The EasySOC Project: A Rich Catalog of Best Practices for Developing Web Service Applications

J. M. Rodriguez M. Crasso C. Mateos A. Zunino
M. Campo

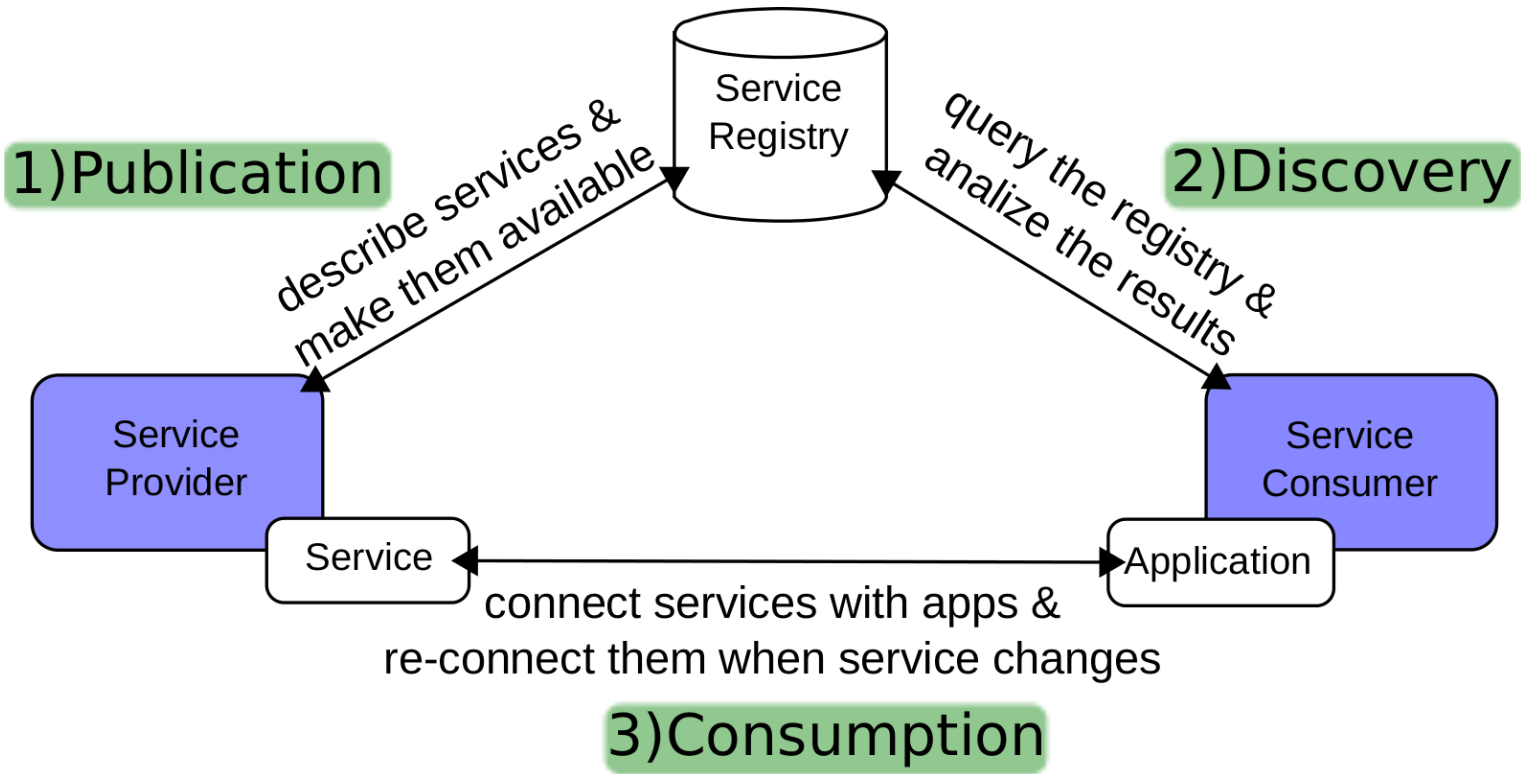
ISISTAN - UNICEN, Tandil, Argentina
CONICET

Conferencia Internacional de la Sociedad Chilena de Ciencia de la Computación

Outline

Paradigm Actors & Activities

- SOC is a relative new paradigm for distributed computing



Catalog Motivation

- Web Service technologies are widely used to implement SOC
  - They allow implementing the paradigm over the Internet
- Web Services are incessantly achieving effectiveness

*However the definition of **guidelines** for **developing** SOC-based software is still an incipient research topic*

Catalog Motivation

- Web Service technologies are widely used to implement SOC
  - They allow implementing the paradigm over the Internet
- Web Services are incessantly achieving effectiveness

*However the definition of **guidelines** for **developing** SOC-based software is still an incipient research topic*

Guidelines for Service Publication I

Goal

- Lift service chances of being discovered, understood and reused

Actor

- Provider

Summary

- Describe services offered functionality EFFICIENTLY
- Avoid WSDL discoverability anti-patterns [1]
- 6 steps for improving service APIs using WSDL
- Tool support

Guidelines for Service Publication II

1. Separating the schema –i.e. XSD code– from the definitions of the offered operations.
2. Removing repeated WSDL and XSD code.
3. Putting error information within Fault messages and only conveying operation results within Output ones.
4. Replacing WSDL element names with self-explanatory names if they are cryptic.
5. Moving non-cohesive operations from their port-types to a new port-type.
6. Properly commenting the operations.

Guidelines for Service Discovery I

Goal

- Reduce effort needed for discovering external services

Actor

- Consumer

Summary

- Make queries representing consumers' needs
- Combine Query-By-Example and Query Expansion [2]
- 3 steps for effortlessly deriving queries from source code
- Tool support

Guidelines for Service Discovery II

1. Defining the expected interface of every application component that is planned not to be implemented but outsourced to a Web Service.
2. Revising the functional cohesion between the implemented (i.e. internal) components that directly invoke, and hence depend on the interfaces of, the components defined in step 1.
3. Naming and commenting each defined interface and internal component by using self-explanatory names and comments, respectively.

Guidelines for Service Consumption I

Goal

- Facilitate service-oriented applications maintenance

Actor

- Consumer

Summary

- Decouple client-side components from specific service APIs
- Treat service interfaces as crosscutting concerns [3]
- 3 steps for shielding consumers' apps. from service VARIABILITY
- Tool support

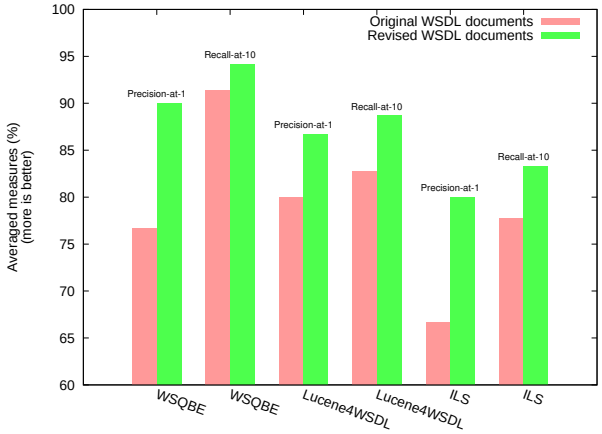
Guidelines for Service Consumption II

1. Defining the expected interface of every application component that is planned not to be implemented but outsourced to a Web Service.
2. Adapting the actual interface of a selected service to the interface that was originally expected, i.e. the one defined in the previous step.
3. Seamlessly injecting adaptation code into each internal component that depends on the expected interface.

Experiments

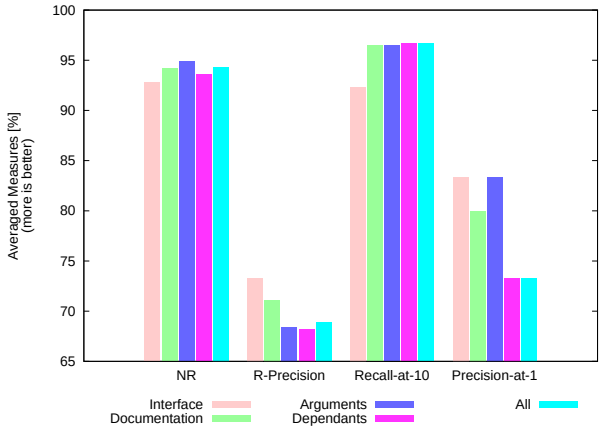
Publication Guidelines

- 392 WSDLs
- 8 heuristics
- avg. accuracy: 98.5%
- Impact on discovery:



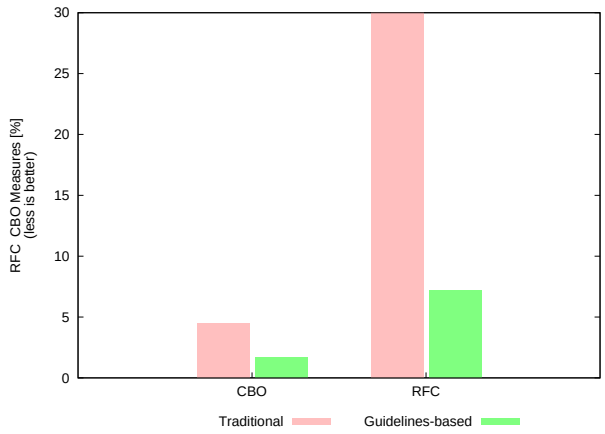
Discovery Guidelines

- 392 WSDL + 30 queries
- *R*-precision ~70%
- Recall-at-*10* ~94%
- Impact on ranking:



Consumption Guidelines

- The same app.
- Two versions
- Decouple apps. and services:

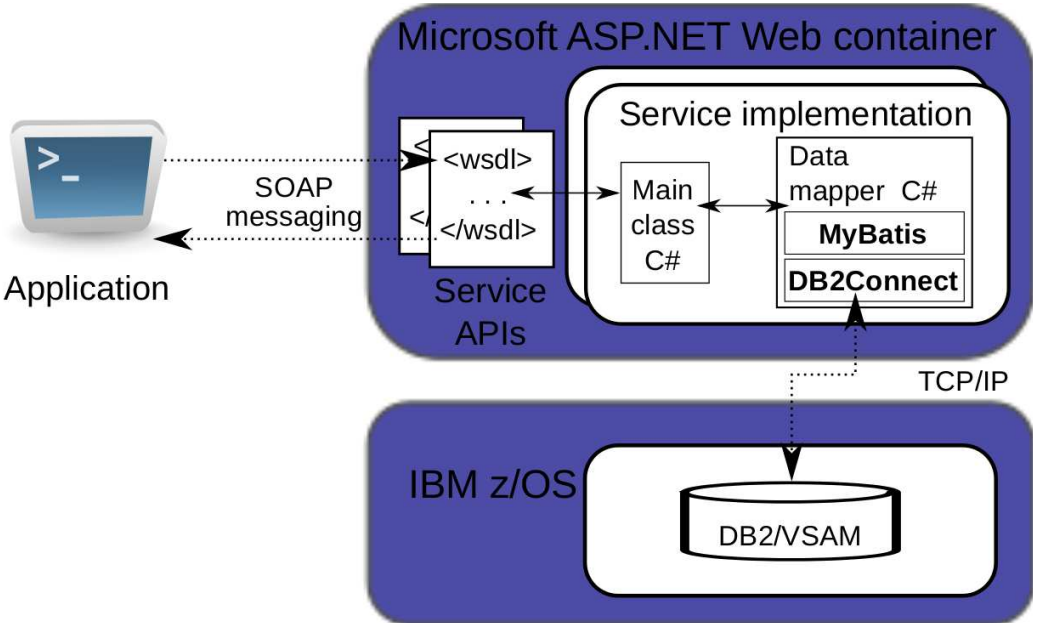


Conclusions

- SOC presents many intrinsic challenges that both Web Service providers and consumers must unavoidable face
- The proposed catalog covers proven best practices for Publishing, Discovering and Consuming Web Services

Future Work

- Recently, we have migrated a mainframe system to SOA
  - 35 years old CICS/COBOL
  - data-centric 0.8 Petabytes
  - resulting services in .NET



*We are specifying best practices for **migration**, which focus on the quality of the resulting service APIs*

For Further Reading I

- The EasySOC Project Home Page (Articles, Software prototypes, and more) `http://sites.google.com/site/easysoc/`

Juan Manuel Rodriguez, Marco Crasso, Alejandro Zunino, and Marcelo Campo.
Improving Web Service descriptions for effective service discovery.
Science of Computer Programming, 75(11):1001–1021, 2010.

Marco Crasso, Alejandro Zunino, and Marcelo Campo.
Combining query-by-example and query expansion for simplifying Web Service discovery.
Information Systems Frontiers, 2009.
To appear.

For Further Reading II


Cristian Mateos, Marco Crasso, Alejandro Zunino, and Marcelo Campo.
Separation of concerns in service-oriented applications based on pervasive design patterns.
In *Web Technology Track (WT) - 25th ACM Symposium on Applied Computing (SAC '10)*, pages 2509–2513. ACM Press, 2010.

Questions...

Thank you!