Flow Webs: Architecture and Mechanism for Sensor Webs

Michael M. Gorlick  Samuel D. Gasster
Grace S. Peng  Michael McAtee

The Aerospace Corporation
El Segundo, California

gorlick@aero.org, gasster@aero.org
Why Net-centric Architectures?

- Share on demand
- Reconfigure on demand
- Compose resources on demand
- Tailor resources on demand
- Redirect to alternates
- Recovery
- Fault tolerance

Are web services the answer?
What is the Architectural Foundation of Web Services?

- REpresentational State Transfer (REST)
  - Explains web behavior and scaling
  - URLs denote abstract resources
  - Resources have multiple representations
  - Small number of basic methods (GET, HEAD, PUT, POST)
  - Web exchanges are context-free

- If SOAP is the answer what was the question?
  - RPC dressed in XML that hijacks HTTP as transport
  - Complexity (high)
  - Performance (low)
  - Encapsulation and composition (none)
Rethinking Web Services

- Make *service exchange* as natural as content exchange
- Zero complexity
- Parsing XML is a waste of resources
- Encapsulation and composition are first-order mechanisms
- Promote the presence of intermediaries (service proxies)
Computational REST (CREST)

- Exchanges of mobile code/continuations/environments is the fundamental action among CREST peers
- URLs denote computational resources
  - Scheme interpreters whose environments contain URL-specific bindings
- Requests are any Scheme program (expression) or continuation
  - Expression/continuation evaluated by URL-specific interpreter
  - Outcome of evaluation (Scheme expression, continuation or environment) is the response
- Scheme becomes the assembly language of the new web
- Justin Erenkrantz/UCI + M. Gorlick/Aerospace
CREST Web Services

- Constructed by users (not providers)
  - Users dispatch custom Scheme programs or continuations to URLs
- Custom Scheme programs are generated by higher-level tools and libraries
- Reduces web services to the common norm
- Continuations allow network:
  - iteration, recursion, mobility, generators, coroutines, exceptions, restart, replay, transactions, workflow, synchronization, ...
From CREST comes Flow Webs

- Arbitrary dynamic directed graphs of network services
  - Nodes are semi-autonomous web peers
  - Edges are flows: *streams of messages constrained in space and time*
- *Flow webs may encapsulate other flow webs*
- *Flow webs may be composed with other flow webs*
- *Flow webs may be edited and modified while executing*
  - *Flow mobility: move flows without message loss*
- *Flow webs implement systems of systems interlinked on demand to synthesize services*
Summary

- Web services are obsolete! Long live web services!
- CREST is user-directed web services
- From CREST comes flow webs for the dynamic synthesis of systems of systems
- Flow webs are the building material for future web systems

Firewatch
(NASA AIST-QRS-06-0047)