20. Situational Applications and Mashups

5 November 2008
Bob Glushko

Plan for Today's Lecture

Platforms for Composite Applications
Mash-ups
Mash-ups {and,or,vs} Composite Applications
A Vision: Rapid Service Discovery and Composition

An "agile," "adaptive," "on demand" [insert other buzzwords here] enterprise needs to be able to quickly and cost-effectively change how it does business and who it does business with (suppliers, business partners, customers), etc.

The IT organization must respond by extending existing applications, building new custom applications, or licensing and implementing new packaged applications -- but these take time.

Similarly, in "crisis management" or "emergency response" contexts novel configurations of national, state, and local government agencies need to consolidate and exchange information to coordinate their response.

So there is a clear need to be able to rapidly discover and combine multiple services and information resources.

Requirements for Efficient Rapid Composition

Do I know what services I'm looking for?

Do I know where to look? Can I find the service or information resource I'm looking for?

Are the services compatible from a business perspective?

Is the service technically compatible? (technical interface compatibility)

Can I transform the service inputs or outputs to achieve compatibility?

Do the services meet my quality and performance requirements?
The Reality -- Hand-Crafted Composition

Most services and information resources aren't listed in authoritative public registries

Service descriptions are often limited to technical interfaces, lacking information needed to assess business model, business process, and QoS compatibility

Emerging platforms for composite applications are limited in the kinds of document interfaces they can handle (e.g., optimized for services operating on relational databases) or still require some "programming" expertise

(Some platforms support reuse of integration semantics, but they are not sufficiently grounded in ontologies or reference models to enable automated mapping and transformation)

Sun Java Composite Application Platform
Sun Java CAPS Description


Java CAPS combines the functionality of existing legacy and packaged applications, as well as recently developed, reusable services, in an intuitive composite application integration infrastructure.

.. a proven set of end-to-end graphical tools that deliver optimized source code for standards-based collaboration and business process execution...

Developers may operate on the graphical model or directly on the source code which is synchronized with the graphical model.

Webcast about Java CAPS
(http://webcast-west.sun.com/interactive/08D01561/index.html)
**XFY Description**

xfy is a powerful document-based composite application framework...

that allows organizations to rapidly unify content and data in a single dynamic document.

With no technical knowledge, business users can rapidly assemble rich document-based applications that bring together isolated content and data from disparate sources.

xfy dynamic documents are, in essence, applications

**Lowering the Programming Threshold**

Composite applications platforms reduce the amount of "programming" expertise needed to design and assemble an application, but don't eliminate it.

Nevertheless, if 12 million people say they "program in the workplace" but only 3 million have that as their job, then people who aren't programmers are doing something they call programming (from "Changing the corporate IT development model")

Some of this might actually be programming, but most of it is something else, especially "programming" using spreadsheets

How low can the threshold go?
Nonprogrammer Programmers

Incorporates existing functionality
Not a formal project
Very context specific; intentionally "brittle" in focus
Informal software engineering
Low levels of programming expertise
Is Using Google "MyMaps" Programming?

Google MyMaps
- plot your data on a map without having to use Google Map API

Defining "Mashup"

A “mashup” is a lightweight tactical integration of multi-sourced applications or content into a single offering (Gartner)

Web application hybrid that combines data from more than one source into a single integrated tool (Wikipedia)

Mashups are loosely coupled distributed systems... to the extreme. The developers of the individual components do not know each other and possibly do not even know that their application is being used as a component by another application (Wilde)

Data flows that take information originating from one source (or user input), applying web services to augment or transform the information, and then visualizing the results
This is Not a New Idea

The original and still most popular mashup tool is the spreadsheet... which made the PC happen in the 1980s.

Musical remixing and sampling has a long history, and a music mash-up is a song created out of pieces of two or more songs, usually by overlaying the vocal track of one song seamlessly over the music track of another.

Roots in "Portal" Applications

Yahoo! created a style of end-user tailorable application in which the user customizes the content and layout of the Yahoo! Home page.

Each piece or channel of content is contained in a “portlet” (see OASIS Web Services for Remote Portlets Specification (http://docs.oasis-open.org/wsrp/v2/wsrp-2.0-spec.html))

In portals, content aggregation takes place on the server, and in the UI is “side by side” aggregation.

In mashups, content aggregation takes place on the client side in “melting pot” style.
Mashup Motivation: Portal Customization

Roots in E-commerce Catalog Integration and Price Comparison

Dumb "screen scrapers" used for price comparison in the early days of web commerce (e.g., bizrate 1998)

"Smart" approaches to "unifying heterogeneous information models" also in use at same time (e.g., Tesserae) http://doi.acm.org/10.1145/274946.274954
So Why The Hype About Mashups NOW?

The amount of information available on the Web as XML "content" (as opposed to HTML "presentation information") has been growing rapidly.

Much of this information is now accessible via web services, as well as through lower-effort programming models like RSS and ATOM feeds.

New "mashup making" tools continue to lower the programming threshold.

A surprising amount of tolerance and even encouragement by service and information providers to allow the reuse of their content.
Minimal Mashup Architecture

Map Mashups

Most common type of mashup

Emerged because Google (and later the web-based map apps) offered access to "geographical information" in a vastly more accessible way than previous Map/GIS

Hard to believe that map mashups are only 3 years old, and that there was some doubt that Google would even allow them!

Not entirely clear when map is "visualization widget" for database content and when it is really a mashup
Map Mashup Examples

Housing Maps (http://www.housingmaps.com/) - mash of craigslist and google
Oakland crimespotting (http://oakland.crimespotting.org)
Geocoding (http://mapsapi.googlepages.com/batchgeo.htm)

Categorizing Mashups

By targeted user (for yourself only, anyone, someone in your business)
By content type (map, photo, video, news)
By primary function (search, shop, locate)
5 Types of APIs from "the Big 5" Firms

Mashup Design Dimensions

Search capability?

Static, or tracking changes/monitoring?

Browser-based, or widget/gadget based?

Personalizable? Is the mashup designed for content creation or just exploration/browsing (can user enter information or just make selections/filter on other data)

Based on an existing folksonomy or tag set?
Programmable Web (3469 as of 3 Nov 2008)

"What do we mashup" (Hong and Wong)
http://doi.acm.org/10.1145/1370847.1370855

Greasemonkey

Somewhat similar in purpose to mashups and portal platforms are
"Greasemonkey" scripts (http://userscripts.org/)

Greasemonkey is a Mozilla Firefox extension that installs scripts that make
on-the-fly changes to most HTML-based web pages

These changes made to the web pages are executed every time the page is
opened

Typical Greasemonkey functions:
  - Auto-filling of forms
  - Text highlighting and other format alteration
  - Removing specific content
  - Add content
  - Add links, buttons, or any other HTML element
Mashups vs Composite Applications

User expectations about the "-ilities"
Depth and robustness of integration
The "pragmatic - experiential" continuum

Dimensions of User Expectations

Data completeness and quality
Performance and QoS
Scaleability
Security, access control
The Integration Continuum

instance-based (e.g., from scraping)
datatype-based
schema-based
model-based

Pragmatic - Experiential Continuum
Mashups as Vehicles for "Community Creation"

Where does the application fall on the continuum between pragmatic and experiential activity?

Is the mashup a way to get work done, or to create a community?

Mashups might (intentionally or unintentionally) violate social norms about information disclosure or "what should be talked about"

(E. Goodman & A. Moed, "Community in mashups; The case of personal geodata?")

The Changing Role of Corporate IT

"from solution developer to solution enabler"

the emergence of "situational applications" introduces complexity in monitoring, event analysis, patch management, troubleshooting, and other systems management tasks
IBM's Situational Applications Environment

IBM "Mashup Center"

The IBM "SAE" - an internal development environment - apparently has been productized as IBM "Mashup Center"

Capabilities: Unlock information sources - create easily consumable feeds from a variety of information sources inside and outside your enterprise.

Rapid assembly of mashups via "drag and drop"

Discover and share assets in a catalog

Combine feeds and transform information

Create dynamic widgets - includes a powerful widget creation environment that enables developers of all skill levels to rapidly generate widgets without coding
Reading for November 10