Carnegie Mellon
Software Engineering Institute
Pittsburgh, PA 15213-3890

Opportunities for Research Collaborations with SEI ISIS Initiative

Dennis Smith

Integration of Software Intensive Systems (ISIS) Initiative

Sponsored by the U.S. Department of Defense

Carnegie Mellon
Software Engineering Institute

Outline

Students are looking for topics for Master's thesis and doctoral dissertations

At ISIS we are conducting hands-on evaluations of technologies that are relevant for achieving interoperability

We invite you (or your students) to contact us for interesting research collaborations that may benefit all concerned

Examples:

- Model problem approach
- Analysis of reuse potential of legacy components for SOA migration
- End to end SOA engineering

Model Problem Approach

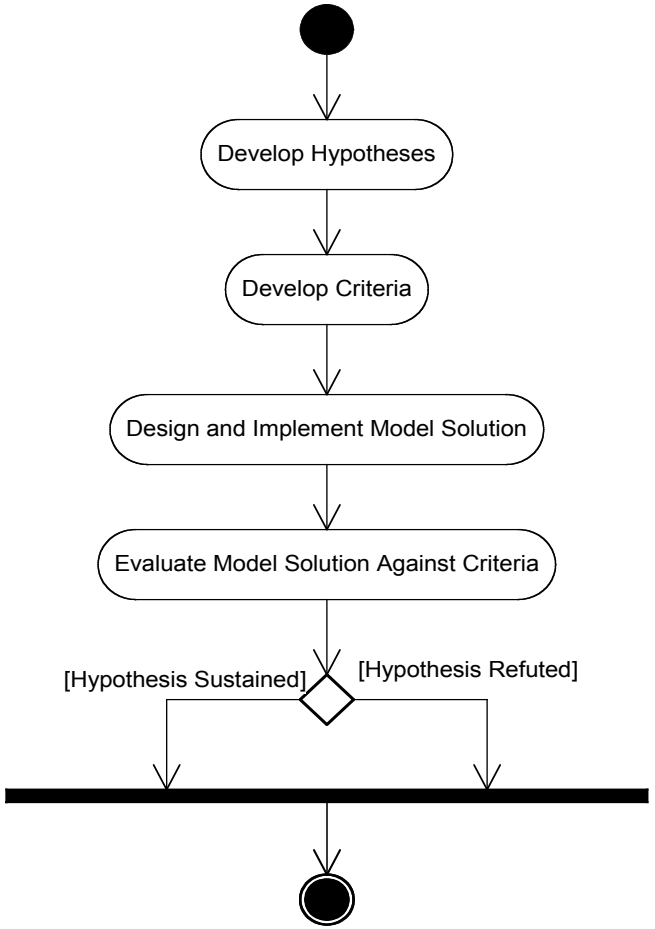
Purpose: To examine the gaps between what technologies and approaches offer and what users expect of them.

End goal: To provide users with information about what can be expected by the current state of technology and to provide technology suppliers with information about user expectations, hoping to help bridge these gaps.

The model problem approach involves

(1) formulating hypotheses about the technology

(2) examining these hypotheses against very specific criteria through experimentation.



Carnegie Mellon
Software Engineering Institute

MDA Hypotheses

1. The use of MDA reduces development time
   - Partially refuted
   - Development time is not reduced for the development of the first application for which a tool is used.
   - It may be considerably reduced for subsequent applications developed for the **identical** platform.

2. The use of MDA-based development tools frees the developer from learning the low-level details of the target platform and underlying infrastructure
   - Refuted
   - Tool, models, and transformations must be configured before any type of code is generated.
   - Configuration requires very good knowledge of the infrastructure target platform.
   - Frees the developer from having to write infrastructure code, but it does not free the developer (or whoever is doing configuration) from learning the low-level details of the target platform and underlying infrastructure.

Carnegie Mellon
Software Engineering Institute

Web Services Hypotheses

1. It is fairly easy to connect components developed for the same platform using Web Services (Sustained).

2. There is a large number of public, easily-locatable, and high-quality Web Services that can be used in applications (Refuted).

3. There will not be problems regarding data types if Web Services are used to connect components developed for different platforms (i.e., J2EE and .NET) (Sustained).

The more interesting learning is that we have had first-hand experience with the technologies and have been able to understand what their real status is.

Carnegie Mellon
Software Engineering Institute

ISIS Model Problems

Model Problems in Technologies for Interoperability series.

Published
- Model-Driven Architecture (MDA)

In Process
- Web Services
- Ontology Web Language for Services (OWL-S)

Planned
- Open Grid Service Architecture (OGSA)
- Standards and technologies related to Web Services, such as
  - Universal Description Discovery and Integration (UDDI)
  - WS-Security and Security Assertion Markup Language (SAML)
  - Web Services Flow Language (WSFL)
  - Web Services Conversation Language (WSCL)
- Component Frameworks
  - Java 2 Enterprise Edition (J2EE)
  - Microsoft .NET

Carnegie Mellon
Software Engineering Institute

Migration of Legacy Systems to SOAs

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

Carnegie Mellon
Software Engineering Institute

Service Migration and Analysis Reuse Technique (SMART)

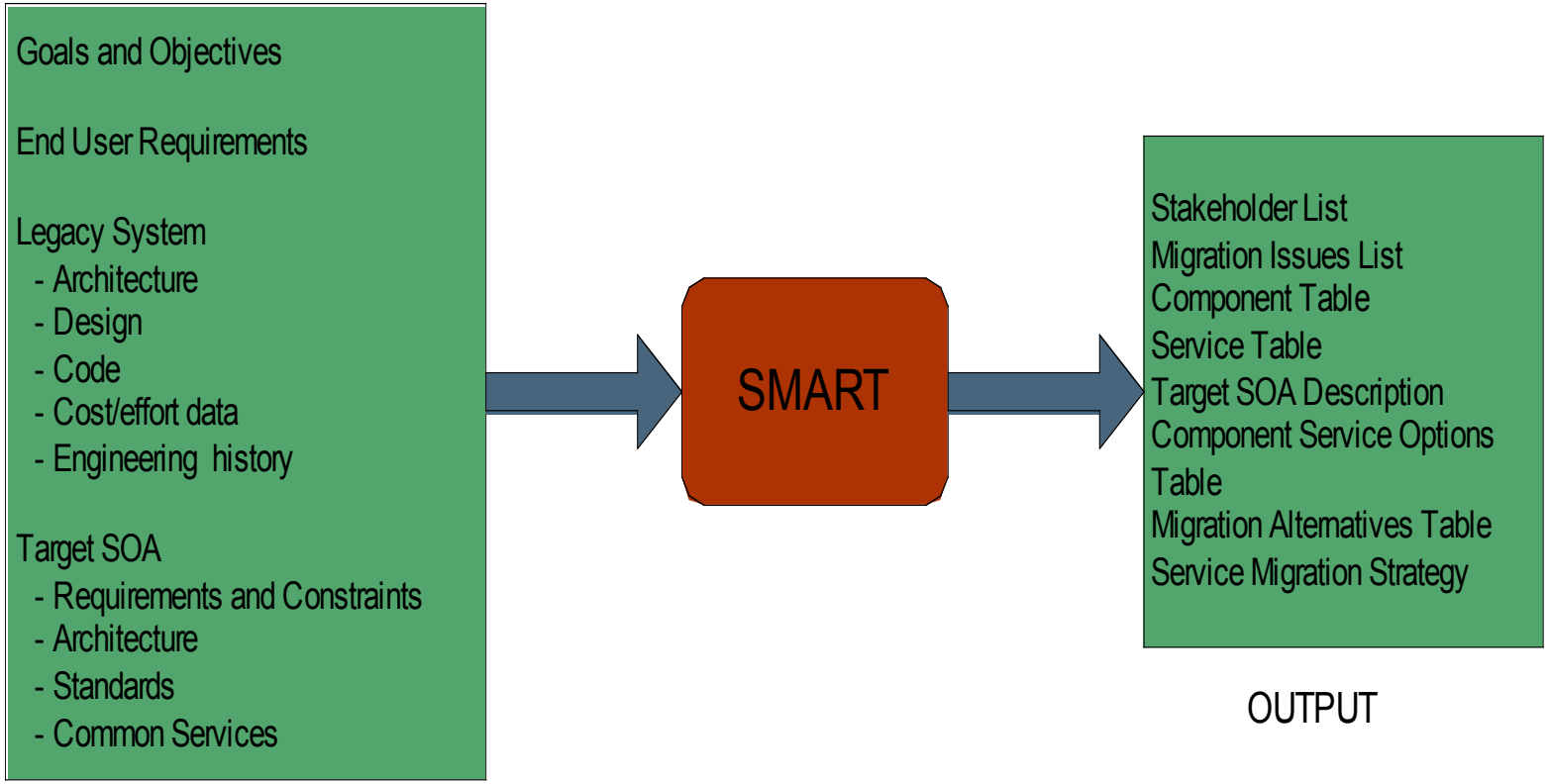
SMART provides a preliminary analysis of the viability of migrating legacy components, to services, migration strategies available, and the costs and risks involved.

- What components can reasonably be used to derive what services?
- What sorts of activities must be performed to accomplish the migration?
- What strategies are most appropriate for the migration effort?

Research Issues:

- New pilots
- Tools
- Further development of method

SMART Input and Output

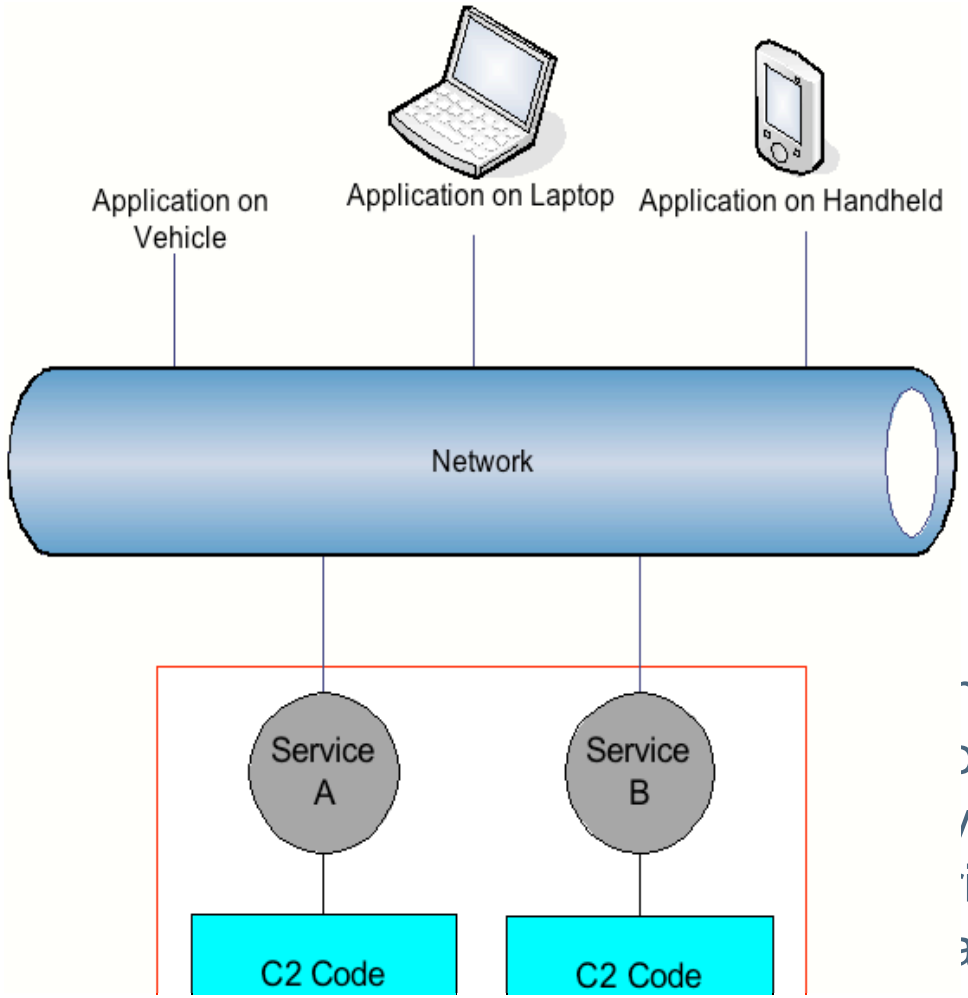


Need for End to End SOA Engineering

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.

his is the only ortion of the system ver which the iginal developers ave control.

Carnegie Mellon
Software Engineering Institute

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.

Carnegie Mellon
Software Engineering Institute

For Further Information

Grace Lewis
glewis@sei.cmu.edu

Dennis Smith
dbs@sei.cmu.edu