WATERFORD INSTITUTE OF TECHNOLOGY

Keara Barrett

Technical Report CMP-001-2006
June 2006

Supervisors:
Dr. William Donnelly
Dr. Sven van der Meer
John Strassner

ONTOLOGY-BASED POLICY RULE SPECIFICATION AND INTEGRATION

Keara Barrett

Telecommunications Software and Systems Group (TSSG),
WIT, Cork Road, Waterford, Ireland

Abstract: A formal transformation between knowledge contained in Operations Support Systems (OSS) views is required to automate the deployment of OSS. This paper details progress towards the integration of policy languages at the business view of the TeleManagement Forum (TMF) Next Generation Operations Support System (NGOSS), with the specification of a formal language for the TMF's Shared Information and Data Policy Aggregated Business Entities.

Keywords: Autonomic Communications Management, Policy Specification, Policy Integration and Ontology

1. INTRODUCTION

The emergence and ongoing development of converged communication networks with heterogeneous access and core network types and diverse device technologies has led to an increase in network management cost and complexity. This, among other factors, has prompted changes in Operational Support Systems (OSS) towards more holistic lifecycles with integrated processes, information models and languages between different views; be it, for example, the Business, System, Implementation and Deployment views of the TMF New Generation Operations System and Software (NGOSSTM) [1]. However, while integration of knowledge between lifecycle views ensures a closer bi-directional correlation among business requirements and the altering state of the communications network being managed, substantial work remains to achieve the goal of enabling the integration of loosely coupled and distributed components that compose the OSS.

Network management complexity has also prompted the emergence of autonomic communication management, with its proverbial self-configuring, self-healing, self-optimising and self-protecting concepts. Incorporating autonomic principles into OSS further complicates integration with the prerequisite for dynamic transformation and mappings between the views' information models and the languages applied at each view for specifying policy rules. Additional challenges also arise when interfacing between and merging businesses with distinct methodologies; integration and interoperability of distinct information models, processes and languages at the same view but between different businesses or even between different communities within one business must be considered.

Inter-view and intra-view transformation between the knowledge contained in OSS views must be accomplished to provide an autonomic holistic OSS. In this paper we detail progress towards the dynamic integration of policy languages at the business view of the OSS, with the specification of a formal language for the TMF's Shared Information/Data Model (SID) Policy Aggregate Business Entities (ABE) [2].

2. FORMAL LANGUAGE FOR SID POLICY MODEL

Policy plays an imperative role in an OSS as it formalises the concept of decision making, indicating that policy is specified at all of the OSS views. While the policies at each view may at first appear disparate they must be resourcefully linked, particularly for autonomic holistic management. Hence the notion of a "single" policy is limited. John Strassner has identified this limitation and has defined the Policy

Continuum to highlight the concern of associating policies at different views [3]. Each view of the Policy Continuum respects different constituencies within an organisation and has a link to one or more views of the TMF NGOSS; the views of the Continuum and NGOSS are slightly different as they address different concerns. However, the Policy Continuum together with the TMF SID policy model do not currently define a process for linking, statically or dynamically, the semantics of policy defined at each level.

The TMF SID policy model provides a representation of policy independent of the content. It defines policy as a "set of rules that are used to manage and control the changing and/or maintaining of the state of one or more managed objects." These rules, depicted as a UML class diagram in Fig 2-1, are containers for (1) Metadata, (2) Events that trigger the evaluation of a condition, (3) Conditions that must hold true for actions to be executed and (4) Actions that are executed on managed objects when events specified in the policy rule trigger and some or all conditions hold true. To allow policy defined based on the SID policy model to integrate with each other and also with policy defined with other policy models (i.e. policy refinement) a formal specification of the SID model is necessary. An ontology is an obvious option to represent this knowledge as it provides a means to formally specify the semantics of concepts and the relationship between these concepts and can, thus, be used to augment information in the policy models with additional meaning and relationships.

The ontological representation of the SID policy model (or formal SID policy language) briefly outline in this paper was specified with the Protégé-OWL plugin, the leading editor for Web Ontology Language (OWL). OWL is a standard developed by the W3C provides three sublanguages, OWL-Lite, OWL-DL and OWL-Full; OWL-DL is computationally complete whereas OWL-Full is fully expressive and therefore tractability can not be guaranteed. The mapping between the UML specification and the OWL representation was achieved manually; this was a time-consuming process but allowed for an accurate representation to be built and an in-depth knowledge of the SID policy model to be gained. An alternative approach would involve exporting from the UML files to XMI, importing the Ontology-based Policy Rule Specification and Integration XMI to a purpose made tool that would provide output in OWL format. The canonical UMLtoOWL tool designed by Dragan Gašević was was not applicable as it converts from Ontology UML Profile (OUP) models in XML Metadata Interchange (XMI) format to OWL and not from UML itself [4]. Falkovych et al have delineated transformation approaches and discuss ways to handle the conceptual differences between the languages in [5].

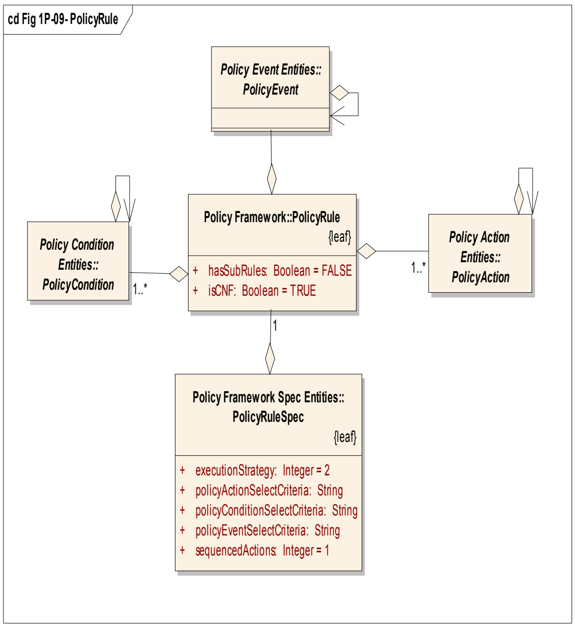

Fig. 2-1. Basic view of the SID Policy Rule Specification

When defining the OWL representation, the UML packages were defined as sub-ontologies and imported into the **SIDPolicyABE.owl**. UML classes mapped directly to an **owl:Class** as they both describe objects and basic types. In UML an attribute is a description of a specified type in a class, in OWL attributes are first class entities. The most appropriate mapping was to define UML attributes as **owl:DatatypeProperty**. However, DIG reasoners, RacerPro, Pellet and FaCT++, were not able to reason over the xsd datatypes, consequently the attributes were defined as **owl:ObjectProperty** with the UML class in which they were specified as an **rdfs:domain** axiom and the appropriate datatype as an **rdfs:range** axiom. The issue with this approach is that **rdfs:domain** and **rdfs:range** constructs are not meant to be viewed as constraints to be checked but rather axioms in reasoning. UML Associations were mapped to **owl:ObjectProperty** with the source class as the **rdfs:domain** and the target class as the **rdfs:range**. Bi-directional associations were represented as two **owl:ObjectProperty** declared as inverse with the **owl:inverseOf** construct. Association multiplicity was handled with Restrictions (**owl:Restriction**), predominately cardinality restrictions. A snapshot of the Protégé OWL representation of the SID Policy Action and Policy Condition are shown in Fig 2-2 and Fig 2-3 respectively.

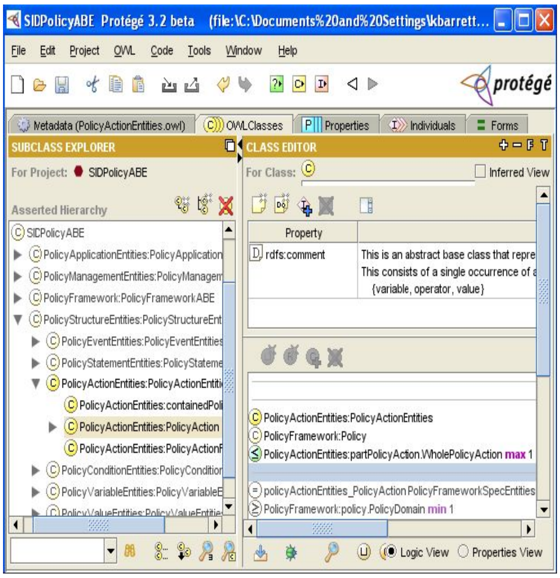

Fig. 2-2. Protégé OWL representation of the SID Policy Action

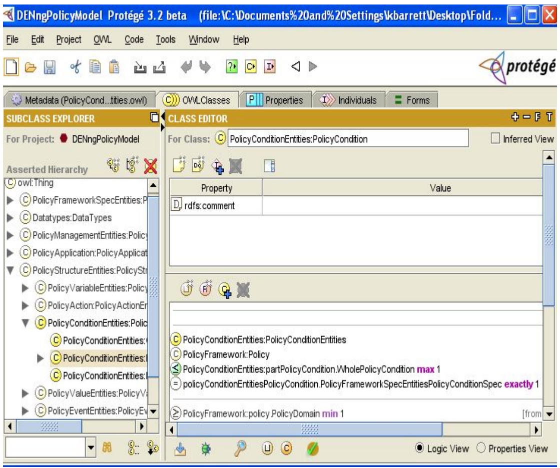

Fig 2-3 Protégé OWL representation of the SID Policy Condition

The following is a snippet of the Resource Description Framework (RDF) code for the SID Policy Condition:

```
<owl:Class
rdf:ID="ContainedPolicyConditionDetails">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty
rdf:resource="#containedPolicyConditionDetails.PolicyCondition"/>
        <owl:cardinality
rdf:datatype="&xsd;int">1</owl:cardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf
rdf:resource="#PolicyConditionEntities"/>
  </owl:Class>
  <owl:ObjectProperty
rdf:ID="containedPolicyConditionDetails.PolicyCondition">
    <rdfs:domain
rdf:resource="#ContainedPolicyConditionDetails"/>
    <rdfs:range rdf:resource="#PolicyCondition"/>
  </owl:ObjectProperty>
<owl:ObjectProperty
rdf:ID="containedPolicyConditionDetailsAttributeContainedConditionGroupNumber">
    <rdfs:domain
rdf:resource="#ContainedPolicyConditionDetails"/>
    <rdfs:range
rdf:resource="&Datatypes;Integer"/>
    <rdfs:subPropertyOf
rdf:resource="#containedPolicyConditionDetailsAttributes"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty
rdf:ID="containedPolicyConditionDetailsAttributeContainedConditionIsNegated">
    <rdfs:domain
rdf:resource="#ContainedPolicyConditionDetails"/>
    <rdfs:range
rdf:resource="&Datatypes;Boolean"/>
    <rdfs:subPropertyOf
rdf:resource="#containedPolicyConditionDetailsAttributes"/>
  </owl:ObjectProperty>
```

3. CONCLUSION AND FUTURE WORK

OWL facilitates interoperability with logical equivalences and other formal relationships between classes and properties in different ontologies. Exploiting OWL to represent policy models, such as the SID policy model, provides a framework to achieve semantic interoperability between policies specified with different languages and separate models at disparate levels of the OSS lifecycle, where the **owl:subclassOf** construct will most likely feature heavily in the integration between views. This interoperability will never be fully automated but transitive mappings will eliminate much work. The aim now is to define integrations between existing OWL policy representations to achieve policy refinement.

REFERENCES

[1] TeleManagement Forum: "GB927: The NGOSSTM Lifecycle and Methodology", November 2004.

[2] TeleManagement Forum: "GB922 Addendum - 1-POL SID Common Business Entity Definitions - Policy v2.0", April 2006

[3] Strassner J.: "Policy-based Network Management: Solutions for the Next Generation".
Morgan-Kaufman Publishers. ISBN 1-55-859-1, September 2003

[4] Gašević, D.: "UMLtoOWL: Converter from UML to OWL". can be found at http: //afrodita.rcub.bg.ac.yu/_gasevic/projects/UMLtoOWL/#

[5] Falkovych K., Sabou M., Stuckenschmidt H.:

"UML for the Semantic Web: Transformation-Based Approaches". In Proceeding of Knowledge Transformation for the Semantic Web, pages 92–106. IOS Press, 2003.