eChallenges e-2009 Conference Proceedings
Paul Cunningham and Miriam Cunningham (Eds)
IIMC International Information Management Corporation, 2009
ISBN: 978-1-905824-13-7

A Modelling and Reasoning Framework for Representing and Orchestrating Service Level Agreement Behaviour

John STRASSNER[1,2], Miguel PONCE DE LEON[1]
[1]*Telecommunications Software & Systems Group, Waterford, Ireland*
Tel: +353 51 302910, Fax: + 353 51 341100, Email: {jstrassner, miguelpdl}@tssg.org
[2]*Pohang University of Science and Technology, Pohang, Korea*
Tel: +82 54 279 5605, Fax: + 82 54 279 5699, Email: johns@postech.ac.kr

Abstract: This paper describes a new electronic Service Level Agreement model that is part of the DEN-ng information model. A new knowledge representation method is used to extract knowledge from the model and associate it with knowledge from ontologies, enabling a system to *reason* about how to best orchestrate behaviour to meet the terms and conditions of the Service Level Agreement, even if business, user, and/or environmental conditions change. Such intelligent automation can save significant operational expenditures and ensure that customers receive the services that they have contracted for, thereby reducing customer churn.

1. Introduction and Objectives

The recent economic downturn has emphasised the importance of being able to use business objectives and processes to determine the set of network resources and services that should be offered. This trend will be continued in work on next generation networks and on the future Internet.

Current network management and operational data does not contain business or system information, making management difficult. A Service Level Agreement (SLA) uses business terminology, and among other things defines contractual obligations that produce revenue or result in cost penalties. In stark contrast, such concepts and terminology are not present in network operational and administrative data. Rather, network engineers work in terms of classifying and conditioning traffic. This gulf in terminology and concepts is exacerbated by the different vendor-specific programming languages and models that each use different syntax and semantics. This inhibits humans from associating business terms, such as revenue, with complex network terms, such as traffic queuing and scheduling. Worse, there is a lack of tools to support this process, which inhibits its scalability and results in hand-woven processes that are extremely hard to automate, thereby increasing Operational Expenditures (OPEX).

This paper focuses on the electronic representation of SLAs and the autonomic orchestration of behaviour driven by SLAs that affects both operational and business support systems. Our approach enables behaviour to be defined by different constituencies, such as business, network, and programming personnel. For example, business policies that maximise service revenue can be translated to a form that enables network architects to implement policies to reconfigure network devices that provide those services. In order to realise translations such as this, a new electronic SLA representation is required. We use the DEN-ng object-oriented information model [1] to represent a system of interest; this enables us to build relationships between SLAs, entities that the SLAs affect, and people that author and implement the SLAs. We augment DEN-ng with the DENON-ng ontology

[2]. DEN-ng model elements, such as classes, properties, and associations, define *facts*; DENON-ng elements, such as concepts and relationships, enable us to add *meaning* to facts and *infer* answers. The combination of the model and ontology forms a *modelling and reasoning framework* for representing SLA requirements and actions to take that ensure network resources and services are compliant with the SLA. The overall result is to proactively increase revenue and decrease penalty costs associated with violating SLAs.

2. Methodology

The FOCALE [13] autonomic architecture has been widely published and implemented. The DEN-ng information model is currently being standardised in the Autonomic Communications Forum; previous versions of it have already been standardised in the TeleManagement Forum and in the ITU-T. The DENON-ng ontologies are currently under development. The knowledge representation framework is mature, and will also be standardised by the Autonomic Communications Forum.

We seek to use this assembly to attract new entities to the Autonomic Communications Forum (ACF), where much of this initial work is being done, and to related future FP7 ICT proposals. Our methodology includes interoperability testing in the ACF along with other groups; use of FIRE-compliant testbeds; prototyping in ongoing FP7 projects; and full implementation in a new, larger FP7 IP.

3. Technology Description

This section provides a description of our technical approach.

3.1 Comparing Information Models

An information model is a structured representation of data independent of platform, language and protocol; a data model is tightly bound to platform, language, and/or protocol [3]. A data model is built from an information model, and is used to define how data is structured and accessed for implementation purposes. The three main models used in network management today are the Autonomic Communications Forum DEN-ng [1], the TeleManagement Forum Shared Information and Data Model (SID) [4], and the Distributed Management Task Force Common Information Model (CIM) [5]. The SID is partially derived from DEN-ng v3.5; the latest version of DEN-ng is 6.6.5.9. The CIM is not related to either DEN-ng or the SID. The TMF has had a five-year liaison relationship with the DMTF, trying to align the CIM and SID models. This work has been unsuccessful because of two important problems. First, the CIM uses its own proprietary metamodel, not that of UML [6]. This means that the concepts used to build CIM models (e.g., classes, attributes, and relationships) are defined differently from the corresponding UML model concepts, whereas both DEN-ng and the SID use UML. In addition, the CIM is in reality a data model, as it contains technology-specific concepts, such as keys and weak relationships, which are not technology neutral. Second, patterns play a critical role in model-driven design, enabling the reuse of successful designs and helping to make models simpler and easier to learn. CIM does not use patterns; SID uses 4; DEN-ng uses many. In addition, CIM does not use roles [7], which both DEN-ng and the SID do. Roles make a design inherently scalable by abstracting individual users, devices, and services into roles that can be played by various managed entities. In DEN-ng and some of the SID, roles are not limited to just people; rather, they may represent resources, services, products, locations, and other managed entities of interest.

Note that there is a significant difference between the design of the DEN-ng and SID models. The top portion of the inheritance tree of the DEN-ng model was completely redesigned in version 5.5 and enhanced again in 6.0, while the root of the SID model has stayed relatively the same (since DEN-ng version 3.5). The redesign of DEN-ng was prompted to (1) enable simpler interworking with ontologies, (2) to correct some semantic ambiguities, and (3) to introduce the MetaData and Value hierarchies (both are unique to DEN-ng). This is explained briefly in the next section and more fully in [8].

3.2 Important Features of DEN-ng

The top-level hierarchy of DEN-ng is shown in Figure 1, and consists of the RootEntity class (which is not shown) and its three subclasses. The Entity class represents classes of objects that model important characteristics and behaviour of the environment being managed. An Entity represents objects that have a separate and distinct existence. DEN-ng is designed to work with ontologies and machine-based learning and reasoning. As such, it models facts that are observed or measured differently than facts that are inferred; this is unique to DEN-ng, and enables knowledge from DEN-ng to be combined with knowledge from ontologies. A Value is an abstract class whose subclasses are used to reify the notion of something that exists but does not have a distinct associated identity (like subclasses from Entity do). Finally, the MetaData class is defined as information that describes an Entity or a Value. Metadata may include descriptive information about the context, quality and condition, or characteristics of the data. Note that the CIM and SID do not define metadata, nor do they have a flexible means to associate data (values) with entities.

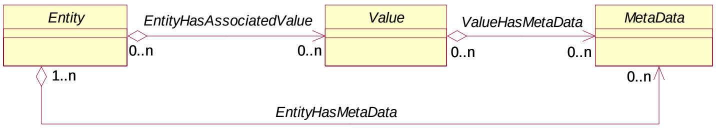

Figure 1. Simplified Top-Level DEN-ng Hierarchy

DEN-ng uses classification theory [9], whereas the SID and CIM do not. This is evidenced by the three well structured class hierarchies of DEN-ng (i.e., all classes are subclassed from the three classes shown in Figure 1). In contrast, the CIM and SID both have a large number of classes that are subclassed directly from their top level class as well as a large number of class hierarchies. This makes those models significantly more difficult to learn and understand than DEN-ng, even though DEN-ng has significantly more classes.

The main DEN-ng classes of the Entity hierarchy are shown in Figure 2. An Entity is refined into three types of subclasses: ManagedEntity, UnManagedEntity, and Event. A ManagedEntity is something that is manageable and belongs to at least one Management Domain, and is governed by at least one Management Application. A ManagedEntity is made up of four aspects and an identity, enabling it to be modelled in whole or in part to suit the needs of the application using it. The BehaviouralAspect class represents statically defined behaviour, such as behaviour that is governed by policy rules or applicable to a specific context or domain. The OrchestrationAspect class represents the relationship between this entity and one or more State Machines that are used to orchestrate its behaviour. The ProducerConsumerAspect represents manageable entities that are produced and/or consumed by other entities (e.g., Products, Resources, and Services). The PersonOrOrganization class represents people and organisations. Finally, Identity enables the system to unambiguously identity a ManagedEntity.

All MeasurableValues can be observed, but not all ObservableValues can be measured. For example, the number of packets dropped in a given time period can be measured. In contrast, a user reporting that his or her cell phone signal is “not as good” as before is a datum that has no underlying measurement basis – it is simply a subjective input. The main classes of the DEN-ng MetaData Hierarchy are Role, different types of entity-specific MetaData subclasses (e.g., EntityMetaData, ValueMetaData, and SemanticMetaData) to capture different types of metadata, and Version (for life cycle control). Role is an implementation of the role-object pattern [7]; this separates the entity from the function that it is performing by defining separate role objects that are dynamically attached to and removed from the core object. The resulting aggregate object represents one logical object, even though it consists of several physically distinct objects. It is important to note that the overall design of DEN-ng enables libraries of reusable models to be built by flexibly attaching different concepts (e.g., metadata) to entities.

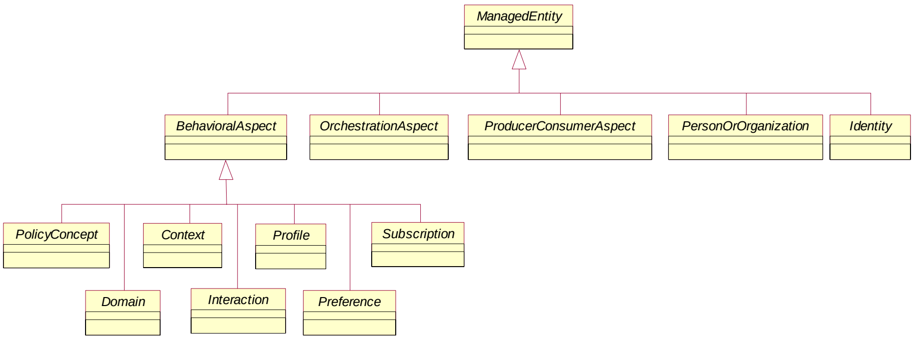

Figure 2. Main Class Hierarchies in the Entity Hierarchy of the DEN-ng Model

3.3 Important Features of DENON-ng

The purpose of an ontology is to represent, using a formal language, a set of concepts within a domain and the relationships between those concepts, so that their meaning is well defined and constraints on their use are clear [10]. The design of the DENON-ng ontology is based on the fact that there is no one ontology that can best answer queries from multiple domains. Therefore, DENON-ng is made up of a set of different building blocks that can be combined to form a generic upper-level ontology. The top layer has been designed to augment the main model elements (e.g., classes and associations) of DEN-ng. This layer serves as the root for building domain-specific ontologies, as shown in Figure 3.

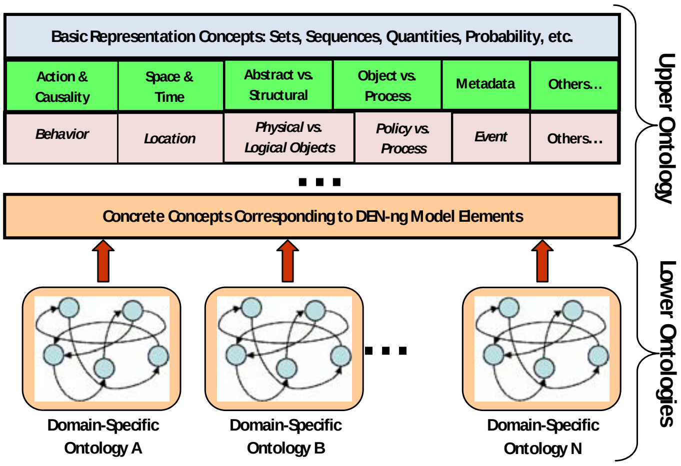

Figure 3. Conceptual DENON-ng Ontology Design

DENON-ng is populated, queried, and managed using distributed agents. Each agent is specific to a particular type of knowledge, and hence is optimised to process a particular type of knowledge. We use ontologies for two purposes: (1) to assign consensual definitions and meaning to model elements in the DEN-ng model, and (2) to reason about model elements. Both of these elements depend on our knowledge representation.

3.4 Our Knowledge Representation

DENON-ng ontologies are federated by a set of linguistic and semantic relationships that link each top node of each domain to each other; other lower-level nodes of a domain can of course also be linked to other domain nodes. We use linguistic matching, which relies on semantic relationships, to associate input data with our reference data, so that we can infer meaning and semantics that may not be explicitly present. Semantic relationships include "standard" relationships, such as synonyms, hyponyms, and meronyms, as well as "custom" relationships, such as semantic relatedness [11]. We use WordNet [12], which provides a set of APIs for computing common linguistic relations, to construct a multi-graph by building a set of semantic edges that associate one or more nodes in a graph representing the DEN-ng model to one or more nodes in a graph representing the DENON-ng ontology. A semantic edge is an edge of a graph that is formed because of the semantic and/or linguistic similarity (or dissimilarity) between the nodes that it connects, such as two nodes that have synonymous definitions. A simplified version of the process is shown in Figure 4.

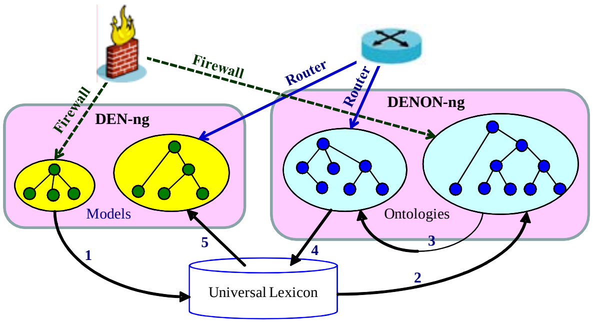

Figure 4. DENON-ng Ontology for Context

In step 1, a lexicon is used to relate data describing the firewall to a consensual set of terms that are then used to map to concepts in the ontology; this augments modelled facts with meanings from the ontology in step 2 using one or more linguistic relationships, such as synonyms. We also define a set of custom relationships, such as "is similar to", to perform specialised mappings. An example of this type of mapping is to define the semantic similarity between a set of m commands from one vendor with a set of n commands from a different vendor. Step 3 uses semantic relationships to relate a matched concept to similar concepts in other ontologies. Step 4 relates discovered concepts from the ontologies to consensual terms in the lexicon, enabling step 5 to relate these terms back to terms in the information model of the router. For example, the verb "accept" can be defined as both a synonym of the verb "forward" as well as an antonym of the verb "drop". Hence, the system now knows that the verb "accept" in the firewall language performs the same function as the verb "forward" in the router language (and also that it performs the opposite function of the verb "drop" in the router language). The process iterates as necessary, and produces XML documents that are annotated with semantic information.

3.5 The DEN-ng SLA Model

A simplified version of the DEN-ng SLA is shown in Figure 5. DEN-ng categorises Services into *customer-facing* vs. *resource-facing*; the former are directly visible and usable by Customers, whereas the latter are not – they are instead internal to the operation of the network. For example, a VPN is a customer-facing service, whereas the forwarding (MPLS) and route advertisement (BGP) services that it uses are resource-facing services. Both CustomerFacingServices as well as ResourceFacingServices can be defined as stand-alone (atomic) or aggregated (composite) entities. ServicePackage is a customisable grouping of services into a Product that can be offered to Customers. For example, Silver Service might define a grouping of VoIP, Internet browsing, and backup services, while Gold provides faster download speeds, more storage, and additional applications, like streaming video. Each of the Services in a ServicePackage needs its own resource-facing services to support it, such as traffic identification and conditioning; these are defined by the ServiceBundle class.

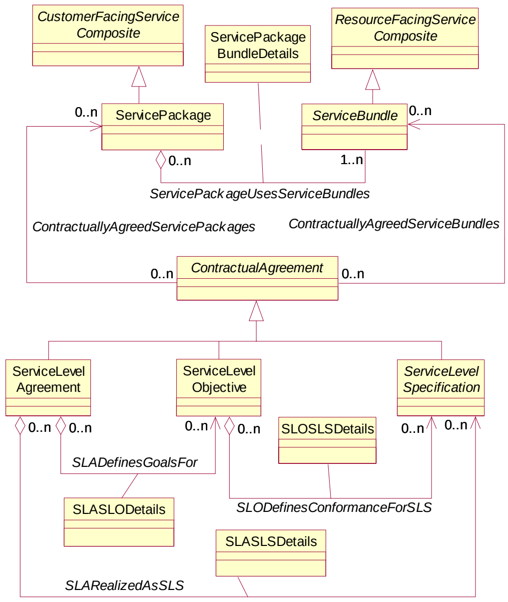

Figure 5. Simplified DEN-ng SLA Model

The ContractualAgreement class models entities that have contractual obligations, and hence compliance and violation concerns; it therefore serves as the superclass of all SLA entities. DEN-ng models SLAs in a flexible way by enabling the developer to define an SLA in terms of either Service Level Specifications (SLSs) directly or in terms of Service Level Objectives (SLOs) that in turn define SLSs. This is done by defining the three aggregations in Figure 5 as optional (e.g., 0..n to 0..n). The three association classes are used to enable external applications to define policies to govern each relationship.

Figure 6 shows the contractual model in more detail. Policy rules can be defined that govern which set of people and/or organisations can approve an agreement, incentive, or penalty. Both incentives and penalties can have a set of policies associated with them that are invoked at the start, during, and/or end of the incentive or violation time period.

3.6 Implementing Orchestrated SLAs

Our approach enables an autonomic manager to seamlessly translate between the different terminologies and concepts used by business, network, and other personnel by using a novel combination of information models and ontologies to represent the characteristics and semantics of SLAs. This enables machine learning and reasoning to be used to harmonise diverse sensed data from multiple sources in order to develop a more complete model of the current environment. Then, approaches such as the FOCALE autonomic architecture can be used to dynamically generate code from models so that, as context changes, network services and resources change in accordance with business objectives and policies.

The DEN-ng information model and the DENON-ng ontologies are used to define a machine understandable representation of business and networking concepts. This enables the autonomic manager to *reason* about which policies to use to orchestrate behaviour. In essence, the DEN-ng model functions as a template that can generate code to reconfigure devices, and the DENON-ng ontologies are used to select functions within the template.

This approach supports reuse and customisation, and enables the automation of large parts of the services contracting and fulfilment lifecycle, including negotiation, provisioning, and others. Benefits of such automation include faster contract set-up, reduced contracting and operating fees, and optimised service delivery for both customer and service provider.

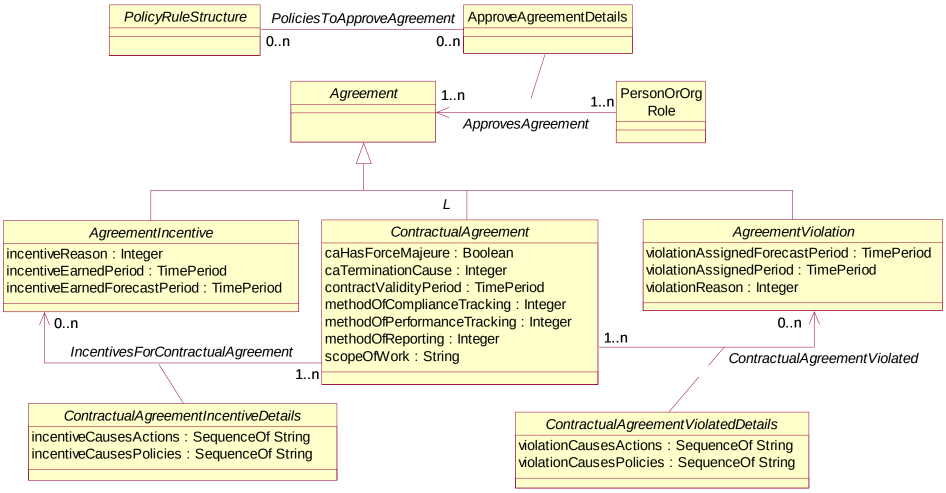

Figure 6. More Details of the DEN-ng Contractual Model

3.7 Use Case

In current Business and Operational Support Systems, it is impossible to directly relate the significance of different types of low-level network alarms to high-level business concepts, such as SLAs. This prevents changing user needs, business goals, and environmental conditions to be directly related to the current network services being offered, and more importantly, makes it difficult to agilely change those services to meet changing needs. In addition, future Internet applications will require context-awareness, which enables the set of offered services and resources to change to accommodate the needs of changing context.

Our approach enables both of these requirements to be met in an extensible manner by providing a platform in which to build reusable behaviours. Context, business goals, and network services are all modelled as objects; FOCALE *orchestrates* the behaviour of any object by dynamically generating code to govern the state changes of the objects being managed. For example, state changes can be related to commands to reconfigure the functionality of network devices, thereby changing the services that they offer. We have implemented examples of this as part of directed research [14] and to support service providers in the US. Hence, all elements of this approach are now ready, and can be viewed as a “toolbox” to be customised to support project-specific needs.

4. Business Benefits and Industrial Significance

The main business benefit of our approach is the ability to *intelligently automate the behavioural orchestration of network services and resources*. This can significantly reduce OPEX. Intelligent automation is achieved through (1) generating reusable code from the model, and (2) reasoning about different conditions and requirements in order to choose the best actions to take given unforeseen circumstances. Capital expenditures can also be reduced, as our approach can be used to identify new ways to reuse existing equipment.

A novel feature of our approach is its ability to model and reuse behaviour. Just as a programmer uses string functions, developers of our approach can define reusable classes and behaviour as libraries; code can then be generated for them to implement behaviour. This enables vendor-specific data to be translated into machine-understandable semantics.

For example, consider two domains that use network devices from different network manufacturers. If their configuration data is passed to each other, it is meaningless. Our approach understands what the configuration is being used for, and instead passes the appropriate semantic data such that each domain can cooperate what the other is doing. In addition, our approach can relate these data to business rules and policies, so that each domain will recognise the services and resources that are important to its collaborating domains and treat them accordingly.

5. Conclusions

We have developed a new approach to representing business and networking capabilities, constraints, and functionality. This is paired with a novel context-aware policy management approach that forms the basis for translating policies between constituencies.

Future work includes developing an enhanced version of the proven FOCALE autonomic architecture to understand and enforce SLAs. Machine based reasoning will be used to optimise service and resource allocation, ensuring that revenue goals are met or exceeded. Machine based learning will be used to capture performance and other information that helps make the control loops used more efficient.

Acknowledgements

This work is partially sponsored by the WCU (World Class University) program through the Korea Science and Engineering Foundation funded by the Ministry of Education, Science and Technology (Project No. R31-2008-000-10100-0) and by the European Union through the AutoI project in the 7th Framework Programme. The views expressed in this paper are solely those of the authors and do not necessarily represent the views of their employers, the AutoI project, or the Commission of the European Union.

References

[1] J. Strassner, "DEN-ng Model Overview", Joint ACF, EMANICS, and AutoI Workshop on Autonomic Management in the Future Internet, May 14, 2008

[2] M. Serrano, J. Serrat, J. Strassner, M. Ó Foghlú, "Management and Context Integration Based on Ontologies, Behind the Interoperability in Autonomic Communications", extended journal publication of the SIWN International Conference on Complex Open Distributed Systems, Chengdu, China, Vol 1, No. 4, July 2007

[3] A. Pras, J. Schoenwaelder, "On the Difference between Information and Data Models", RFC 3444

[4] TM Forum, http://www.tmforum.org/browse.aspx?catID=1684

[5] DMTF, "CIM Schema," http://www.dmtf.org/standards/cim/cim_schema_v2201/

[6] Object Management Group, "OMG Unified Modelling Language Specification", OMG, Version 1.5, March, 2003

[7] M. Fowler, "Dealing with Roles," http://www.martinfowler.com/apsupp/roles.pdf

[8] J. Strassner, J. de Souza, S. van der Meer, S. Davy, K. Barrett, D. Raymer, S. Samudrala, "The Design of a New Policy Model to Support Ontology-Driven Reasoning for Autonomic Networking", Journal of Network and Systems Management, Volume 17, Issue 1, March 2009

[9] J. Parsons, Y. Wand, "Emancipating Instances from the Tyranny of Classes in Information Modelling", ACM Transactions on Database Systems, Vol. 25, Issue 2, June 2000, pages 228-268.

[10] http://tomgruber.org/writing/ontology-definition-2007.htm

[11] A. Budanitsky, "Lexical semantic relatedness and its application in natural language processing" Technical Report CSRG390, University of Toronto, 1999

[12] http://wordnet.princeton.edu/

[13] J. Strassner, N. Agoulmine, E. Lehtihet, "FOCALE – A Novel Autonomic Networking Architecture", ITSSA Journal, Vol. 3, No. 1, May 2007, pages 64-79, ISSN 1751-1461

[14] J. Strassner, "Enabling Autonomic Network Management Decisions Using a Novel Semantic Representation and Reasoning Approach", Ph.D. thesis, 2008