Automated Generation of Semantic Service Descriptions by Leveraging on Service Governance Models

René Fleischhauer, Michael Stollberg
SAP Research Dresden
Software Engineering & Tools

August 2011
Agenda

01 Introduction
02 Related Work
03 Conceptual Design
04 Implementation
05 Evaluation
06 Conclusion
01 Introduction
01 Introduction
Problem Statement

- **Current service management support is insufficient**
  - A lot of services in modern service-based business systems
  - Applied service description models are insufficient (syntactic focus)
    - Additional information is needed
    - Presented in an informal and not machine processable manner
  - Service search & consumption requires a lot of manual intervene

- Existing research and development efforts are not applied in industrial practice
  - Provide sophisticated frameworks for semantic service description
  - **BUT:** Do not consider the cost-effective creation of semantically enabled service descriptions
  - Unaffordable to migrate existing service infrastructures to proposed frameworks
01 Introduction
Motivating Example

Service Search via Enterprise Service Workplace

- Necessary: Manual consideration of provided information
- A lot of manual intervene
- Require expert knowledge
- Hampers effective use of services

- Need for more sophisticated service descriptions which enables automated processing and more precise query languages
Leverage (SOA) Governance Models

- Governance models are organizational structures common in industrial service infrastructures:
  - Definition of patterns and guidelines
  - Ensure compatibility and interoperation
- Quality management process enforce model compliance (governance process)
- Provide a lot of implicit knowledge about organized services
- Enforced compliance enable automated access

Figure: SOA Governance Model for Enterprise Services
Explication of implicit knowledge

Transfer implicit knowledge of Governance Models into explicit representation

- Definition of an appropriate metamodel
- Access and transformation via automated process
- Use of appropriate storage and querying technologies to enable better service management
01 Introduction

Outline

- The presented work provides
  - An infrastructure to create richer service descriptions at **minimal costs**
  - A proof-of-concept implementation for **RESTful services using the OData protocol**

- Outline of remaining presentation
  - Discussion of related work
  - Detailed introduction into conceptual design
  - Short overview about implementation details
  - Presentation of Evaluation results
  - Summary and Conclusion
02 Related Work
## 02 Related Work

### Requirement

<table>
<thead>
<tr>
<th>Enriched Service Description Model</th>
<th>Candidates</th>
<th>Used Technologies</th>
</tr>
</thead>
<tbody>
<tr>
<td>• <strong>Semantic Web Frameworks</strong></td>
<td></td>
<td>• SAWSDL</td>
</tr>
<tr>
<td>(OWL-S, WSMO, SAWSDL)</td>
<td>• <strong>Technical Description Models</strong></td>
<td>• OData</td>
</tr>
<tr>
<td>• (WSDL, WADL, OData)</td>
<td></td>
<td>• USDL</td>
</tr>
<tr>
<td>• <strong>Business Description Models</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(USDL, WS-* *)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Representation Language

| • XML                              | • RDF         | • RDF             |
| • RDF                              | • OWL         |                   |

### Storage and Querying Facilities

| • Relational Data Bases & SQL      | • Triple Store | • Triple Store    |
| • Triple Stores & SPARQL           | & SPARQL       |                   |
02 Conceptual Design
03 Conceptual Design
Top-Level Application Design

Governance Model → defined above → Assignment Rules → targets → Enriched Service Description Metamodel

Governance Model
Assignment Rules
Enriched Service Description Metamodel
RDF(S) Transformation
Triple Store

RESTful Web Services
Extraction Process
Enriched Service Descriptions

determines
defined above
defined above
implements
transforms
operates on
creates
transforms
transforms
03 Conceptual Design
Assignment Rules

Two types of roles:
- For structured information

\[
\begin{align*}
\alpha_k(\text{EntityType}) &= \{\text{EntityType}\} \\
\alpha_i(\text{EntityType.Name}) &= \{\text{EntityType.hasName}\} \\
\alpha_m(\text{EntityType.Property}) &= \{\text{EntityType.hasProperty}\}
\end{align*}
\]

- For structured information
**03 Conceptual Design**

**Enriched Service Description Model**

- **Enriched Service Description Metamodel**
  - **Functional Description**
    - *What does the service do?*
      - Functionalities
      - Capabilities
      - Based on USDL (Functional Module)
  - **Technical Description**
    - *How to use the service?*
      - Service resources
      - Data types
      - Based on OData and USDL (Technical Module)
  - **Search Optimization**
    - *How can the service search be facilitated?*
      - Resource types
      - Applicable (HTTP) methods
      - Externalize protocol knowledge
04 Implementation
04 Implementation
Technical Architecture

Governance Model

RESTful Web Service

Frontend
controls

Extractor
Separated Thread

XML Documents
Assignment Rules Impl.
XML Techn.

Enriched Description

Transformation Process
Java Reflection
RDF Triples

Transformer
Separated Thread

stores

Sesame
05 Evaluation
05 Evaluation
Scenario-based Evaluation (1)

- Services for Customer Mgt?
- Services for Sales Order Mgt?
- Are the services combinable?
Listing 5.1 SPARQL implementation of Query 1

1. PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2. PREFIX esdm:<http://www.sap.com/sapdataprotocol/rdf-s#>
3. SELECT ?service ?serviceUri WHERE {
5. ?service esdm:hasUri ?serviceUri .
10. ?spec esdm:hasEntityType ?entityType .
11. ?entityType esdm:hasName "Customer" .
12. ?entityType esdm:isTopLevelType "true" .
13. ?entityType esdm:hasProperty ?prop1 .
14. ?prop1 esdm:hasName "Firstname" .
15. ?entityType esdm:hasProperty ?prop2 .
16. ?prop2 esdm:hasName "Lastname" .
17. ?entityType esdm:hasProperty ?prop3 .
18. ?prop3 esdm:hasName "SupplierId" .
20. ?navProp1 esdm:hasName "AddressInformation" .
22. ?navProp2 esdm:hasName "CommunicationData" .
24. ?navProp3 esdm:hasName "MarketingAttribute" .
25. }

© 2011 SAP AG. All rights reserved.
05 Evaluation
Scenario-based Evaluation (3)

Results

Solution approach provides substantial benefits in comparison to existing technologies
Precise selection of appropriate service
No manual inspection of informal descriptions needed

Complexity of the provided Enriched Service Description Metamodel allows different query types:
Search for structural and functional features
Search for semantic service behavior
Support for service composition
## Test Data Set

<table>
<thead>
<tr>
<th>Service Type</th>
<th>Service Number</th>
<th>Generated Triples</th>
<th>Elapsed Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP Test Services</td>
<td>105</td>
<td>417,175</td>
<td>4h 21s</td>
</tr>
<tr>
<td>Scenario Test Services</td>
<td>7</td>
<td>17,513</td>
<td>12 min 44 s</td>
</tr>
<tr>
<td>Total</td>
<td>112</td>
<td>434,688</td>
<td>4h 13 min 5s</td>
</tr>
<tr>
<td>Single Service</td>
<td></td>
<td>3,882</td>
<td>1min 16s</td>
</tr>
</tbody>
</table>

## Results

- **Scalable processing** (overall mean: 85.69 ms)
- **High performance** (highest execution time: 239.5 ms)
- **Reliable results** (low standard deviation)
06 Conclusion
Conclusion

- Governance information can be used to achieve richer and more structured service descriptions
- Automated extraction for concrete service allows the creation at minimal costs
- Significantly improved service search and consumption techniques can be supported

Possibilities for Future Work:

- Performance improvements (other triple store, query optimization, SPARQL-to-SQL rewriting)
- Sophisticated frontends
- Framework to integrate other types of governance information and services
Links

Presentation online

PDF and PPT (with animations)
Acknowledgements

This work has been supported and partially funded by the FINest project (FP7 – 285598).

Please refer http://www.finest-ppp.eu/ for further details.
Thank You!

Contact information:

René Fleischhauer  
rené.fleischhauer@sap.com

Michael Stollberg  
michael.stollberg@sap.com
Backup Slides
07 Backup
Usage Study RESTful Web Services

- 2008:
  - SOAP: 25%
  - REST: 60%
  - XML-RPC: 10%
  - JavaScript: 5%
  - AtomPub: 5%

- 2010:
  - SOAP: 15%
  - REST: 74%
  - XML-RPC: 3%
  - JavaScript: 6%
  - AtomPub: 2%
## Amazon S3 How-to's

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActionScript Library for Amazon S3</td>
<td>An ActionScript library for accessing Amazon's S3 service.</td>
</tr>
<tr>
<td>AWS: S3 for Ruby</td>
<td>A Ruby library for Amazon's Simple Storage Service, S3, REST API.</td>
</tr>
<tr>
<td>Java Library for Amazon S3 REST API</td>
<td>A library in Java for working with the Amazon S3 REST interface.</td>
</tr>
<tr>
<td>Java Wrapper for Amazon S3 SOAP API</td>
<td>A library in Java for working with the Amazon S3 SOAP interface.</td>
</tr>
<tr>
<td>Perl Wrapper for Amazon S3 API</td>
<td>A library in Perl for working with the Amazon S3 REST Interface.</td>
</tr>
<tr>
<td>PHP Wrapper for Amazon S3 API</td>
<td>From the site: This class is a standalone S3 REST implementation for PHP 5 using CURL. A PHP implementation with support for reading/writing large objects and streams, access control lists and bucket logging.</td>
</tr>
<tr>
<td>Python Interface for Amazon Web Services</td>
<td>From the site: An integrated interface to current and future infrastructural services offered by Amazon Web Services. Currently, this includes Simple Storage Service (S3), Simple Queue Service (SQS), Elastic Compute Cloud (EC2), Mechanical Turk, and SimpleDB</td>
</tr>
<tr>
<td>Ruby Library for Amazon S3 API</td>
<td>A Ruby Library for Amazon's Simple Storage Service's (S3) REST API.</td>
</tr>
<tr>
<td>Ruby Library for Amazon Web Services</td>
<td>RightScale's AWS gems provide Ruby interfaces to four key Amazon Web Services: EC2, S3, SQS, and SDB. The gems use Amazon's REST and query interfaces to provide full programmatic control. An optional HTTP layer retries and clears transient errors.</td>
</tr>
<tr>
<td>Using Silverlight with Amazon S3 Storage</td>
<td>Overview and source code for using Amazon S3 storage with Microsoft Silverlight.</td>
</tr>
</tbody>
</table>
07 Backup
Enriched Service Description Metamodel

- Search Optimization
- 47 Classes
- 54 Attributes
- 65 References
07 Backup
Detailed Technical Architecture
07 Backup
RDF Transformation
## 06 Backup
Quantitative Evaluation

### Results

<table>
<thead>
<tr>
<th>Query No.</th>
<th>Mean $\mu$</th>
<th>Median $\bar{x}$</th>
<th>Std. Deviation $\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>87.4</td>
<td>86.1</td>
<td>6.45 (7.39 %)</td>
</tr>
<tr>
<td>2</td>
<td>122.3</td>
<td>120.0</td>
<td>10.18 (8.33 %)</td>
</tr>
<tr>
<td>3 (1)</td>
<td>10.5</td>
<td>9.7</td>
<td>3.49 (33.18 %)</td>
</tr>
<tr>
<td>3 (2)</td>
<td>59.0</td>
<td>57.5</td>
<td>5.32 (9.02 %)</td>
</tr>
<tr>
<td>4</td>
<td>79.6</td>
<td>76.7</td>
<td>13.67 (17.11 %)</td>
</tr>
<tr>
<td>5</td>
<td>87.6</td>
<td>85.9</td>
<td>10.43 (11.90 %)</td>
</tr>
<tr>
<td>6</td>
<td>106.1</td>
<td>102.6</td>
<td>12.68 (11.96 %)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Query No.</th>
<th>Mean $\mu$</th>
<th>Median $\bar{x}$</th>
<th>Std. Deviation $\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>59.1</td>
<td>56.9</td>
<td>11.14 (18.84 %)</td>
</tr>
<tr>
<td>8</td>
<td>239.5</td>
<td>233.3</td>
<td>21.36 (8.92 %)</td>
</tr>
<tr>
<td>9</td>
<td>12.8</td>
<td>11.2</td>
<td>11.24 (87.94 %)</td>
</tr>
<tr>
<td>10</td>
<td>66.0</td>
<td>63.3</td>
<td>10.12 (15.33 %)</td>
</tr>
<tr>
<td>11</td>
<td>17.3</td>
<td>14.8</td>
<td>12.75 (73.60 %)</td>
</tr>
<tr>
<td>12</td>
<td>219.1</td>
<td>214.4</td>
<td>16.40 (7.49 %)</td>
</tr>
<tr>
<td>13</td>
<td>33.3</td>
<td>34.6</td>
<td>2.87 (8.38 %)</td>
</tr>
</tbody>
</table>

Overall mean: 85.69 ms
06 Backup
Test Data Set (Quantitative Evaluation)

<table>
<thead>
<tr>
<th>Service Number</th>
<th>Generated Triples</th>
<th>Elapsed Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP Test Services</td>
<td>105</td>
<td>417 175</td>
</tr>
<tr>
<td>Scenario Test Services</td>
<td>7</td>
<td>17 513</td>
</tr>
<tr>
<td>Total</td>
<td>112</td>
<td>434 688</td>
</tr>
<tr>
<td>Single Service</td>
<td></td>
<td>3882</td>
</tr>
</tbody>
</table>

Tests
- Client-Server setup (real-world aligned)
- Use of 13 Queries from qualitative evaluation
- Each query repeated 100-times for statistically valid results
07 Backup
Interaction Backend Components
06 Backup
Example SPARQL Query

Listing 5.1 SPARQL implementation of Query1

```sql
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX esdm:<http://www.sap.com/sapdataprotocol/rdf-s#>
SELECT ?service ?serviceUri WHERE {
    ?service esdm:hasUri ?serviceUri .
    ?spec esdm:hasEntityType ?entityType .
    ?entityType esdm:hasName "Customer" .
    ?entityType esdm:isTopLevelType "true" .
    ?entityType esdm:hasProperty ?prop1 .
    ?prop1 esdm:hasName "Firstname" .
    ?entityType esdm:hasProperty ?prop2 .
    ?prop2 esdm:hasName "Lastname" .
    ?entityType esdm:hasProperty ?prop3 .
    ?prop3 esdm:hasName "SupplierId" .
    ?entityType esdm:hasNavigationProperty ?navProp1 .
    ?navProp1 esdm:hasName "AddressInformation" .
    ?entityType esdm:hasNavigationProperty ?navProp2 .
    ?navProp2 esdm:hasName "CommunicationData" .
    ?entityType esdm:hasNavigationProperty ?navProp3 .
    ?navProp3 esdm:hasName "MarketingAttribute" .
}
```