Service Modeling in TMOS Using UML 2.0

Francesco Caruso, Telcordia
Josephine Micallef, Telcordia
Manan Thakkar, Lockheed Martin
Agenda

- SOA Motivation
- SOA Modeling Requirements
- TMOS Service Model
- Lessons Learned
Motivation for Using SOA

SOA is a **paradigm** for realizing business processes and functions that span large distributed systems

- Alignment of business/mission and IT processes
- Interoperability across systems – despite Heterogeneity
  - No matter what the system designs and implementations
  - Rather than fighting it, accept it, deal with it, design for it
- Integration of applications and systems
  - e.g., Legacy systems, departmental applications or systems
- Change
  - Evolvable for new uses and new users for capabilities
  - Minimizing impacts of changing implementations and technology
  - Rather than fighting it, accept it, embrace it, design for it
SOA Services In TMOS

• Self-contained, reusable business functionality provided and implemented in components

• Represents capability to perform tasks that form a coherent functionality from the point of view of provider and requester entities consistent with constraints and policies (e.g. QoS, SLA) as specified by the *service specification*

• Service specification represents both *structure and behavior* – i.e., interactions between provider and consumer

• Realized (i.e., implemented) in a physical component using Web Service technologies or other distributed technology (e.g., CORBA, Java RPC, Messaging)

SOA in TMOS is broader than Web Services (WS), which is just one possible implementation technology
TMOS SOA Modeling Requirements

• Expressive for complex mission services that use rich message exchange patterns (MEP), such as
  – Request followed by multiple responses (e.g., req/ack/resp)
  – Update request with notification (req/resp/notify)

• Supports implementation in variety of implementation technologies tuned to QoS requirements
  – E.g., Web services, CORBA, MOM, Java RPC, …

• Exploits model-driven development (MDD) to
  – Manage change effectively, both in system requirements and computing platform
  – Improve productivity by generating developmental artifacts and documentation
  – Improve quality by automating parts of the development process
Challenge Addressed

- SOA Modeling Requirements common to large, complex system-of-systems

- Pieces of solutions exist
  - Proprietary service modeling techniques provided by software design tools
  - Work in progress in standards organizations to define a UML Service Model
  - Model-driven generation capabilities supported by software design tools

- WSDL, Web Service Description Language, is the right concept but ties to implementation

- UML, Unified Modeling Language, is at the right level, but does not define a standardized service model

Develop a UML Service Model that meets the needs of TMOS (and similar systems) in alignment with emerging standards
• Service modeled as a UML Port typed with service specification class
• Simple request/response MEP modeled by provided interface
• Rich MEP, such as send multiple responses or notifications, are modeled with additional required interface
• Consumer service invocation is modeled by requisition port typed with the dual of the service specification
- Used UML Profile mechanism, which specifies extensions of UML for domain and architecture specific modeling
- To define UML 2.0 Profile for modeling SOA services in TMOS, inspired by emerging standards
  - OMG UML Profile and Meta-Model for Services (UPMS), standardization in progress
Lessons Learned

• Existing service description language – WSDL – too low-level for TMOS modeling needs
  – Defined a service model that was sufficiently expressive and also can be used to generate the necessary WSDL(s)

• TMOS service model leveraged emerging standards
  – Used OMG’s UPMS foundation and OASIS Reference Model for Service Oriented Architecture
  – Positions TMOS to leverage COTS tool implementations once standards are ratified and supported

• Represent service model as a TMOS project-wide UML 2.0 profile to promote SOA design governance (e.g., consistency, interoperability)
  – TMOS service model was non-trivial to define
  – Implementation of the TMOS service model in Rational Software Architect was straightforward

• MDA approach has demonstrated benefits
  – Generated service documentation (e.g. ICD, IRS, Architecture description), program CDRLs and traceability reports
BACKUP SLIDE
UML Profile for TMOS SOA Services

- **UML Port**
  - `tmosService`
  - `tmosServiceRequisition`

- **UML Class**
  - `tmosServiceRequstionSpec`
  - `tmosServiceSpec`

- **UML Interface**
  - `tmosService_IF`
  - `tmosService_IFCallback`
  - `tmosService_IFNotification`

- **UML Class**
  - `tmosMessageType`
References

• UPMS
  – http://www.omg.org/docs/soa/06-09-09.pdf

• OASIS