Lively Mashups for Mobile Devices

Feetu Nyrhinen, Arto Salminen, Tommi Mikkonen

Tampere University of Technology

Antero Taivalsaari

Sun Microsystems Laboratories



Outline

- Background
- Mashups
- Mashup Development and Tools
- Lively Mashups
- Qt as a Mashup Platform
- Mashup demos
- Experiences
- Conclusions



Background

- Web as the platform
 - End-user software is moving to the Web.
 - Typical examples: project management, calendars, document management, instant messaging, social networking, ...
 - Web browser acts as a replacement for the conventional OS.
- Mobile devices are becoming web-enabled, but there still are constraints such as smaller screen size, battery consumption, lower CPU speed and network bandwidth.



Mashups

- Mashup: A web site that combines content from more than one source (multiple web sites) into an integrated experience.
- Mashups leverage the power of the Web to support worldwide sharing of content that would not have been easily accessible or reusable before the Web.
- In principle, the content to be combined can be anything (text, source code, maps, video, blogs, product reviews, price data, ...) as long as it can be meaningfully combined with other content.
- See, e.g., http://woozor.us/ (Weather conditions on Google Map)



Mashup Development and Tools

- There is a plethora of various tools for the mashup development.
- However, general tools are still fairly limited in functionality and many of those are far from finished applications.
- Some common trends:
 - Using the web not only for executing applications but also for developing them.
 - Visual programming techniques.
 - The web server is used to host and share mashups.
 - Direct connections to existing web services.
- Mashup development for mobile devices is still a field tampere university of with big challenges.

Lively Mashups

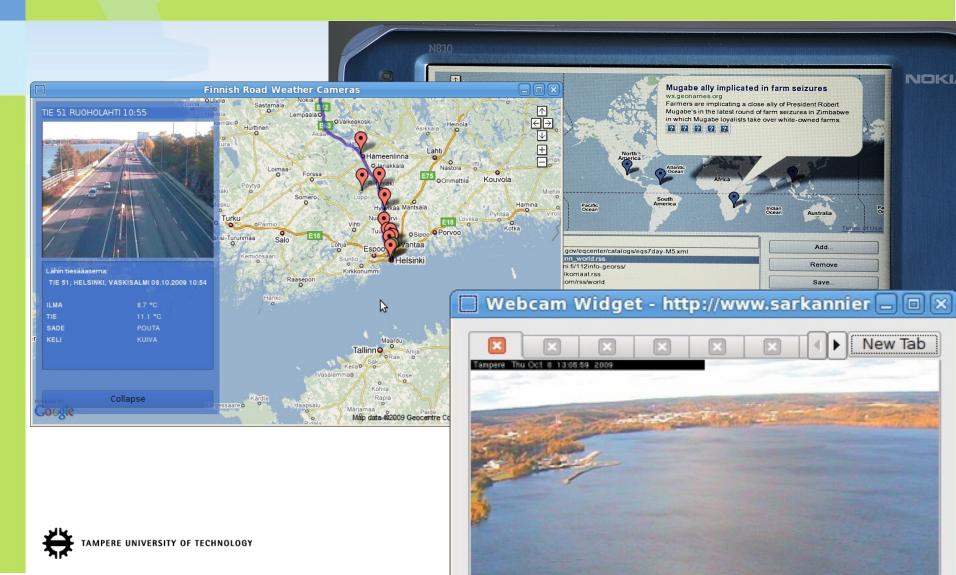
- Our target was to develop practical cross-platform web mashups with a special emphasis on mobile devices.
- JavaScript, AJAX and JavaScript bindings of Qt used as central building blocks.
- Qt (http://qt.nokia.com/) is a mature, rich, crossplatform application framework owned by Nokia.
- Qt offers a comprehensive set of APIs for graphics, networking, widgets, layouts, web, etc.
- Our work is partly based on a parallel project, *Lively for Qt* (http://lively.cs.tut.fi/qt) development environment enabling development of applications



Ot as Mashup Platform

- Qt libraries include for instance a complete web browser based on WebKit (http://webkit.org/) browser engine and DOM and XML APIs for parsing, manipulation and generation of new web content.
- Qt makes it possible to use JavaScript both inside the browser component and in QtScript that includes an ECMAScript capable JavaScript engine.
- Tools such as a built-in debugger make the JavaScript development a less painful task.
- The Qt APIs can be made accessible in JavaScript by a tool called QtScriptGenerator.

Demos: QtWeatherCameras, QtMapNews, QtScrapbook



Experiences

- There are same kind of problems as in the web development in general, i.e. usability, compatibility and security.
- Especially the following two areas are critical regarding the mashup development:
 - Lack of well-defined interfaces
 - Security-related issues
- Qt APIs made many tasks in the development less painful: XML parsing, networking, debugging etc.



Mobile Mashups

- Mobile mashups and their specialties in comparison to desktop mashups:
 - Usability issues
 - small screen size, font differences, touch screen etc.
 - Connectivity issues
 - network connection may not be always at hand
 - → provide feedback to the user when problems are encountered
 - Performance issues
 - enhanced JavaScript performance highly desirable (high performance JS engines are needed)
 - Computation on a mobile client



Conclusions

- Developing mashups based on Qt and JavaScript has been an interesting task.
- Qt APIs make the development easier.
- Fields for future work:
 - Location-aware applications making use of specialties of mobile devices, e.g. GPS or various sensors.
 - Collaborative mashups enabling real-time collaboration.
- Mashups open new opportunities, but suffer from same problems as the traditional web development and a set of others (instability, strong dependency of other services).
- Performance, security and lack of well-defined interfaces remain central problems.



Thank You! Questions?

