



# The Programmable Web

## Agile, Social, and Grassroots Computing

IEEE International Conference on Semantic Computing special session on  
The World of Semantics beyond the Semantic Web

**E. Michael Maximilien**

IBM Almaden Research Center  
San Jose, CA USA

**Ajith Ranabahu**

Wright State University  
Dayton, OH USA

**v0.9.3**

September 14, 2007

getting  Agile@IBM

Irvine, CA  
September 17-29, 2007





# Agenda

- Efforts to achieve a *programmable Web*
  - ▶ Web services
  - ▶ Semantic Web
  - ▶ Web 2.0
- Framing the problem
- Why Web 2.0 is the correct approach
  - ▶ Agile and social computing
  - ▶ Grassroots computing
- Summary and parting thoughts





# Programmable Web?

- The Web as a substrate for
  - ▶ **applications**
  - ▶ business **communication**
  - ▶ business **integration**
- Various **challenges**
  - ▶ What **protocols**?
  - ▶ What **format** for messages? **interactions**?
  - ▶ What (if any) a priory **agreements**?
- **Three** separate approaches





# Web services

- **Primary goals** and premises
  - ▶ Web as a substrate for **enterprise IT**
  - ▶ **XML** as data interchange format
  - ▶ **protocol agnostic**
  - ▶ **directories** for discovery and integration
- Various **industry** and **academia** backers
- Tried to deal with **nonfunctional** aspects
- Led to **WS-\*** proposed set of standards
- **Vendor support** via tooling, training, ...





# Complexity hell!

- So many proposals, led to **WS-Nothing**
- **WS-\*** “standards”
  - ▶ **fragmented** (early SOAP implementations)
  - ▶ **competing** (on some conceptual aspects)
  - ▶ too early and **unproven** (e.g., UDDI)
  - ▶ **complex** which leads to need for tooling
- Very difficult to use all standards
- **No simple nor nimble** solution





# Source of complexity?

- WS-\* creator have lots of experience in distributed computing and the Web
- What are some **causes** of **complexity**?
  - ▶ Web protocols, standards are **minimal**
  - ▶ Forgot about the Web's main **purpose**
    - ✓ **connecting** people
    - ✓ **lightweight** (HTML) and **inclusive**
  - ▶ **Nonfunctional** properties support
- **Evolution** by **committee**
- Forgot the **roots** of the Web





# Semantic Web

- Primary **goals**
  - ▶ **explicate meaning** of Web's heterogeneous content, resources, and services
  - ▶ create software **agents** to **reason** about meaning and do useful work
- **Roots** in **AI** subfields
  1. description logics
  2. knowledge representation and engineering
- Backed by **academia and research community**
- **Limited** and domain-specific **successes**





# Broken promises

- Web's **strength** is arguably from
  - ▶ **heterogeneity**
  - ▶ **simple** to use and implement (HTTP protocol)
  - ▶ **freedom** and no centralized authority
- Semantic Web's **misguided assumptions**
  - ▶ **consensus building** is hard
  - ▶ return on investment (**ROI**) is **small**
- No **incentives** to do big design up front (BDUF)
- Need for **heavy tooling**





# Web 2.0

- **Social** and **participation** Web
- Programmable Web 2.0
  - ▶ **Informational** services (mostly data-based)
  - ▶ Use **HTTP protocol** as intended
  - ▶ **Lightweight** (not much tooling required)
  - ▶ **Remixing** of application logic and data
  - ▶ **Informal, grassroots, and *situational***
- ***Mashups***





# Success factors

- What is the **source** of **success**?
  - ▶ **simplicity**
  - ▶ **low barriers** to entry
  - ▶ people are **contributing**
- **Evidence** is based on vast number of startups
- Web **APIs** directories (e.g., ProgrammableWeb) see more than 80+% of APIs submissions as non-SOAP
- **Enterprises** are also seeing benefits



# Framing the problem

- Build programmable Web on basic Web standards
- Ease of **programming** is key (e.g., JSON)
- Facilitate **collaboration**
- Language and framework **agnostic**
- **Agility**
- **Seven fallacies** of distributed systems (P. Deutsch)
  - ▶ network is reliable, zero latency
  - ▶ network is secure, topology is constant, ...





# Web 2.0 is social

- The Web is about **connecting** people
- Web 2.0 make people **contributors**
- The **usage** of data, content, and services **is as important** than it's meaning and intent
  - ▶ **idiosyncratic** usages lead to innovations
  - ▶ adds **unintended** (but **useful**) usages and meanings aka **mashups**
  - ▶ **pragmatism** over global design
- Heterogeneity can be partially solved using **human collaboration assisted** with mining techniques





# Web 2.0 and agility

- Agile methods and practices are about **embracing changes**
  - ▶ **iterative** and **evolutionary** approach to system design and development
  - ▶ *minimize* **assumptions** and **ceremonies**
  - ▶ *minimize* **BDUF**
  - ▶ *minimize* **non-executable** artifacts
  - ▶ **accept limits** and **context** vs. fighting them
- The **Web 2.0** is **prototypical large agile system** (constant change)





# Web 2.0 is grassroots

- **Idiosyncratic** groups of people contributing and forming communities, e.g., Digg.com, Slashdot.org
- **Communities** mean
  - ▶ common intent and purposes
  - ▶ context and domain is restricted
  - ▶ semantics evolve with people's needs and usages
- **Pragmatic** evolution of approaches, e.g., JSON for data representation (vs. XML)
- **Open source software** governance model





# Summary

- **Programmable Web** is very much **desirable**
- **Three** separate efforts
  1. Web services is “complete” but complex, not agile, and difficult to use
  2. Semantic Web strayed from Web realities
  3. Web 2.0 gaining in popularity
- Basic issues in distributed systems need attention
- Programmable Web must remain true to the Web’s key success factors (simplicity and human connection) and be **agile, social, and grassroots**





# Backup





# Parting thoughts

- **Security** is still of primary importance
- **Nonfunctional** guarantees **not always** needed
- **Human interface** implies certain performance needs
- **BDUF** not always needed
- **Agility** is paramount
- **Grassroots** means sharing and **evolving** with usages vs. global enforcements or standards





# References

- E. M. Maximilien, A. Ranabahu, and S. Tai “Swashup: Situational Web Applications Mashups”, OOPSLA 2007 poster, *to appear*
- E. M. Maximilien, H. Wilkinson, N. Desai, and S. Tai “A Domain-Specific Language for Web Services and APIs Mashups”, ICSSOC 2007 paper, *to appear*
- E. M. Maximilien “A Partial Solution to the SWS Challenge: The Ruby on Rails Approach”, ICIES 2007 Workshop