Semantic Layered Architecture to Integrate FR/NFR in Software Performance Engineering

Isaac Lera and Ramon Puigjaner
University of the Balearic Islands
Departament de Matemàtiques i Informàtica
Palma, Spain 07122
{isaac.lera, putxi}@uib.es

1. ABSTRACT

The integration process using UML SPT or MARTE specification presents some deficiencies due to the inability to formalize the representation criteria and to relate information in a transparent way between the outputs of performance models and software designs. The drawbacks of FR/NFR model integration is crucial for the real application of several SPE techniques, since the performance modelling and the software modelling are not semantically related because there are not explicit and formal semantic interpretation of the concepts. To avoid this, we propose a semantic architecture based on well-known Web Semantic Layered Architecture

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Modeling techniques; D.3.1 [**Formal Definitions and Theory**]: Semantics

General Terms

Performance

Keywords

Semantic Modeling, Requirements, Performance

2. SEMANTIC ARCHITECTURE

Re-utilization, interoperability, automation and knowledge design understanding of multiple related works may present an expensive cost. We can establish that several number of NFR integration problems are due to transformations between different representation languages and partial specifications. Thus, performance aspects are defined in a separate model where each aspect is related to SA specifications. The performance model is complemented with SA parameters using specific tools and then it is transformed in performance evaluation models with other specific tools. Performance evaluation output offers to SA designer suggestions to customize the software architecture. These performance suggestions has to be interpreted by SA designer because specific tools are usually not bidirectional. All these stages may depend on personal designer criteria.

The use of syntactical terms for SPT annotations, e.g. $PADemand = ('est', 'mean', 0, 5, 'ms')$, has to resort to manual interpretation to understandable. If SA designers fail to recognize basic performance assumptions, they never understands the meaning of performance annotations and also do not correlate the performance annotations and ulterior performance evaluation values (suggestions), which it is crucial for SPE success. The future SPE integration tools should need an intelligent supervision to manage the context of design and to introduce sense (meaning) in the evaluation process. Thus, we propose the use of a formal and semantic language to define the SA and FR/NFR together. This representation uses Ontology Web Language (OWL) to represent Software Architectures (SA) but focusing on represent non-functional requirements, specially, performance requirements.

W3C proposes an stack layer architecture (see the left part of fig. 1). Each layer has an specific goal to achieve a formal logic representation shared across the web: URI and UNICODE layers define the vocabulary and the identification of resources through hyperlinks. RDF model consolidates the baseline with hierarchical and properties relationships. The ontology layer improves the semantic including the descriptive logics formalism. The next group of layers defines specific knowledge of the representation domain. The rules are based on first order logics. The logic layer provides functionality on data. The penultimate layer, provide of explicit and implicit facts. To sump up, the last layer gives representation credibility according to external criteria. To conclude, the two vertical layers guarantee security aspects to manage the authors or entities that define the whole information and data encryption. We are able to integrate FR/NFR in the same software platform (see the right part of fig. 1). Our SPE representation relies on the group of layer which imply semantics definitions in the SW (the three intermediate layers of white color). This implies modifying the management of these data with new tools, new protocols, new diagrams, new interactions, etc. On this semantic representation, we define several types of NFRs. Each requirement is put into more general groups depending to its areas. Each area represents a specific domain, with its own rules, logics, and facts. Thus, each area has its own three layers of the SW (rules, logic, and proof) where they characterize the knowledge and the operational work of their specific domains. For example, in the fig. 1 (right part), we have defined three areas where are grouped several NFRs about: performance, usability, and security. In the case of usability, we could know whether the group of NFRs defined comply with some ISO standards.

Advantages of use ontologies to represent FR and NFR are evident facts: (i) Formal Representation. With OWL there are more possibilities to mix operators based on Descriptive Logics. It is not necessary to use annotations with specific and not general known syntax. (ii) Trustability and

WOSP/SIPEW'10, January 28–30, 2010, San Jose, California, USA.
ACM 978-1-60558-563-5/10/01.

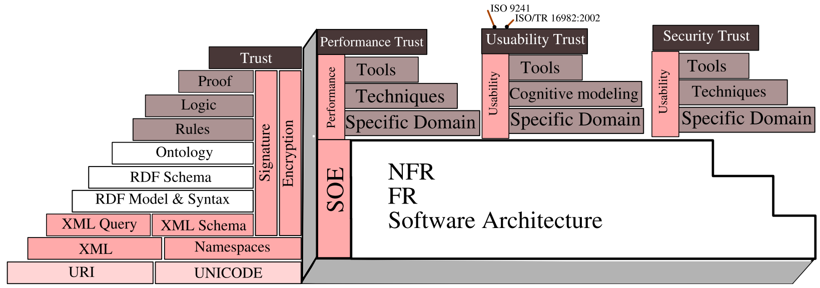

Figure 1: NFR & FR design using Semantic Web Layered Architecture

consistency. OWL constructors restrict and delimit concept interpretation, giving opportunity to the ontology editor to prevent the definition of contradictory facts during ontology life-cycle. On the contrary, UML needs upper tools to provide this feedback. (iii) Term disambiguation. Concepts are defined with restricted constructors and they are connected with other concepts of the same context. (iv) Collaborative definition of terms. Collaborative work can be realized to define repositories with consensus data, as Wikipedia. To have this well and formal knowledge could improve several development scenarios. (v) Reutilization of terms (scalability / mapping). Due to a formal representation the ontology ambiguity is reduced, then we can take some part of domain to use in other specific contexts. Modify these partial representation for our propose requires less time than use other representation where the syntax is unknown, and it is stored across multiple sources. (vi) Formal process is able to obtain output to feedback directly to SA designer without interpretations. Tools can operate with semantic representation without third part software to transform syntax which can originate lost data. (vii) Direct Reasoning on Data. Multiples reasoner engines can work directly with data to provide implicit information.

3. OWL-BASED PERFORMANCE MODELS

Starting from a baseline representation as SPT, we have translated their terms into ontology representation using only basic OWL constructors. Once defined the ontology that represent and extend the SPT profile, the next phase is the representation of units. We have developed an ontology that define metrics using others ontologies (reutilization) which represent time units [1].

In order to clarify the ontology construction and inference processes let's introduce through an example. We define a service time (`DeviceServiceTime`). With UML annotations an instantiation of this service time should be: `PAService=('assm','dist',("constant",'0.012,'ms').`
Instead, using our ontology, that service time is defined as a fact `(owl:Individual)`, being one type of *UnitDerivedByScaling* with properties: *derivedFromUnit second* and data property: *hasValue* with value *0.012*. Furthermore, an inference engine infers the next facts: (i) Since it is derived from *second* unit then it is a unit of kind *Time*. (ii) It is a derived unit from other basic unit. (iii) As it is a kind of second unit then it is also basic unit, and has a special symbol of representation: "s". (iv) It has been set a scaled metric, which is related with other facts derived from *second*, due to it is a derived unit. e.g, in the ontology there is a fact called *hour* which has a scale value of 3600 per second.

[1]http://sweet.jpl.nasa.gov/1.1/time.owl

Another way to represent domain information is using logic rules which are represented through Semantic Web Rule Language. Rules are perfectly integrated within the ontology structure, where they relate axioms (complex definitions) through antecedent and consequent forms, e.g. we can set a quality aspect of web system to validate its response time which is a non functional requirement. Then, we may define a quality service indicator that is triggered when its value exceeds some threshold. Other task in our architecture is to represent some performance technique. We have defined an ontology that describes Stochastic Petri-Nets extending the previous work of Gaŝević et al and their tool called P3 [2] which represent PNs. Integrating both domains: SPN and our performance ontology we can describe SA process through SPN. With P3 tool we interact graphically with both ontologies. The tools show the system behaviour but also offer information about SLA compliance or any other agreement. The validation is done in the last layer: trust layer. Its decision is based on the facts generated by tools, which obtain results from performance model (logic perform) and from specific rules and facts about domain.

4. CONCLUSIONS

To sump up, We have provided an example to illustrate NFR performance definitions according to our architecture. We define a semantic performance model based on the syntactic representation of UML-SPT profile, we add information reusing ontologies, new terms, and creating new relationships among new and existing data. Finally, we also illustrate the case on which importing Petri Nets ontologies and building new relations are possible. We have used a third-party tool to visualize the Petri Net represented. Thus, our example shown that it is feasible to integrate NFR and FR in a single representation by exploiting the idea of SW. As future work, we have to analyse the feasibility of building a web portal where other researchers may collaborate developing ontologies to represent NFR. In particular, we are going to build a P3-based tool capable to support SPN.

5. ACKNOWLEDGEMENTS

This work is supported by the project *TIN2007-60440* from Spanish Ministry of Education and Science.

[2]http://afrodita.rcub.bg.ac.yu/∼gasevic/projects/PNO/OWL/PNO-core.owl