

# WS-Governance Tooling: SOA Governance Policies analysis and authoring

Antonio Manuel Gutierrez, Jose Antonio Parejo, Pablo Fernandez and Antonio Ruiz-Cortés  
School of Computer Engineering  
University of Seville  
Email: {amgutierrez,japarejo,pablofm,aruz}@us.es

**Abstract**—Governance is a capital issue in current Service Oriented Architectures, and governance policies are at its base. The governance policies definition must be supported by proper languages and tools, allowing for comfortable and collaborative editing, consistency checking and the evaluation policy meeting. In this paper we present a policy analyzer for WS-Governance (a governance policy definition language created by authors, described in [1]) together with an online editor and test suite with classical examples of WS-Governance Documents for consistency validation. Both the test model and the analysis tool prove the suitability of WS-Governance to define SOA governance policies.

## I. INTRODUCTION

SOA adoption brings an increase on the number of elements of the IT architecture, with proper management and control becoming capital issues. In this context, SOA Governance is defined as the management process aimed at delivering the SOA promise of reusing, business goals support and responsiveness. [2], [3].

Effective governance requires a formalization of the governance policy management, including : (i) the definition of policies that encode governance rules and (ii) the establishment of appropriate conformance testing and enforcement mechanisms for defined policies. Moreover, in [1] authors have identified through a case study the need to incorporate the structure of the organization as essential information to take into account when the governance policies are designed.

The current governance tools market is vendor-driven and turbulent, where tools are based on proprietary technology and their features are guided by the specific aspects where their vendors have expertise [4], [5]. A plethora of languages have been created for policy definition, some of them supporting web services as their subjects, specifically WS-Policy. However, this language has been mainly focused on the definition of policies related to the specific capabilities of individual services such as security, reliability, etc; through defining Domain Specific Languages that integrate on the general language framework. Unfortunately, describing assertions for SOA governance policies is a bit more complicated (see later) and, to the best of our knowledge, no single DSL has been defined to describe these kind of policies. This language, named WS-Governance, is proposed in [1]. This paper presents tools for the online editing and analysis of documents written using this language (named Governance Documents).

## II. GOVERNANCE DOCUMENT

A governance document in WS-Governance comprises three different sections:

- SOA Description; including Organizations, Applications, Services and their relationships.
- Domain vocabulary, that defines properties for the elements specified in the previous section (possibly including their values).
- Policies, that define assertions on properties for a set of elements of the SOA (determined by the policies' scope).

An automatic analyzer has been developed according to this model, to validate the correctness of the document, named Governance Document Analyzer (GDA).

### A. Governance Document Analyzer

GDA is a system which takes a Governance Document as input and generates a consistency report as output. The consistency analysis is done in three levels:

- Individual policy consistency; i. e. verifies that the policy has no contradictions impossible to meet.
- Consistency of the set of policies in the document; i. e. verifies that all policies could be met at the same time.
- Consistency of the whole Governance Document; i. e. verifies that the SOA and properties valuation described in the document meets all the policies.

This analysis starts with the transformation of GD statements to a Constraint Satisfaction Problem (CSP), as [1] proposes. The resultant document is solved with a CSP solver [6]. The tool has been developed as a web service so it can be easily integrated into other tools and interoperate. Furthermore, a command line client is provided for direct usage. It can be downloaded from <http://www.isa.us.es/gda>.

On top of this, a test suite of Governance Documents has been created, in order to ensure that the analyzer works properly. It contains sample documents with intrinsically inconsistent policies, mutually inconsistent policies and bigger groups of policies where only the whole set is consistent. Additionally, it contains documents where the modeled SOA violates policies.

## III. SYSTEM DEMONSTRATION

As result of the application of the model to the case study described in [1], a SOA and a domain vocabulary have been

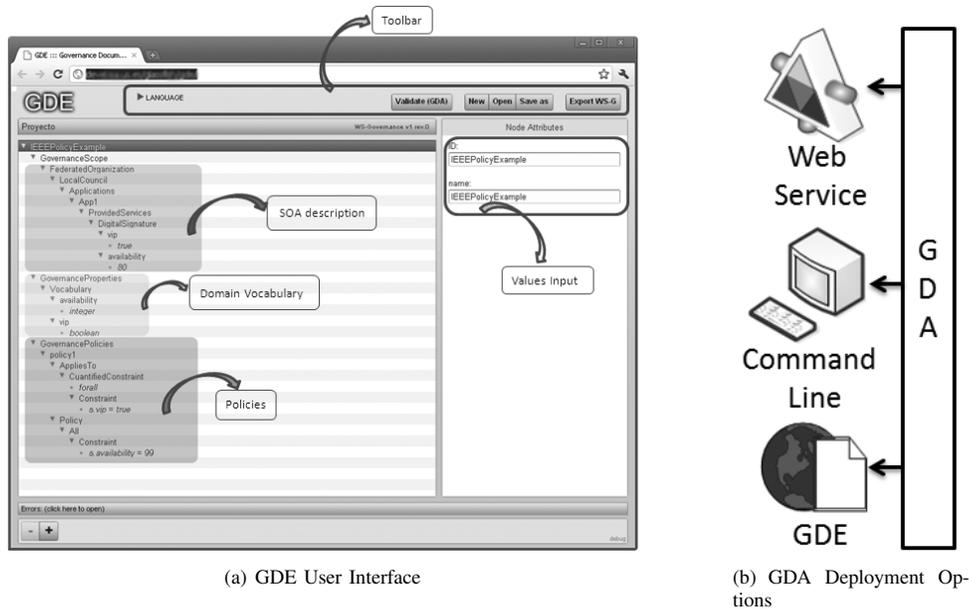


Fig. 1. GDE User Interface and GDA Deployment Options

obtained. The domain vocabulary is made up of two properties, *Availability* and *Importance*. Each service in the SOA has a value for those properties.

The demonstration system is made up of the analyzer and a test suite based on the case study. The test suite is composed of a set of governance documents with well known documented inconsistencies in the three levels (individual inconsistency, set of policies inconsistency and governance document inconsistency). Every sample in the test suite is defined together with the expected results.

#### A. Governance Document Editor

Governance Document Editor is a web application to edit government documents in an assisted way. The editor window is divided in three areas as in Figure 1 (a). On the top section, there are different options to open, save and analyse GDs. On the left side of the main area, every section in the GD is organized in a tree view. Every node on the tree is attributed on the right-hand side. The main tree has separate sections for each subsection in GD, that is:

- 1) SOA Governance Organization
- 2) Domain vocabulary, that is, their properties.
- 3) Governance policies

Language syntax elements are automatically generated from the tree model, so edition focuses on the relevant government contents, avoiding syntax errors. The editor is available on-line at <http://labs.isa.us.es/apps/gde/> and further documentation can be found in <http://www.isa.us.es/gda/>.

#### IV. CONTRIBUTION

The Governance Document Analyser provides an automatic validation of the consistency of GDs. This validation is impor-

tant for two reasons. Firstly, a real scenario test suite reinforces the suitability of the WS-Governance model for the definition of governance policies. Secondly, as the analyser has been developed with the inherent goal of interoperability, it detects consistency error simultaneously to defining policies (with the proposed editor or any other editor) providing an automated edition of, free of consistency errors, Governance Documents.

#### ACKNOWLEDGMENT

This work has been partially supported by the European Commission (FEDER) and Spanish Government under CICYT project SETI (TIN2009-07366), and Andalusian Government projects ISABEL (TIC-2533) and THEOS (TIC-5906) .

#### REFERENCES

- [1] J. A. Parejo, P. Fernández, and A. R. Cortés, “Ws-governance: A language for soa governance policies definition,” Applied Software Engineering Research Group. University of Seville, Available online at <http://www.isa.us.es/publications>, Tech. Rep., 2010.
- [2] E. A. Marks, *Service-Oriented Architecture Governance for the Services Driven Enterprise*. John Wiley & Sons, 2008.
- [3] M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann, “Service-oriented computing: State of the art and research challenges,” *IEEE Computer*, vol. 40, no. 11, pp. 38–45, November 2007.
- [4] L. F. Kenney and D. C. Plummer, “Magic quadrant for integrated soa governance technology sets,” Gartner RAS Core Research, Tech. Rep., March 2009, available at <http://mediaproducts.gartner.com/reprints/oracle/article65/article65.html>.
- [5] K. Kontogiannis, G. A. Lewis, and D. B. Smith, “A research agenda for service-oriented architecture,” in *SDSOA '08: Proceedings of the 2nd international workshop on Systems development in SOA environments*. New York, NY, USA: ACM, 2008, pp. 1–6.
- [6] F. Laburthe, N. Jussien, G. Rochart, H. Cambazard, C. Prud'homme, A. Malapert, and J. Menana, “Choco, java library for constraint satisfaction problems (csp), constraint programming (cp) and explanation-based constraint solving (e-cp).” One Source Software (BSD license) available online. [Online]. Available: <http://www.emn.fr/z-info/choco-solver/>