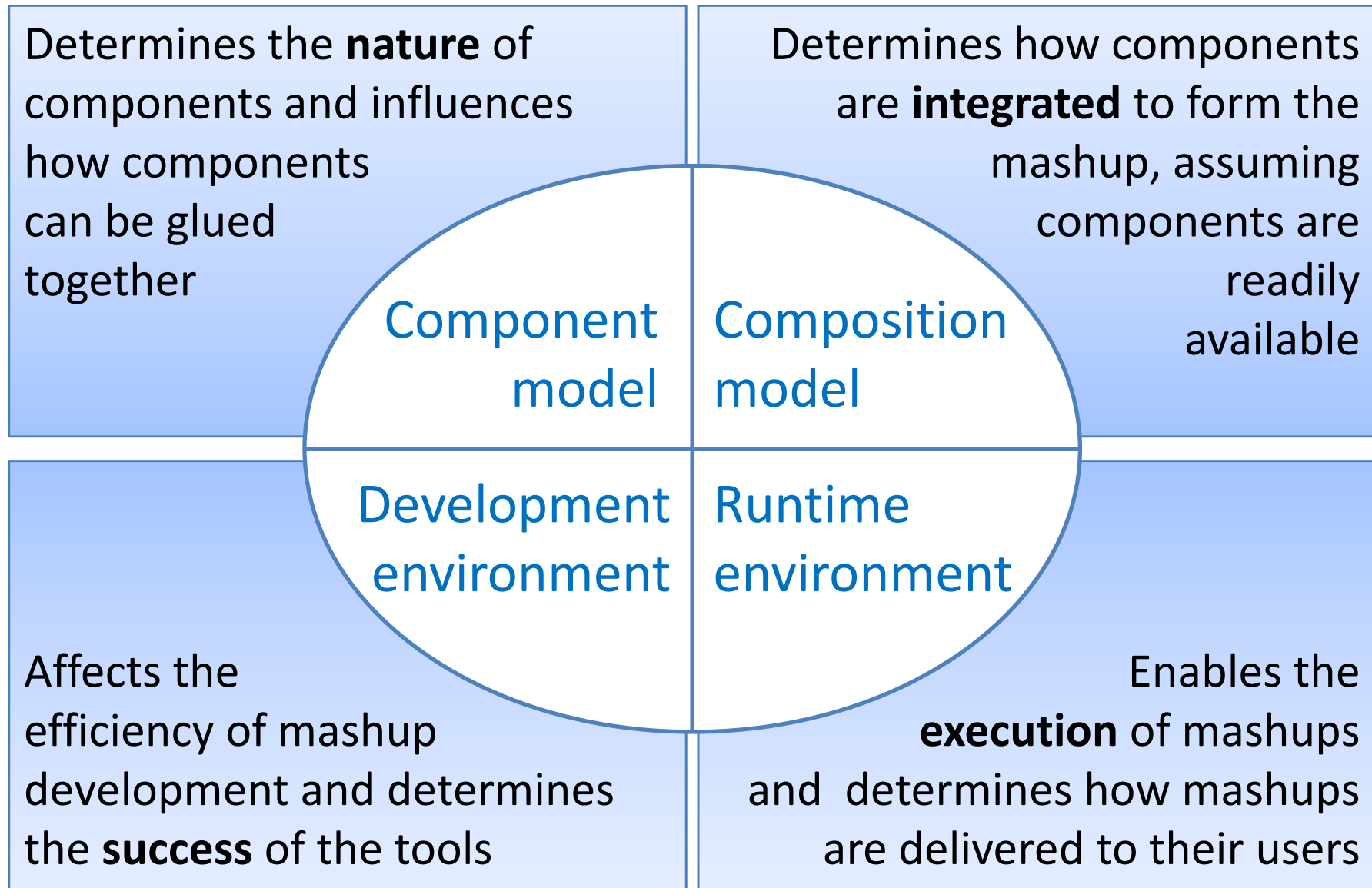
Microsoft Popfly



IBM QEDWiki



Characterizing the tools



Component model

- **Type**
 - Data (DA), application logic (AL), user interface (UI)
- **Interface**
 - CRUD (Create-Read-Update-Delete) interface, APIs in specific programming languages or in IDL/WSDL, XML/HTML markup, or only graphical user interface elements for end users
- **Extensibility**
 - Is the component model extensible?

Composition model

- **Output type**
 - Data (DA), application logic (AL), user interface (UI) output
- **Orchestration style**
 - Flow-based approaches vs. event-based approaches vs. layout-based approaches
- **Data passing style**
 - Data flow approach (data flows from component to component) vs. blackboard approach (data is written to variables that can be used by components)
- **Instance model**
 - Instance-based or continuous

Development environment

- **Interface paradigm and target users**
 - Visual drag-and-drop features, textual editors, or a combination of the two. The interface can be targeted at the average web user, at advanced (tech-savvy) users, or at programmers
- **System requirements**
 - The execution of the mashup tool may require specific additional modules, plug-ins, or browser features

Runtime environment

- **Deployment style**
 - Stand-alone on own web server or hosted on third party web server
- **Runtime location**
 - Server side, client side, or both. Server-side approaches: engine-based or web application-based
- **System requirements**
 - Browser plug-ins or extensions?
- **Scalability**
 - Number of data sources, in the number of models (compositions), or in the number of users

Yahoo Pipes

- **Component model:** supports DA and AL components through operators that provide access to RSS/Atom feeds and external Web Services; DA components have a read only interface, and external web services have a RESTful interface based on JSON or RSS; the component models are fixed.
- **Composition model:** probably the best representative of DA output (pipes are RSS feeds); its graphical modeling language is flow-based; accordingly, data is also passed via data flows; Pipes is instance-based.
- **Development environment:** provides a pure visual drag-and-drop AJAX editor targeted at advanced users with basic programming skills; the editor is executed in a standard web browser with support for the XMLHttpRequest JavaScript object.
- **Runtime environment:** compositions are hosted on a Yahoo server; pipes are computed and assembled at the server side (apparently engine-based); hence, executing a pipe does not pose any particular system requirements on the client; however, the server-side engine that executes the pipes might suffer if many pipes are run or the same pipe is accessed by a large number of users or composed of hundreds of sources.

Google Mashup Editor

- **Component model:** supports DA, AL, and UI components; DA components are typically interfaced via markup, AL components via JavaScript, and UI components via both markup and JavaScript; component models are flexible.
- **Composition model:** produces UI output; it is event-based; data passing is achieved through event parameters in a data flow fashion; components with UI are continuous.
- **Development environment:** supports a browser-based textual AJAX editor with syntax highlighting and automatic tag completion targeted at programmers; it can be fully executed in a standard web browser.
- **Runtime environment:** mashups are hosted on a Google server; mashups are executed at the server side; there are no particular system requirements; mashups are compiled into conventional web applications.

Microsoft Popfly

- **Component model:** supports DA, AL, and UI components; all components are interfaced via JavaScript; component models are extensible.
- **Composition model:** produces UI output; it uses an event-based orchestration style; data passing is based on a data flow approach; components are continuous.
- **Development environment:** offers a graphical and textual editor with drag-and-drop support for web users; Popfly is based on Microsoft's Silverlight technology, a mandatory browser plug-in.
- **Runtime environment:** mashups are hosted on a server by Microsoft; the execution of a Popfly application is however performed at the client side; the execution typically requires the availability of the Silverlight plug-in; the client-side execution facilitates scalability, as the integration of multiple sources is mostly done at the client side.

Intel Mash Maker

- **Component model:** supports DA components extracted from annotated web pages (e.g., table and map) and UI widgets; their interface can be interpreted as XML markup; the component models are fixed.
- **Composition model:** focuses on UI output; contents are glued together either in a layout-based style (whole page merging) or in a flow-like style (item-wise merging); data extracted from annotated web pages is used in a blackboard style; its instance model is most similar to the instance-based one.
- **Development environment:** supports a point-and-click UI that allows advanced users/programmers to annotate pages and web users to extract and merge data via copy-and-paste; Mash Maker requires the installation of a dedicated plug-in that extends the browser with mashup features.
- **Runtime environment:** mashups are stored on the client PC, and mashups are executed inside the web browser; execution however demands for the installation of the Mash Maker browser plug-in; although it seems unlikely that large numbers of sources will be mashed up, Mash Maker should be able to scale adequately.

IBM QEDWiki

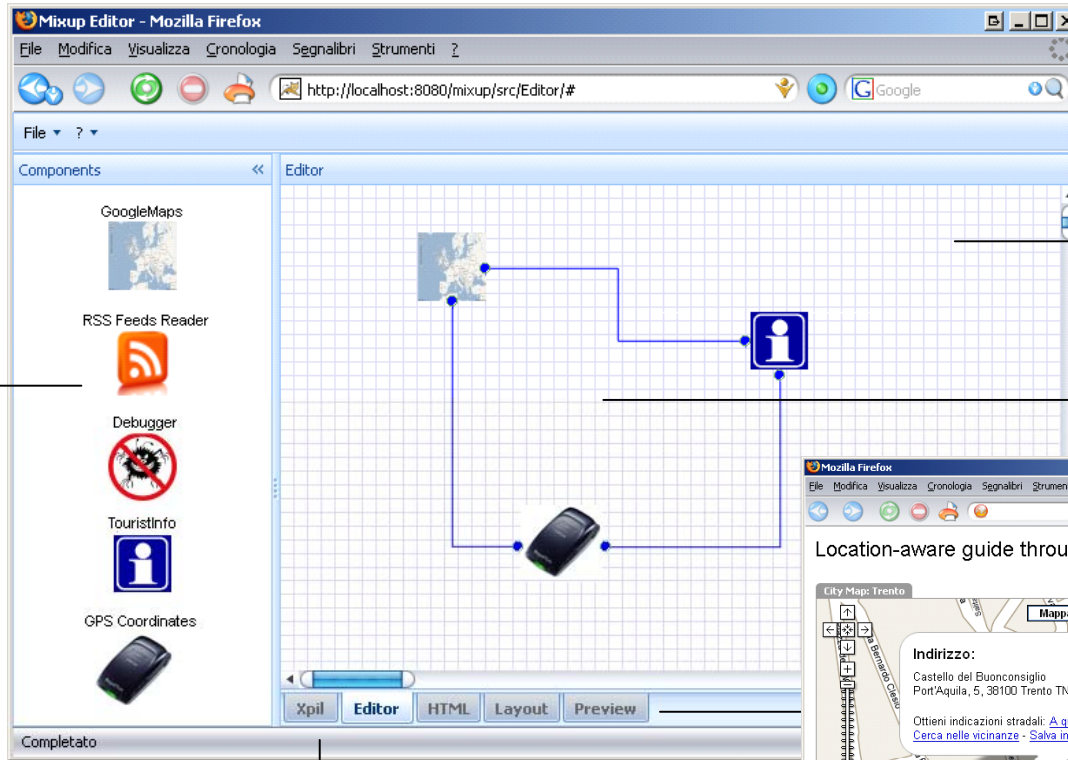
- **Component model:** mainly focuses on UI components (DA and AL components are supported via the Mashup Hub); components are equipped with a JavaScript interface; components can be extended.
- **Composition model:** produces UI output; it proposes a layout-based orchestration style; data among components is passed in a blackboard fashion; widgets are continuous.
- **Development environment:** comes with an easy-to-use drag-and-drop interface for advanced users, where components are immediately visualized; it runs in a standard web browser and does not require any plug-ins.
- **Runtime environment:** pages are hosted on an IBM server; mashups are executed mostly on the client-side; a standard web browser is able to execute QEDWiki pages; the wiki engine might encounter difficulties with a large amount of sources to integrate.

Mashup research at UNITN

- **UI integration**
 - Stand-alone web apps as **UI components**
 - **Synchronization** among components
- **Universal integration**
 - UI, application logic, and data components
 - One **component model**: abstract components, highlight similarities
 - One **composition model**: one formalism for synchronization and orchestration
 - **Hosted** development and execution

UI integration: visual editor

List of **application components** available for the mashup. Additional components may easily be **loaded** into the editor by referencing the respective online resource.



Mahup logic modeling canvas.

Graphical model of the composition logic.

Tabs that allow the designer to switch between different views (e.g. composition logic vs. layout) on the composite application under development.

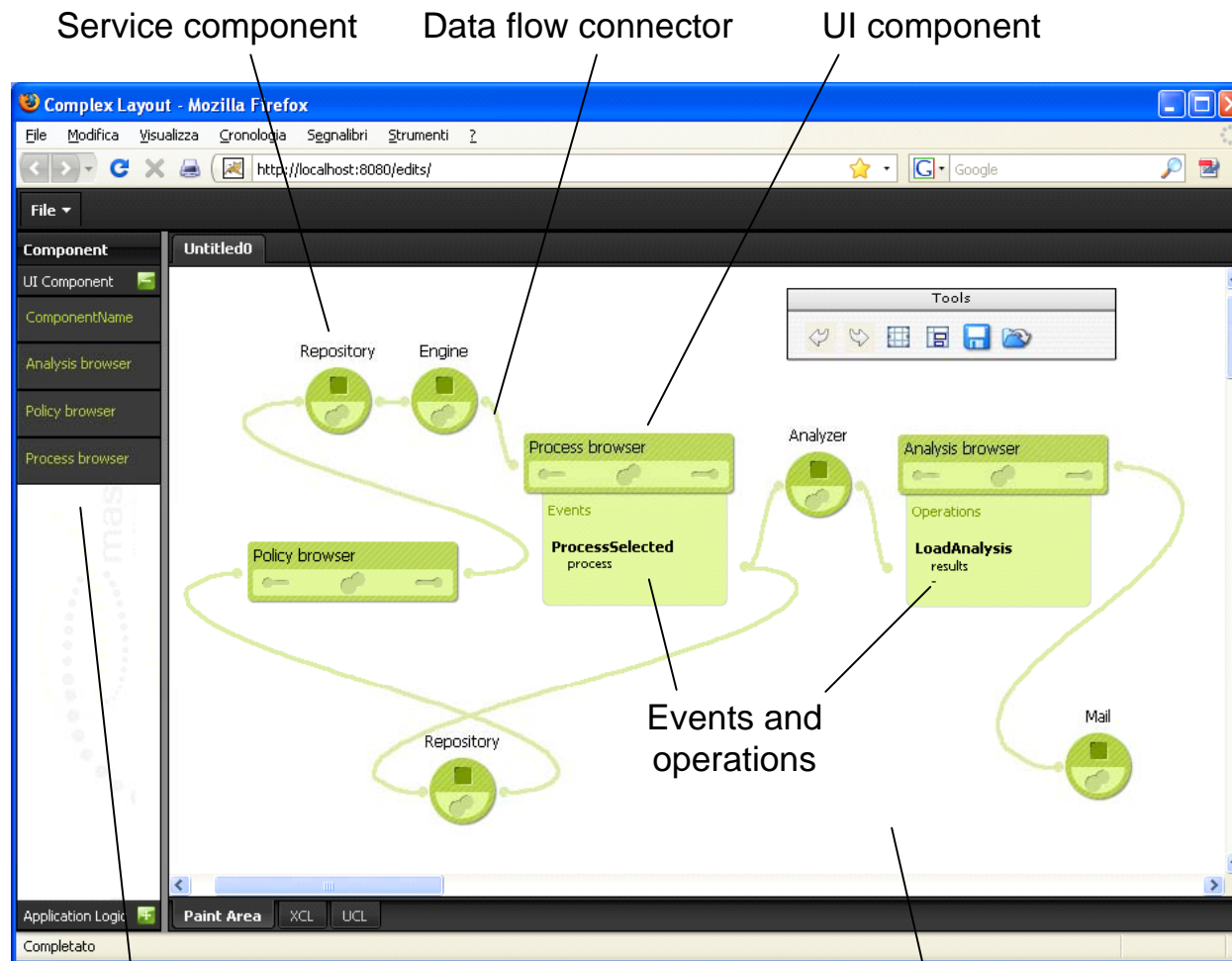
Deployment

The mashup application running in a standard web browser



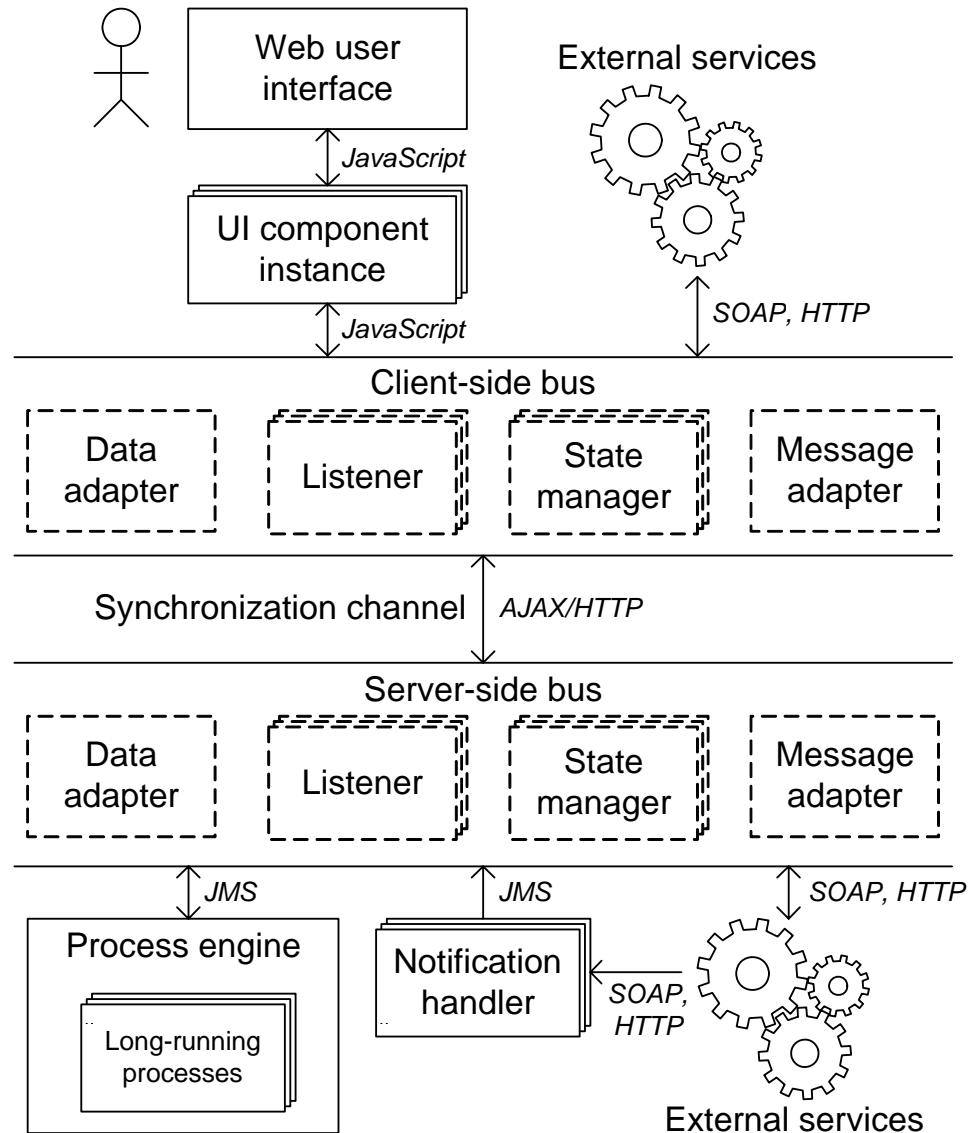
Universal integration

- The mashArt editor for universal compositions:



Hosted execution environment

- A distributed architecture:



Hosted execution environment

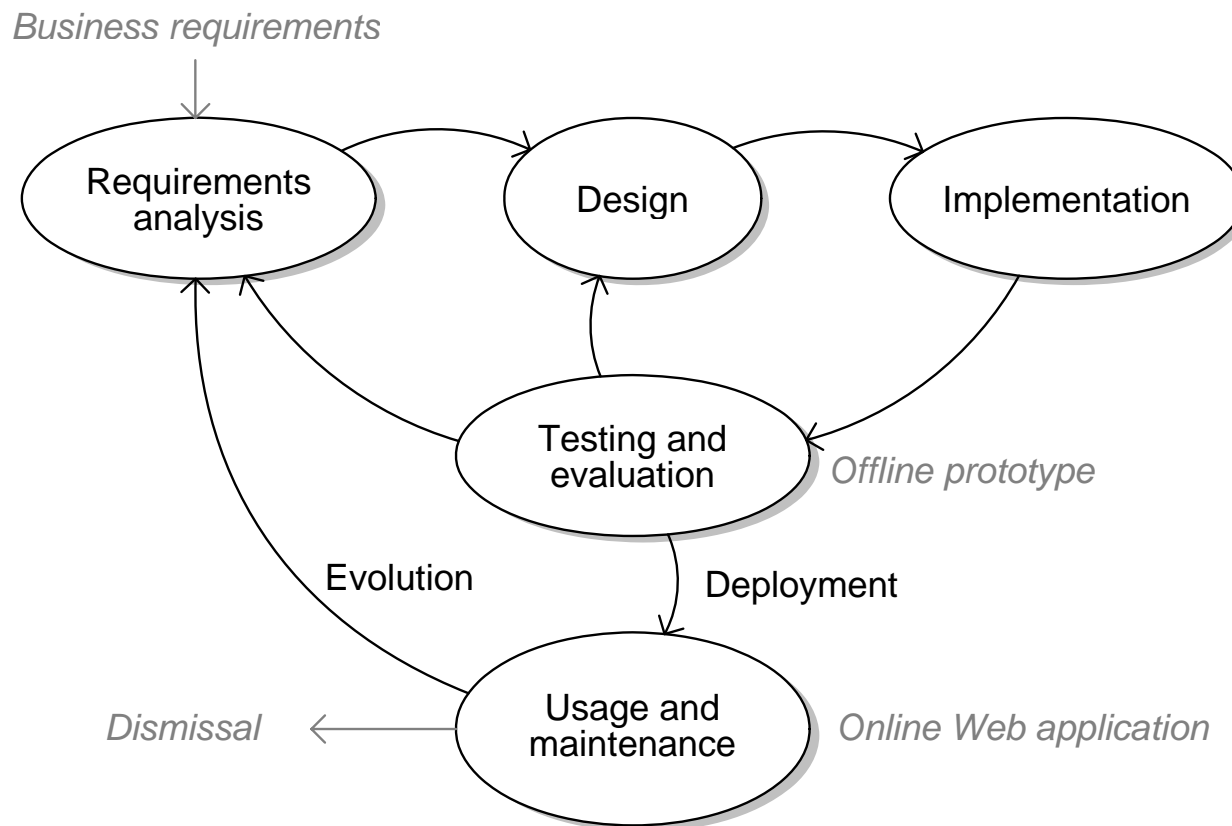
- Development challenges:
 - Seamless integration of **stateful** and **stateless** components and of **UI** and **service** components
 - **Short-living** and **long-running** process logics in the same environment
 - **Distribution** of execution tasks over client and server
 - Transparent handling of multiple **communication protocols**

A new development paradigm?

- Characteristics of **modern** web applications
 - **Fast** development cycles (Internet time)
 - **Incremental** development (prototype-based)
 - Continuous online **evolution**
- The software life cycle of modern web applications is **no longer** captured by traditional life cycle models (e.g., the spiral or the waterfall model)
- And what about user-driven composition of web applications and **mashups**?

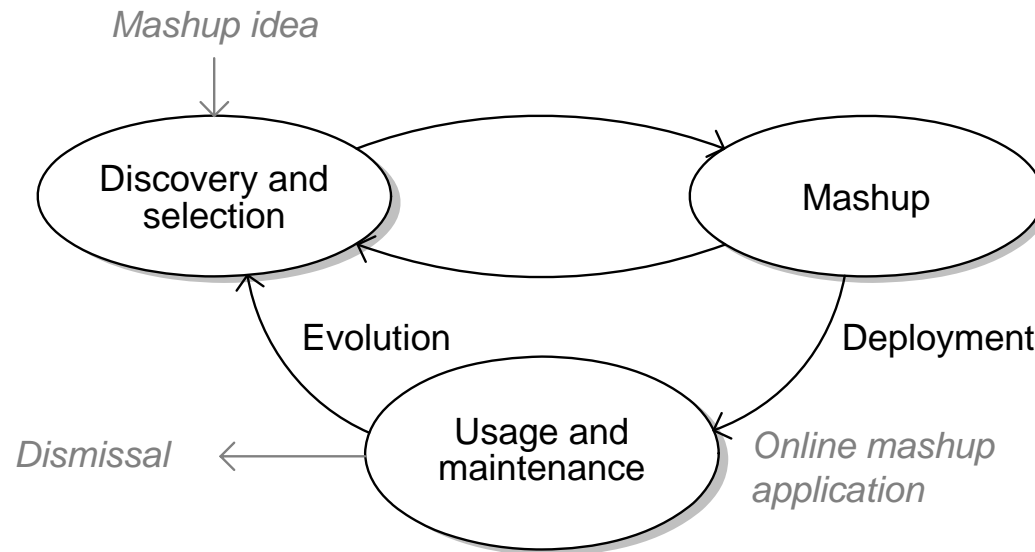
The evolution life cycle model

- A model for modern web applications (e.g., Google):



The mashup life cycle model

- A model for the **mashups to come**:



- End users doing “development”: this indeed will be a **paradigm shift** in web development!

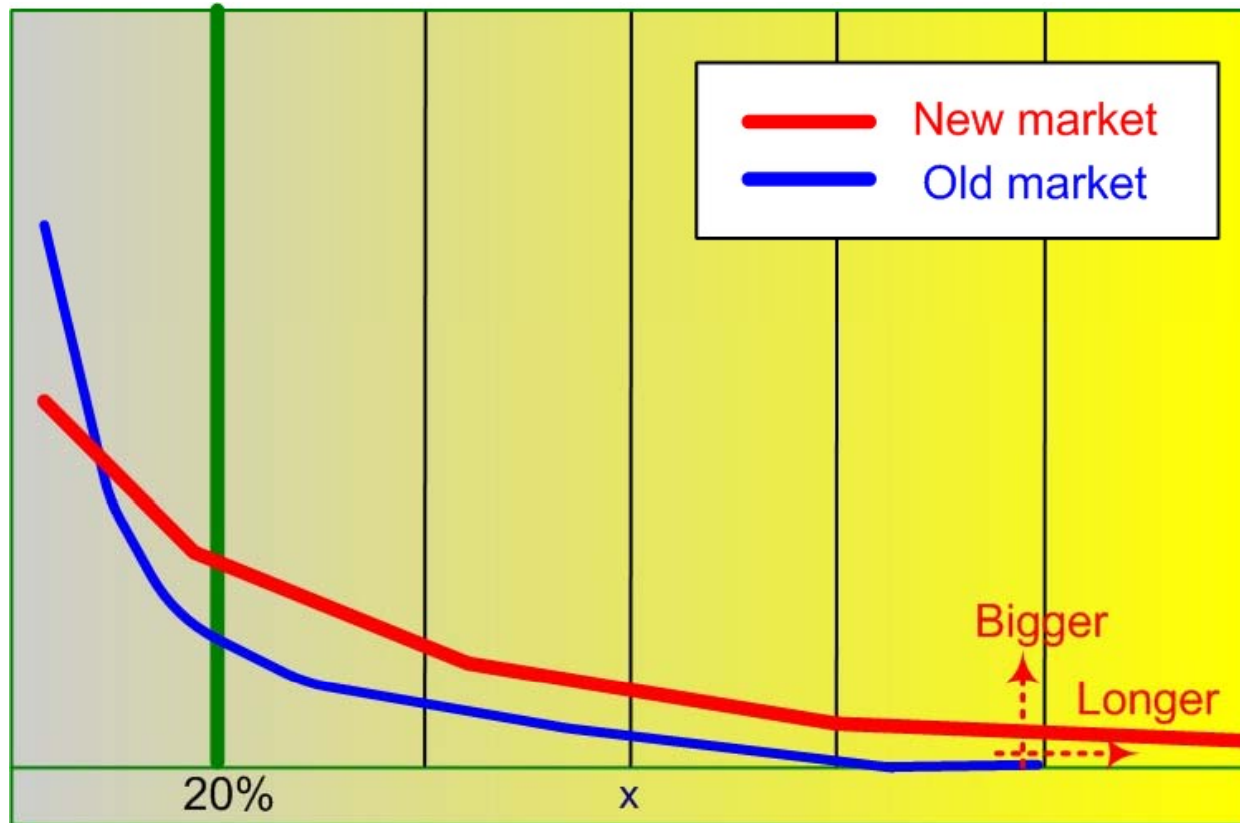
Applicability of mashups

- But what about the **utility** of mashup applications?
 - Remember the examples...
 - Do we need such kind of applications?
- Mashups as **situational applications**
 - What is now still looking like a hobby of a limited group of tech-savvy freaks...
 - ...might be the future of **business** applications, as envisioned by Jhingran [Enterprise Information Mashups: Integrating Information, Simply. *VLDB'06*, pp. 3-4]
 - **Situational application**: a short-living application for a limited audience and with limited but specific capabilities

Applicability of mashups

- Perfectly **understanding customer needs**, in order to customize software and satisfy as much users as possible, is costly – if not impossible
- Only very few innovations are really **breakthroughs**, most innovations only create little value
- Mashups may leverage “**user innovation**”:
 - Users themselves know best what they want
 - Mashups enable them to build their own applications

The long tail of the SW market



Research issues to be addressed

- **Reusable components**
 - Expressive component models, description languages and discovery and selection facilities
- **Composition languages**
 - Simple yet expressive execution languages, plug-in style development
- **Composition tools**
 - Composition languages equipped with graphical modeling formalisms, computer-aided development
- **Execution platforms**
 - Execution support in form of interpreters or parsers, hosted solutions
- **Interoperability**
 - Components with cross-platform reusability, mashup-specific standards

Conclusion

- Mashups: a new way of data, application, and UI **integration on the Web**
- They're **end-user-driven**
 - People just mash up things, whether we support them or not
 - So, let's help them in their effort!
- There is a growing interest in mashups especially from **business community**
 - Just think about who is investing most money: Yahoo, Google, Microsoft, Intel, IBM,...