



Software Tooling for Web 2.0, Web 3.0, ULS

Rinaldo Di Giorgio
Senior Staff Engineer Sun Microsystems
Doctoral Student at Pace University(DPS)

Present

What will the Net look like?

- Present
 - internal versus external, 2 tier, 3 tier, some SOA
- Ultra-Large-Scale Systems ULS
 - new systems larger, dependencies, new development
- Web 2.0 (R/W)
 - Software becomes a service with a community, being developed the old way
- Web 3.0 (R/W/E)
 - Interactions are complex transactions over many servers

Present

- Do most companies Build and Deploy on a fixed release train
 - Internal Applications
 - External Applications
 - Use IDES
 - Some put a bunch of tools into the IDE
 - Use third Party tools

Web 1.0 2.0 and 3.0, Stephen Baker

- I've always thought of Web 1.0/2.0/3.0 in parallel to Tim Berners-Lee's notion of the read-write web, which is often used as an alternative way of describing Web 2.0.
- First, we had Web 1.0 - the read-only web. Then came Web 2.0 - the read-write web - all of these services that make it easy for us to contribute content and interact with others. If you keep up the programming analogy, the next phase would be Web 3.0 - the Read-Write-Execute Web.

ULS (DOD view)

- Orders of magnitude larger than today's systems, and systems of systems
 - LOC, Number of users, Data in the system, number of connections, interdependencies software/hardware, People will be users and parts of the system
 - New demands
 - software acquisition, production, deployment, management, documentation, usage and evolution practices
 - New research
 - HCI, Computational Emergence, Design, Computational Engineering, Adaptive System Infrastructure, Adaptable System Quality, (System health, GDP), Policy Acquisition and Management

Web 2.0

- For every AJAX or Ruby idea you may have 20 or 30 others probably have one too (Security/Performance)
- Mechanics for these apps are understood
- Server and network load will increase
 - See Blog on Verizon, Disney and CNN
 - Consider 10,000,000 people demanding 2.0 during special events
 - <http://digiorgio.com:8090/blojsom/blog/Rinaldo/Talks%2C+presentations+of+interest/2006/12/03/Amazon-Ebay-CNN-and-Etrade-Observations-On-Dynamic-Properties-of-respective-web-sites.html>

Web 2.0 (1 of 2) Apps

- DoubleClick
- Ofoto
- Akamai
- mp3.com
- Britannica Online
- Personal Websites
- Evite
- Google AdSense
- Flickr
- BitTorrent
- Napster
- Wikipedia
- Blogging
- upcming.org/EVDB

Web 2.0 apps (2 of 2) [1]

- Domain Name Speculation
- page views
- screen scraping
- publishing
- content Management Systems
- directories(taxonomy)
- stickiness
- search engine optimization
- cost per click
- Web Services
- Participation
- Wikis
- tagging(folksonomy)
- syndication

Web 3.0

- API Services
 - Everyone will be able to post an API
- Aggregation Services
 - RSS Aggregators, StrikIron
- Application Services
 - Allow users to use multiple services
- Clients
 - Two types

Software Tools

- What's more important the Tools or the Language, Framework?
 - Java, Ruby --> JSF/J2EE, Rails
- An Unsolved Problem – Literate Programming
 - Code Documentation
- Hackystat - Telemetry based Environments
- New Tools
 - Ready to Go Stacks -ROR, JackPot, Mercurial
- Survey

Tools versus Language

- Modern systems are not just about the language
 - A Modern developer must manage a complex tool set
 - Interested parties have poor visibility into all the tools
 - AccuRev
- What can we do to combine the languages, and the tools so that everyone can participate in the process
 - Look at an existing attempt Hackystat
 - Challenge will be the third party vendor specific tools

Literate Programming(1 of 3)

- The philosophy behind CWEB is that an experienced system programmer, who wants to provide the best possible documentation of his or her software products, needs two things simultaneously: a language like TeX for formatting, and a language like C for programming. Neither type of language can provide the best documentation by itself; but when both are appropriately combined, we obtain a system that is much more useful than either language separately.

Literate Programming(2 of 3)

- Besides providing a documentation tool, CWEB enhances the C language by providing the ability to permute pieces of the program text, so that a large system can be understood entirely in terms of small sections and their local interrelationships. The CTANGLE program is so named because it takes a given web and moves the sections from their web structure into the order required by C; the advantage of programming in CWEB is that the algorithms can be expressed in "untangled" form, with each section explained separately. The CWEAVE program is so named because it takes a given web and intertwines the TeX and C portions contained in each section, then it knits the whole fabric into a structured document.

Literate Programming(3 of 3)

- The structure of a software program may be thought of as a "WEB" that is made up of many interconnected pieces. To document such a program we want to explain each individual part of the web and how it relates to its neighbors. The typographic tools provided by TeX give us an opportunity to explain the local structure of each part by making that structure visible, and the programming tools provided by languages like C make it possible for us to specify the algorithms formally and unambiguously. By combining the two, we can develop a style of programming that maximizes our ability to perceive the structure of a complex piece of software, and at the same time the documented programs can be mechanically translated into a working software system that matches the documentation.

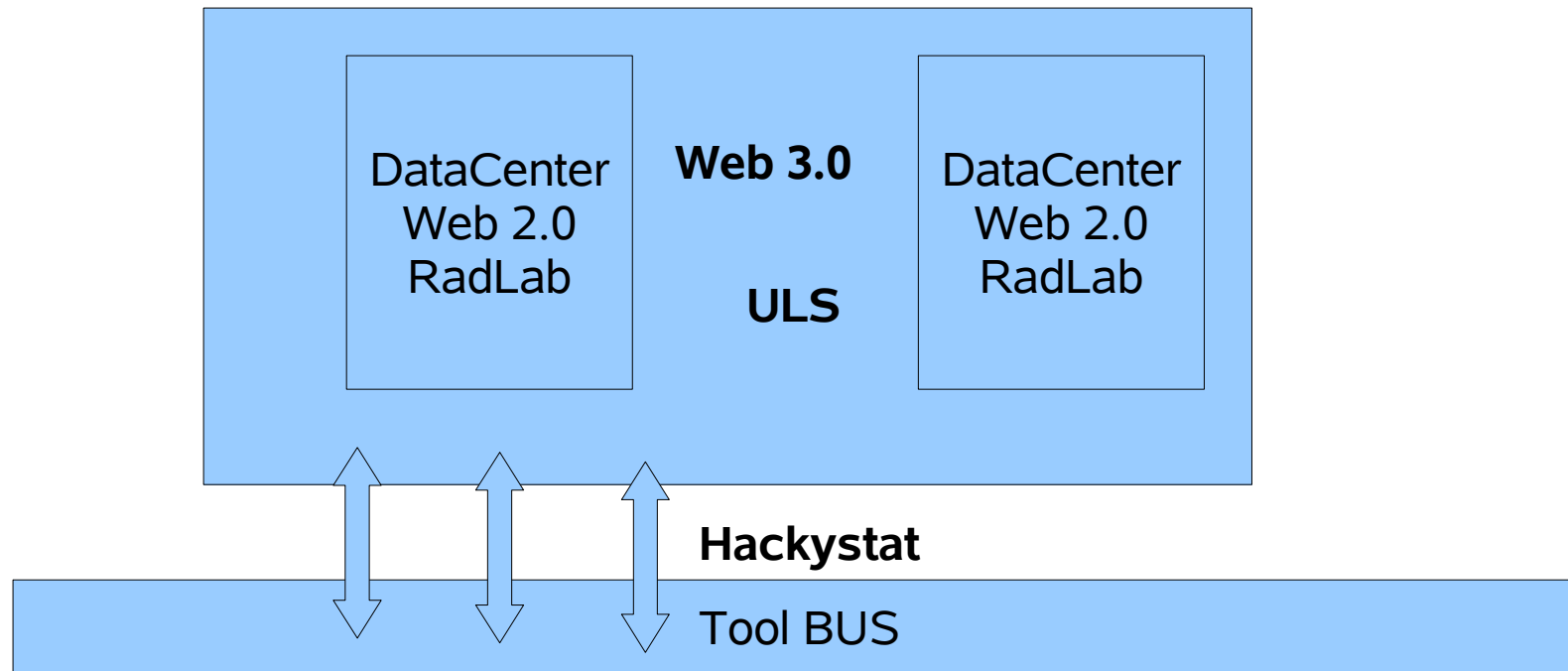
Hackystat

- Sensors that gather process and product data
 - Ant, BCML, CCCC, Checkstyle, CLI, CppUnit, CVS, DependencyFinder, Eclipse, Emacs, Emma, FindBugs, JBlanket, Jira, JUnit, Jupiter, LoadTest, LOCC, Office, PMD, SCLC, SVN, Vim, VisualStudio
- Sensor data types
 - Activity, BuffTrans, Build, CLI, CodeIssue, Commit, Coverage, Dependency, FileMetric, Issue, Perf, ReviewActivity, ReviewIssue, UnitTest
- Applications
 - Personal metrics, Project metrics, Cross-Project metrics, Workflow, Software Telemetry

[Application|Program|System]Pedia

- All phases of software production are captured sometimes with effort, other times silently. All stakeholders have different views into the system looking at one consistent data store.
- A customer reporting a bug can trace all the way back to the requirements and see activity around the code, tests, deploys and how it was delivered to him securely with performance constraints.

ApplicationPedia



- 1) Developers
- 2) Program Managers
- 3) Marketing
- 4) Security
- 5) Operations
- 6) Networking
- 7) Customers

- 1) IDEs
- 2) Maven
- 3) Ant
- 4) Third party tools
- 5) Cruisecontrol
- 6) Code Reviews
- 7) BugTracking

Developments needed

- Tools that communicate with each other
 - may require ontological descriptions and use of DAML
- Improvements in languages to facilitate documentation
 - Annotations or new languages
- Role of CMMI, Agile
- How to store the data and relationships

Survey

- Available online here
- Purpose is to determine state of tools
 - What is in use?
 - What would be useful?
 - Can you handle more tools?
 - Are tools complicated and hard to use?

References

- [1] What is Web 2.0, Daniel Lewis
- [2] What to expect from Web 3.0, Phil Wainewright
- <http://digiorgio.com:8097/tiki-index.php?page=Tools&bl>
- http://digiorgio.com:8097/tiki-take_survey.php?surveyId=1