Service-oriented architectures and AADL modeling

Oleg Sokolsky
Real-Time System Group
University of Pennsylvania

SAE AADL Working Group Meeting
January 24-27, 2006
Outline

• **Motivation**
  - Enhance AADL models with information on component interaction

• **Service-oriented architectures**
  - in embedded domains
  - based on approach of Ingolf Krueger (UCSD)

• **Service orientation in AADL**
  - Pre-pre-pre-proposal...
AADL: connections and flows

- Flows: abstraction of communication
- Associate QoS properties with flows
  - End-to-end delays, transfer rates, etc.
AADL: connections and flows

- Implementations map flows to connections

- Architectural checks enable validation of implementations
Motivation: flows vs. interactions

• Flows are very suitable for stream processing
  - Sense, transform/compute, actuate
• Flows are less appropriate for reactive systems
  - independent
  - unidirectional
• Component interactions in a reactive system
  - bi- and multi-directional
  - follow a protocol
Related ports and connections

• We need to be able to specify that
  - A group of ports are semantically related
  - Connections on these ports are established together and provide a certain protocol
• Port groups serve this purpose to some extent
Interacting flows

• We need to be able to specify that
  - A group of flows establishes a protocol
  - Collectively provide certain QoS
Refinement of interactions

• We need to be able to refine multi-party interactions into coordinated sequences (more generally, patterns) of connections and flows.
The perverse beauty of UML

- One reason for UML popularity is the variety of supported views:
  - architecture
  - behavior
  - interaction
  - ...

- Problem:
  - absence of rigor
  - too much diversity, too little unity
Outline

• Motivation
  – Enhance AADL models with information on component interaction

• Service-oriented architectures
  – in embedded domains
  – based on approach of Ingolf Krueger (UCSD)

• Service orientation in AADL
  – Pre-pre-pre-proposal...
Services

• A uniform way of distributed access to software functions in distributed systems
• “Web services” over Internet
  - Definition of functionality and interaction
    • WSDL
  - Abstraction of physical distribution
    • Portability
• Embedded systems domain
  - Open Services Gateway Initiative (OSGi)
    • Services on a LAN
Services in the automotive domain

- **Automotive Multimedia Interface Collaboration**
  - Collaboration of eight major automakers
  - Specifications for
    - vehicle service interface
    - human-machine interface
    - off-board navigation services
  - Example services:
    - status: ignition, brake, door lock
    - odometer readings, engine oil pressure
    - vehicle location (GPS), identification, etc.

Interaction in architectural modeling

- Services delivered by a software system are cross-cutting aspects of the architecture
  - Should be represented explicitly on the level of ADL
- Services as architectural building blocks
- Services are elicited from user requirements
- Design problem:
  - Explore multiple architectural configurations that implement a given set of services

Definition

• Service:
  - Interaction pattern required to accomplish a specific task

• Many-to-many:
  - A component may provide more than one service, and multiple components may be involved in a service

• Consider services independently from components
Service-oriented architecture

• A service is an interaction pattern between a set of roles
• Roles are “placeholders” for components
  - A component can play many roles
  - Component can switch between roles
• Architectural configuration
  - Mapping between roles and components
Service-oriented development

use case graph
roles
services
domain model

architecture
mapping
component configuration

C1:R1
C2:R2
C3:R2
C4:R3

R1
R2
R3
R1
R2
R1
R3

C1
C2
C1
C2

configuration
mapping
architecture

Outline

• Motivation
  - Enhance AADL models with information on component interaction
• Service-oriented architectures
  - in embedded domains
  - based on approach of Ingolf Krueger (UCSD)
• Service orientation in AADL
  - Pre-pre-pre-proposal...
AADL and service orientation

- Many key notions of service-oriented architectures are present in AADL
- A message set that comprise a service can be given by a port group
  - A stricter semantic interpretation is needed
- A two-layer architecture with mapping between the layers is given by software vs. platform components
Services as existing features

- Extend a port group with the description of interaction
  - MSC or interface automaton
  - Works for bi-party services
Services as modified features

- Allow multi-party interactions
- Relax the “single unit” requirement for external connections of a port group
  - Probably not a good idea
Services as new features

• Introduce a service feature that has references to ports involved in the interaction
• Is there a difference between “provide service” and “require service”?
Service implementations

- Set of rules to enable architectural checks:
  - Subcomponents properly participate in services of the containing component implementation
  - Subcomponents enable service features of the containing component
Interactions and behavior

- Specification of interactions is complementary to specification of component behaviors
- Component behavior can be used to check that the component will comply with the protocol
Summary

- Interactions between components in the system can and should be modeled on the architectural level
- The notion of a service encapsulates component interactions as a feature
- New architectural checks can further improve design exploration

- Comments?