NeMo Developer Session – Service Development Tools

Johannes Fähndrich and Nils Masuch
DAI-Labor, Technische Universität Berlin
Berlin, Germany
e-mail: johannes.faehndrich@dai-labor.de
Agenda

• The NeMo Architecture

• Practical challenges for efficient service integration into networks

• Developing & utilizing NeMo services
  • Service process development with VSDT
  • Service description & search with SSM
  • Service execution environment
Actors in the NeMo Hyper-Network

- End-user
- Vehicles
  - OEM backend
  - electroMobility Service Provider
- Charging
  - electroMobility Service Provider
- Energy
  - electroMobility Service Provider (Smart Grid Integration)
- Business Integration
  - 3rd Party Service Provider

NeMo Hyper-Network

Charging Network
Billing & Payment
panEuropean eRoaming framework
Development Tool Kits
Common Information Model
Service Creation
Service Management

Information & Communication Technology

Privacy & Security
Data Translators
Actors’ monitoring
Service optimiser
Partner Management
Contract Management
Standard Protocols
Business Models
NeMo Business Network

Shared distributed Database / Ledger:
1. Partner Management
2. Services Store
3. Contract Management

12/10/2017 NeMo, Developer Session
NeMo Service Development & Integration
Agenda

• The NeMo Architecture

• Practical challenges for efficient service integration into networks

• Developing & utilizing NeMo services
• Service process development with VSDT
• Service description & search with SSM
• Service execution environment
Motivation

Software bugs cost U.S. economy $59.6 billion annually, RTI study finds [1]

BPMN is referenced in more than 24,000 scientific publications
BPMN is used in at least 300 patents [2]

Automation is still pending

[1] Research Triangle Park, NC
http://www.rti.org/newsroom/news.cfm?obj=DA7F8E6-4A4F-4BFD-B7E0FA3C04D9E22

http://doi.org/10.1016/j.infsof.2016.02.002
Motivation

50% Java
Program Logic described in Java by a developer

55% WSDL
Web Service Description Language is created out of Java Annotations

98% OWL
EMF to OWL converter to create a Semantic Representation

70% EMF Model
An EMF Model is generated out of the Java classes

Java to OWL-S
From the java methods we extract the OWL-S
Motivation

• Complexity of distributed systems is increasing
  – Huge amounts of services
  – High degree of dynamics
  – Heterogeneous service providers
• Management of an efficient interoperability gets more and more difficult
• Further dynamic behaviour in huge distributed systems is a key requirement for intelligent systems/agents/components
• Semantic Web Service concepts, such as Service Matchmaking and Service Composition are promising approaches...
Challenges

• An important topic is the autonomic interpretation of services’ functionality

• There are multiple semantic service description languages:
  – WSMO, OWL-S, SAWSDL, SA-REST, etc.

• However:
  – Syntactical complexity of the descriptions high
  – Manual creation cumbersome and error-prone
  – Relation between development effort and benefit still not sufficient
Challenges

• An important topic is the autonomic interpretation of service’s functionality

• There are multiple semantic service description languages:
  – WSMO, OWL-S, SAWSDL, SA-REST, etc.

• However:
  – Syntactical complexity of the descriptions high
  – Manual creation cumbersome and error-prone
  – Relation between development effort and benefit still not sufficient
Goals

• Provide **support for the semantical enhancement of functionalities** without changing the developers workflow completely

• Facilitate the development of **E-Mobility services** that automatically find, invoke and combine other services to reach a certain goal
Agenda

• The NeMo Architecture

• Practical challenges for efficient service integration into networks

• Developing & utilizing NeMo services
  • Service process development with VSDT
  • Service description & search with SSM
  • Service execution environment
NeMo Service Development

Provide a Service Development Environment that
• allows for the specification of service processes
• integrates service search at design-time based on semantic service descriptions
• enables the composition of services to value-added services
• is itself running within the cloud infrastructure
• comes with testing features
NeMo Service Development

[Diagram of NeMo Service Development process]

12/10/2017 NeMo, Developer Session
NeMo Service Development

[Diagram of NeMo Service Development process]
Agenda

• The NeMo Architecture
• Practical challenges for efficient service integration into networks
• Developing & utilizing NeMo services
  • Service process development with VSDT
  • Service description & search with SSM
  • Service execution environment
Question: How to create a Service?
Service Process Development
Use Case
Use Case on Services for Electric Mobility
Service Process Development with VSDT
Service Process Development with VSDT

- VSDT (Visual Service Design Tool)
- Description of processes with BPMN to compose services
- Adaption points through semantic templates
Agenda

- The NeMo Architecture
- Practical challenges for efficient service integration into networks
- Developing & utilizing NeMo services
- Service process development with VSDT
- Service description & search with SSM
- Service execution environment
Question: How to create a Semantic Service Description?
Service description & search with SSM

- Eclipse View integrated in a larger tool-suite called Toolipse 3
- Support for the development of OWL-S service descriptions using OWL and SWRL
- Based on a OWL-S Factory using OWL API 3.5
- Integration of a Service Matchmaking component called SeMa² [Masuch2012]
SSM Ontology Management

- IO Parameters and Preconditions/Effects are based upon OWL concepts and relations
- SSM offers an Ontology Browser
  - Integration of local and external ontologies with automatic import reloading
  - Quick Search of concepts over multiple ontologies
  - Overview about concept’s properties including Domain Range
SSM Ontology Management

- OWL-S allows for formulating preconditions and effects using rule language SWRL (Semantic Web Rule Language)
- SSM features Human Readable Syntax:
  Example for precondition: `EmailAddress(?x) ∧ hasStringIdentifier(?x,"@") → null`
- SWRL Rule Parser implemented using ANTLR v4.1
Service description & search with SSM

Question: How to find the right Service?
SSM Search Service

- Designer can find available services at design time
- Search Tab offers the possibility to define a search template
- SSM invokes Service Matchmaker and searches for appropriate functionalities on the platform
SSM Integration into BPMN
SSM Integration into BPMN
Agenda

• The NeMo Architecture

• Practical challenges for efficient service integration into networks

• Developing & utilizing NeMo services
• Service process development with VSDT
• Service description & search with SSM
• Service execution environment
Service Execution Environment

Question: How to execute such a service?
Service Execution Environment
Goals - Review

- **Support for the semantical enhancement of functionalities** through tools like the SSM

- Facilitate the development of **E-Mobility services** by adding a semantic layer and integrating a SOA.

- Processes can be modeled in BPMN with additional flexibility
Thank you!

Johannes Fähndrich
Nils Masuch
DAI-Labor, Technische Universität Berlin
johannes.faehndrich@dai-labor.de
Agenda

• Practical challenges for efficient service integration into networks

• The NeMo Architecture

• Developing & utilizing NeMo services

• Service process development with VSDT

• Service description & search with SSM

• Service execution environment

• NeMo Use Case
Question: How to integrate your ontology?

MOVE BEFORE SERVICE SEARCH
SSM Code Integration
SSM Code Integration

12/10/2017 NeMo, Developer Session
SSM Code Integration
Question: How to integrate service templates?
VSDT Process