



Orchestration of Public Administration Services using the semantic technologies

Marek Skokan, Peter Bednár

Web Services

- remote program procedures
- Service Oriented Architecture
 - decoupling of service requester and service provider (“weak” binding)
 - dynamic search and invocation of services
- Web services technologies
 - WSDL – describes structure of messages and technical access to the service (i.e. port, protocol, etc.)
 - SOAP – XML protocol for data exchange over the network
 - UDDI – registry for web services

Semantic Web Services

- Limitations of current technologies – data is described on the “syntactical” level only => limited discovery, composition, execution
- Semantic web services => add semantic annotations to support
 - *discovery* – find service capable to solve given problem
 - *composition* – create composite service to solve more complex problems
 - *execution* – monitoring, compensation, auditing
 - *mediation* – solve various heterogeneity problems

Orchestration and Choreography

- important for automatic execution and composition
- both describe behavior of web services from two different perspectives
- *Choreography* – how the requester can communicate with one service to achieve given goal (interaction *requester* ↔ *service*)
- *Orchestration* – how the composite service is using other services to achieve more complex goal (interaction *composite service* ↔ *services*)

Access e-Gov context

- FP6 EU project
- integrate existing “traditional” and electronic services government services
- create complex **hybrid** scenario for the specific life event to guide citizen/company to achieve particular goal
- semantic annotations – Web Service Modeling Ontology

WSMO model for orchestration and choreography

- based on Abstract State Machines
- state is represented as the ontology
- transitions
 - *if* some condition *then* change state (i.e. add/remove some instance or change value of some instance's attributes)
- too “abstract” for Access e-Gov
 - difficult to navigate citizens
 - non-intuitive modeling
 - not finished yet, unspecified dataflow

Access e-Gov process model

- workflow based
- consists of: activity nodes, control links and dataflow links
- **activity nodes**
 - communication nodes – *Send, Receive*
 - AchieveGoal and InvokeService
 - control node – *Decision, Fork and Join*
- **control links**
 - specify execution order – various workflow patterns – branching, cycles, parallel execution

Access e-Gov process model

- **dataflow links**
 - copy and additionally transform value from source node to target node
 - source logical expression \Rightarrow target logical expression
 - mediation on the data level and on the process level – i.e. it is possible to express mediation between parent process (orchestration model) and children processes (choreography models)

Modeling of traditional services

- **customizations**
 - questions used to identify specific citizen case and customize process model
 - questionnaires modeled as concepts, answers as instances, *Receive* nodes (optionally conditioned)
- list of **pre-conditions**
 - required artifacts (i.e. documents, application forms, payments, etc.)
 - *Receive* nodes (when user can provide artifact directly), *AchieveGoal* (when service is needed)
- list of **post-conditions**
 - provided artifacts

Conclusions

- workflow based process model for orchestration and choreography in hybrid scenarios
- allows coordination of all actors – i.e. traditional services, electronic services and human actors (i.e. citizens)
- compatible with existing modeling notations – UML, BPMN
- dataflow model which allows data mediation and process mediation