



Carnegie Mellon
Software Engineering Institute

Pittsburgh, PA 15213-3890

Impact of Differing Perspectives of SOA Stakeholders

Dennis Smith
Liam O'Brien
Grace Lewis

NCC Workshop: STEP2005, September 25, 2005

Sponsored by the U.S. Department of Defense
© 2005 by Carnegie Mellon University



“Simple” Problem

We were given the following problem:

- “We (an Army organization) have an existing command and control application. What would it take to reuse these components as services within two specific SOAs.”

We did an analysis and developed an approach for doing such an analysis

- Different aspects of method reported in *Architecture Recovery* and *Integration and Interoperability* workshops, as well as forthcoming SEI technical note

In speaking with different stakeholders, we realized that there is a Tower of Babel in which “never the twain shall meet”

- We are interested in exploring the development of an end to end engineering approach for SOAs

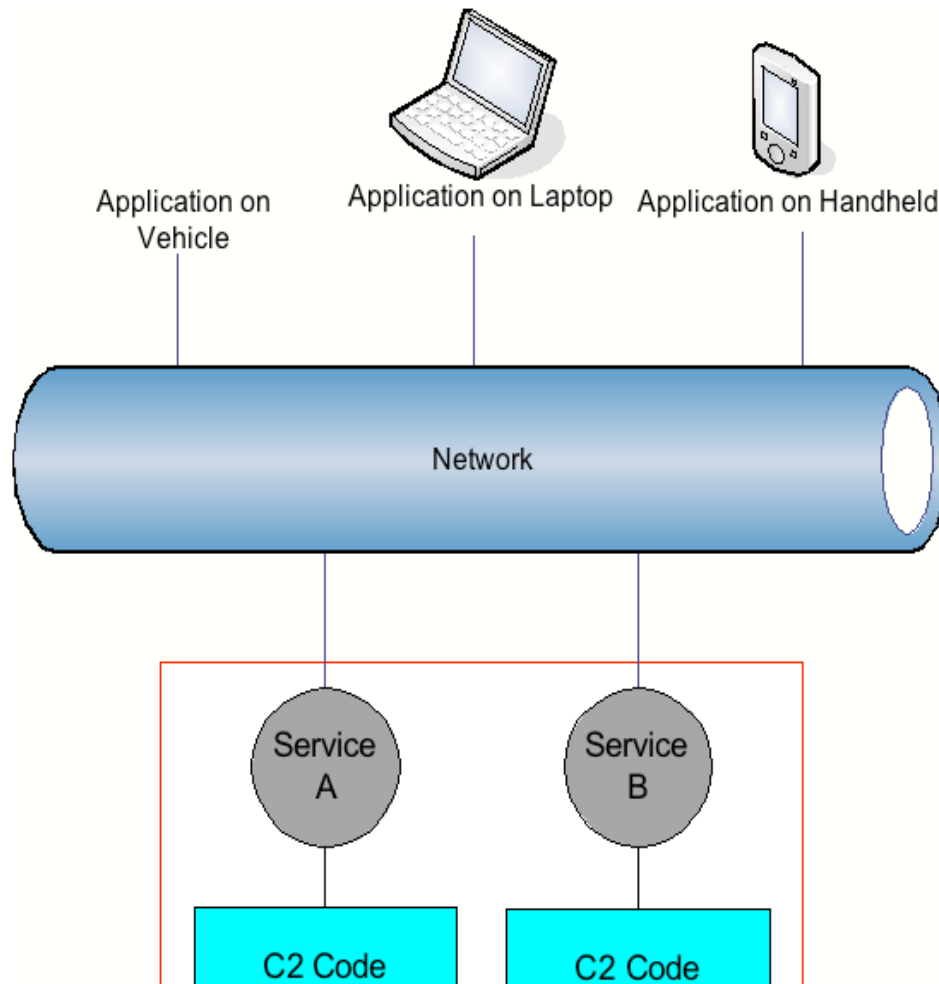


Context for the problem

Services are discovered and used as part of external applications.

Details of the network and its users are transparent to the service.

Current functionality is migrated as services.



Applications are written by third-parties who have specific requirements.

This is the only portion of the system over which the original developers have control.



Three Development Perspectives Within SOAs

Application Developers

- Focus on the discovery and connection to services - either statically or dynamically.

Service Providers

- Focus on the description and granularity of services.

Infrastructure Developers

- Focus on providing a stable infrastructure for discovery, communication, security, and QoS requirements.



Application Developers

- Focus on the discovery and connection to services - either statically or dynamically.
- Challenges include
 - Articulate realistic requirements
 - Identify the right services
 - Understand the semantics of the information being exchanged
 - Determine rules to follow when services are no longer available (in the case of static binding)



Service Providers- 1

Focus on the description and granularity of services

Challenges include

- Mapping of service requirements to component capabilities
- Description and granularity of services with acceptable Quality of Service (QoS)
- Determination of migration feasibility of potential services from legacy applications through methods such as SMART
- Investigate target data model to determine compatibility and redundancy.



Service Providers- 2

Challenges (continued)

- Writing new service specific code and wrappers
- Negotiate relevant interfaces with infrastructure developers
- Determine the effort and analyze the feasibility of translating legacy data types into serializable data types that can be transmitted in messages (some platforms call this data readers and data writers).
- Determine the effort required to describe the services.



Infrastructure Developers- 1

- Focus on providing a stable infrastructure
- Challenges include:
 - Providing common infrastructure services for discovery, communication, security, and QoS requirements.
 - Analyze the effort involved in writing the code that receives the request, translates it into calls into the legacy systems, and produces a response
 - Communicate the target environment.
 - Bindings, messaging technologies, communication protocols, service description languages, and discovery services.



Infrastructure Developers - 2

When based on commonly accepted standards, it should be easy to enable service providers to write compliant services

- We have seen cases where common standards are used in non-standard ways
- The SOA, in such a case, might deviate enough so that it becomes a standard of its own – requiring its own long term evolution and maintenance

In other cases, for reasons of security and performance, infrastructure developers are building the target architecture from the ground up

- This creates the usual problems of complex development, understanding of user needs, definition of interfacing standards, documentation, training, and perpetual upgrades



Problems

No good published strategies for building end-to-end systems based on SOAs.

- Current work usually focuses on only one of the development perspectives and assumes an “ideal environment”.

There are no approaches for understanding end-to-end quality of service.

Technologies that SOAs are based on are still maturing.

- It is not often clear what works and what does not work.



Research Needs

Strategies and guidance for end-to-end systems based on SOAs

- Issues of dynamic composition (critical for long term vision, but very immature with current technology)

Understanding and guidance for SOA issues from perspectives of each type of development perspective

- Application Developer
- Infrastructure Developer
- Service Provider

Comprehensive understanding of end-to end Quality of Service issues



CarnegieMellon
Software Engineering Institute

Discussion

Application Developer Perspective

Infrastructure Developer Perspective

Service Provider Perspective



Research Issues

- Where do providers come from; what is their background
- Development processes (standards) for SOAs
- Identification of when/where to use selected technologies; maturity of available solutions vs. “Design via PPT” proposal
- Scenarios towards roadmap for guidance; model problems
- Use of architectural styles
- Superimposed QAs (e.g., security) on top of technology infra.
- Separating hype vs. reality (cf. NCC ‘97)
- Scalability; internal infrastructure vs. global deployment; standards, ontologies for specific domains; Babel fish
- Rosetta net (electronics industry), Oasis standard
- Relaxed requirements; semantics, QoS for actual use
- Syntax ->* Semantic transformations; intermediate levels
- Enhanced requirements engineering in light of QoS issues
- Model transformations w/meaning preserved as expected; synchronizations, equivalence



Research Issues ...

- The semantic layer cake (tasty research!)
- Use of rich vs. poor schemas; information extraction
- OceanStore (UCB) project
- Testing / analysis / evaluation
- Gap analysis of current protocols (mainly syntax)
- Managing complexity; weight of solutions
- Visibility into the solution (cf. PRDS)



Discussion: ML-P2P

- How to implement? Experiment? Evaluate?
- Geographical distribution