Role of Semantics in Autonomic & Adaptive Web Services and Processes

Amit P. Sheth

Wright State University - Main Campus, amit@sc.edu

Follow this and additional works at: https://corescholar.libraries.wright.edu/knoesis

Part of the Bioinformatics Commons, Communication Technology and New Media Commons, Databases and Information Systems Commons, OS and Networks Commons, and the Science and Technology Studies Commons

Repository Citation
https://corescholar.libraries.wright.edu/knoesis/666

This Conference Proceeding is brought to you for free and open access by the The Ohio Center of Excellence in Knowledge-Enabled Computing (Kno.e.sis) at CORE Scholar. It has been accepted for inclusion in Kno.e.sis Publications by an authorized administrator of CORE Scholar. For more information, please contact library-corescholar@wright.edu.
Role of semantics in Autonomic & Adaptive Web Services and Processes

Amit Sheth, Kno.e.sis Center, Wright State University.

The emergence of Service Oriented Architectures (SOA) has created a new paradigm of loosely coupled distributed systems. In the METEOR-S project, we have studied the comprehensive role of semantics in all stages of the life cycle of service and process-including annotation, publication, discovery, interoperability/data mediation, and composition.

Among the concrete contributions, we offered a broad framework of semantics consisting of four types [1]:

1) Data semantics: Modeling of the inputs and the outputs of a service,
2) Functional semantics: Modeling of the functional capabilities of a service,
3) Non-Functional semantics: Modeling the non-functional requirements and capabilities including policy and/or agreement between services and
4) Execution semantics: Modeling service execution including interaction pattern and adaptation to failure.

The data semantics is essential towards addressing issues related to data heterogeneities, which are common in service compositions and service interactions [2]. Modeling functional semantics allows for better service discovery [3] and also helps in addressing issues in process composition. Non-functional semantics plays a very important role in partner selection based on requirements and guarantees. In a distributed ecosystem like SOA, it is very likely that services in different domains often want to communicate with each other. It is in this context, modeling of non-functional semantics, via mechanisms like WS_Agreement [4] or WS-Policy [5], allows for better matching and selection of partners. The SAWSDL candidate recommendation, which originally started as WSDL-S, enhances the current WSDL standard to model the data and functional semantics using WSDL.

We recognized the need for an incremental and evolutionary approach to add semantics to SOA that is consistent with W3C recommendations and industry's investment in tools and skills in Web Services related technologies. This led to our proposal for WSDL-S, which was refined in collaboration with IBM and submitted to W3C [6]. W3 Working group (of which we are a member) adopted key aspects of WSDL-S and has defined the Semantic Annotation of WSDL (SAWSDL) specification, which is now a W3C candidate recommendation. Significant tools and use cases for SAWSDL already exist [7].
Lately we have focused on the using semantics to add dynamism and adaptability to services and processes in SOA-based systems. This led to the proposal of Autonomic Web processes presented as a vision talk at ICSOC [8]. Here we propose a self-healing, self-optimizing framework for SOA. Modeling execution semantics is critical towards realizing that vision. Execution semantics can be used to specify, what a system should do in the event of failure doing execution. In the METEOR-S project, this led to extending the notion of task skeleton, first proposed in [9], towards creating an adaptation framework for Web processes.

This talk would expand on the short motivation towards the need for the four types of semantics, its standards-based support through WSDL-S/SAWSDL, and the efforts towards realizing dynamic and adaptive SOA.

References/Additional Reading:


