TaaSOR – Testbed-as-a-Service Ontology Repository

Milorad Tosic[1], Ivan Seskar[2], and Filip Jelenkovic[1]

[1] Univeristy of Nis, Faculty of Electronic Engineering, 18000 Nis, Serbia
[2] Winlab, Rutgers The State University of New Jersey, NJ, USA
mbtosic@acm.org, seskar@winlab.rutgers.edu,
filip.jelenkovic@gmail.com

Abstract. This demonstration is introducing Testbed-as-a-Service (TaaS) infrastructure that illustrates use of community based approach to building experimental ontologies and generation of supporting testbed resources applied to OMF based testbed. While this TaaS demo is initially primarily targeting virtualization and community collaboration, the final objective is to support domain specific experimental description languages and resource management in federated testbeds.

Keywords: Web Services, Ontology Repository, Semantic Web, Interoperability, Testbed.

1 Introduction

The research question addressed by the presented demo is: could we facilitate the Testbed-as-a-Service federated infrastructure by using existing XML based aggregate managers (or services), generating ontologies out of their XML descriptions, then using these ontologies, enable semantic service annotations and facilitate humanized interaction with experiment controller and resources supporting the range of what-if scenarios (including "what parameters may I change?", "do I break some constraints?", "give me range of the parameter") . These new services are potentially leading to an experiment specification language, ontology and even automated (OEDL/Ruby) code generation for OMF testbed management framework.

We identify *two crucial facilitators for achieving the vision of total system scalability in federated testbeds*: Experiment Virtualization, and Community Growth. We propose to exploit Semantic Technologies in general and Semantic Web in particular to develop the facilitators.

2 Community-Driven TaaSOR

TaaSOR (available at http://www.orbit-lab.org:8080/tasor) currently targets OMF framework [1], and is limited to internal use while being externally available as a proof-of-concept only. Here, we demonstrate the following: 1) Importing XML service description and generating the corresponding set of resources; 2) Browsing the set of

T. Korakis, M. Zink, and M. Ott (Eds.): TridentCom 2012, LNICST 44, pp. 419–420, 2012.

resources; 3) Loading new domain (NDL) and application specific ontologies; 4) Context and domain adaptation of the GUI; 5) Resolving semantic conflicts within the loaded services; 6) Semantic annotation of services; 7) Publishing the created semantics as an ontology as well as Rubi source code; 8) Command line access to the RESTful API.

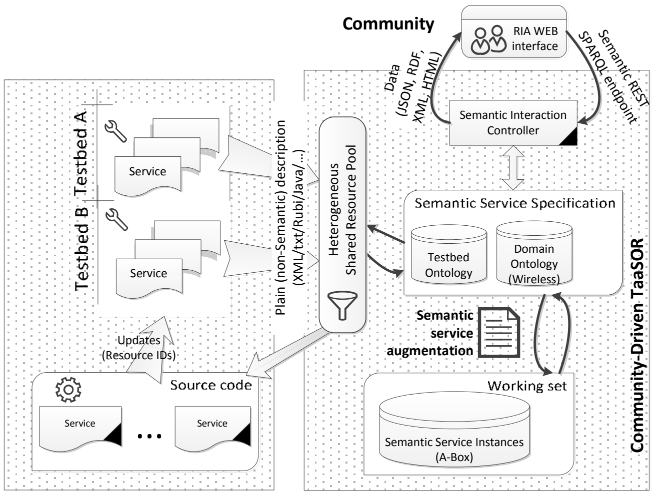

Fig. 1. Implementation architecture. Federation of services is built by importing plain service descriptions (XML or other) and transforming them into the corresponding set of RDF resources. The resources are further annotated by community using loaded domain (NDL) and application ontologies. The Working set developed that way may be published on the system's URL either as RDF ontology or Rubi code (a semantic positive feedback loop is created).

Fig. 2. RIA Web Interface: a) Drag-and-drop semantic statements builder with auto-complete. b) Hierarchical browsing of services' resources: All resources are clickable, draggable, and context sensitive. Knowledge pane presents convenient set of resources relevant for the last clicked resource. c) Working set of created semantic statements.

Reference

1. Rakotoarivelo, T., Ott, M., Jourjon, G., Seskar, I.: OMF: A Control and Management Framework for Networking Testbeds. ACM SIGOPS Operating Systems Review 43, 54–59 (2010)