Ontology Mapping for the Interoperability Problem in Network Management

Alfred Ka Yiu Wong, Pradeep Ray, *Senior Member, IEEE*, N. Parameswaran, and John Strassner

***Abstract*—Interoperability between different network management domains, heterogeneous devices, and various management systems is one of the main requirements for managing complex enterprise services. While substantial advances have been made in low-level device and data interoperability using common data formats and specifications such as simple network management protocol's (SNMP's) SMI and TMF's SID, various interoperability issues including semantic interoperability offer interesting research challenges. While semantic interoperability is a difficult problem in its own right, the semantic web that incorporates intelligent agents necessitates an interoperability solution requiring agents to communicate unambiguously and reason intelligently to perform cooperative management tasks. Agents need a formal representation of knowledge; an ontology is capable of modeling the rich semantics of the managed environment (and especially, relationships between managed entities) so that agents can act on them. This paper presents an ontology-driven approach for solving the semantic interoperability problem in the management of enterprise services, illustrated here with a router configuration management application.**

***Index Terms*—Integrated network management, ontology, ontology mapping, semantic interoperability.**

I. Introduction

THE PRESENCE of dissimilar network models and standards necessitates interoperability as a means of achieving ubiquitous connectivity and management [1]. While network management standards [e.g., Internet Engineering Task Force (IETF) SMI and OSI GDMO] have developed useful abstractions for representing many of the physical and logical characteristics of network entities, the problem of semantic interoperability has not been solved. Emerging standards and recommendations (e.g., DMTF CIM, ARM API, and TMF SID) provide knowledge representation required to solve the semantic interoperability problem among heterogeneous network devices [2]. However, this approach has three major limitations. First, these models are not adequate for next generation intelligent management solutions that would have to adapt with changing environments. Second, there is a substantial lag between the emergence of new technologies and the release of related standards. Finally, many standards do not adequately address the issue of the interoperability with other related standards that are part of a complete solution. The Internet-based e-business environment highlights the above requirements for management.

Manuscript received May 1, 2004; revised April 16, 2005. This paper was supported in part by ARC Discovery under Grant DP0451650.

A. K. Y. Wong, P. Ray, and N. Parameswaran are with the University of New South Wales, Sydney 2052, Australia (e-mail: alfred.ky.wong@gmail.com).

J. Strassner is with the Motorola Laboratories, Schaumburg, IL 60196 USA.

Digital Object Identifier 10.1109/JSAC.2005.854130

We propose an ontology-driven approach to the semantic interoperability problem, and illustrate it with a solution framework. This paper is organized as follows. Section II investigates the semantic interoperability problem at different levels of network management using the ITU-T TMN model, and discusses existing standards-based interoperability approaches. Section III illustrates our ontology-driven approach emphasizing two main components: similarity function and ontology mapping. A case study validation of this approach is illustrated with router configuration management in Section IV. The paper concludes with discussions in Section V.

II. Interoperability Problem and its Existing Approaches

The interoperability problem is clearly evident when management across heterogeneous devices, network elements, services and organizations is attempted. Interoperable systems must exhibit one or all of the following capabilities: 1) data/knowledge exchange; 2) coordinated behavior; and 3) cooperative problem solving. Interoperability may be classified into four levels in an increasing order of complexity and difficulty. These are interoperability on the physical, data, specification, and semantic levels [2]. While the physical and data format level interoperability solutions are known, the specification level interoperability involves the definition of software structures (e.g., CORBA) to help solve the semantic interoperability problem for a range of applications. However, the semantic interoperability problem is a difficult one and standards help solve the problem for certain domains of applications. Due to the limitations of standards-based approaches (discussed in Section I), some disciplines (e.g., healthcare) have adopted a more fundamental ontology-based approach, as discussed in detail in Section III onwards. Recent advances in semantic web research recommend the use of ontologies to be used by software agents [4].

The interoperability solutions may be at different levels—application interoperability, platform interoperability, management information base (MIB) interoperability, and so on. The ITU-T Telecommunication Management Network (TMN) has defined a standard model with a number of management layers that help in managing the complexity of telecommunication network management (Fig. 1). Although the TMN model does not address the semantic interoperability problem of networked services, we use this model due to its familiarity in the network management world.

A. Business Management Layer Interoperability

Business Management deals with the executive-level functions of enterprise management as a whole, including the issues

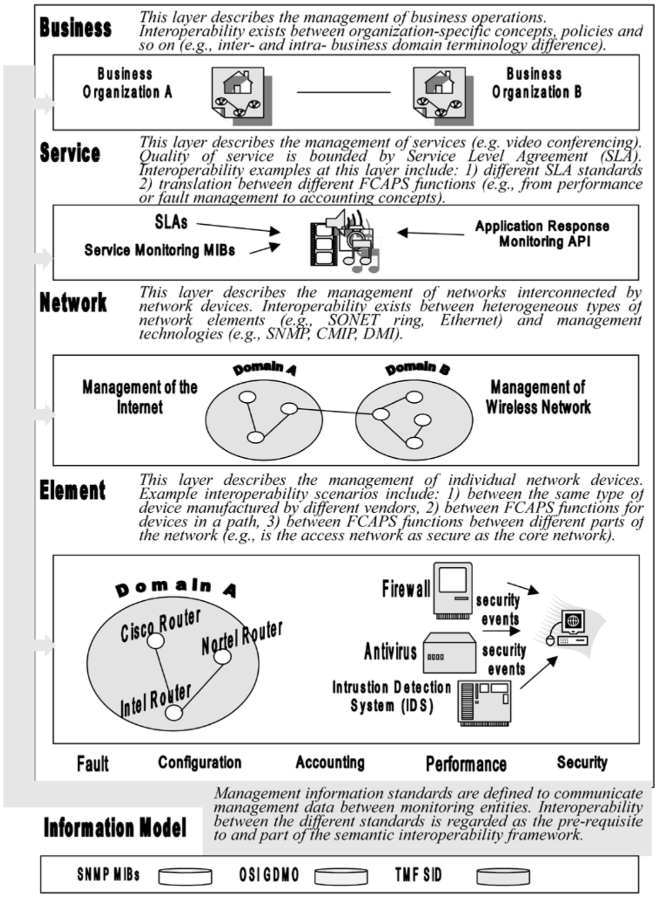


Fig. 1. Illustration of the interoperability problem between and within the FCAPS functions based on the TMN model.

such as strategic planning, cost, revenue, capital investment, market analysis, labor force, etc. The business level management is best analyzed with respect to four factors: people, organization, process, and technology. The interoperability problem is very large at this level because of differences in semantics of different business domains (e.g., healthcare versus finance), different organizational cultures (e.g., Japanese versus European), different languages, different business processes (e.g., transaction management in banking versus that in healthcare), and different terminologies (e.g., for different subdisciplines in healthcare) that could lead to the same meaning for different terms, or different meanings for the same term. These differences would affect the management of e-businesses that need to work seamlessly across organizations, countries, and disciplines.

Consider an interoperability scenario in telecom management between a vendor (e.g., a router vendor) and a customer (e.g., a telco). At the business layer, they both talk about "long term commitment," "low cost," and "ROI." Unfortunately, due to their different subject domains, these common terms usually mean totally different things. This creates confusion and conflicts, and prevents the efficient negotiation of an Service Level Agreement (SLA) between them.

In the telecommunication domain, the TMForum e-TOM provides an example of a business level management interoperability framework. Such standards are evolving in different e-business sectors, such as healthcare and finance. However, standards relate to specific situations and they have to be updated and interpreted by humans to develop solutions. Furthermore, most standards cannot be interpreted by software agents.

B. Service Management Layer Interoperability

Service Level Management (SLM) using SLAs enables business management to be linked with technology infrastructure management (e.g., services provided by the network). However, the success of SLM depends on the ability to standardize the semantics of SLAs [e.g., quality-of-service (QoS)] across different domains. For example, one Internet service provider (ISP) might be providing service to both healthcare (e.g., videoconferencing) and finance (e.g., processing of share transactions) domains; hence, it would be necessary to have common semantics for SLM [2].

Interoperability problems arise when business requirements are translated to service requirements. For example, the service requirements for the video conferencing for an e-healthcare business that conducts online surgery can be quite different from the ones for a financial company that conducts online consultation with clients. The former might demand maximum availability (e.g., in a telesurgery application) and quality (e.g., high-resolution graphics for scan images) for the duration of surgery, while the latter requires continuous availability and flexible delivery of information (e.g., stock prices).

Standards such as GB917 (SLA Management), SES (Solution Exchange Standard), SIS (Service Incident Exchange Standard) and ITU-T X.790 aim to enable multiple service providers to share management knowledge for service incidents, billing information, and other solutions [2]. These standards focus on the need for the telecommunication industry. Although other business sectors (e.g., finance and healthcare) can benefit from these standards, the semantic requirements of SLM in other industries are quite different. Besides, the existing standards-based approaches to SLM are not suitable for agents, and the solutions are not complete and are not adaptable to the evolving dynamic threats (e.g., cyber attacks).

C. Network Management/Element Management Layer Interoperability

This section presents the interoperability problem of the network and element management layers of the TMN model. Most networks are built from equipments manufactured by different vendors. The semantic interoperability is caused by the fact that different vendor equipments have different functionality, interfaces, and programming (customization) models. For example, at the time of this research, Cisco routers had more than 18 000, while Nortel routers had much less—about 2000 configuration functions due to the differences in the design philosophy and the implementation of the command line interface (CLI) used by Cisco and Nortel.

This situation could be illustrated with a simple example (more detail can be found in [11]). Let us assume that a network engineer has a Border Gateway Protocol (BGP) peer configuration task for a router network containing Cisco and Nortel routers. As can be seen on Fig. 2, the BGP peer configuration programs are completely different, although they express the

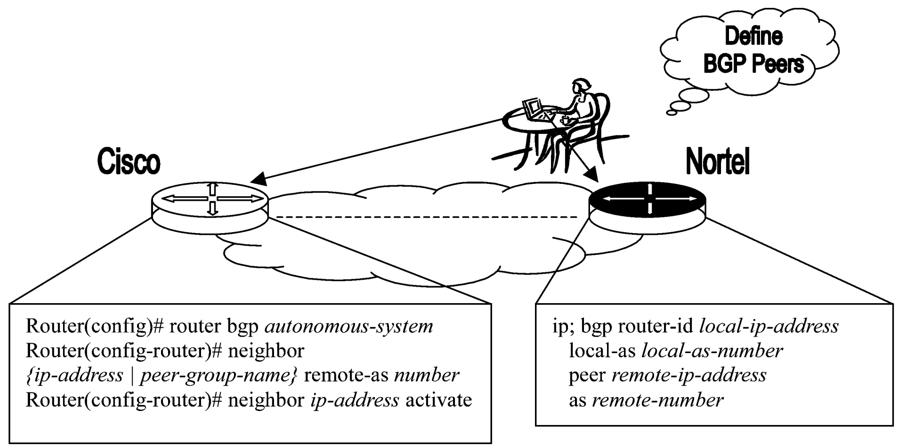


Fig. 2. Router configuration interoperability scenario.

same configuration task. More importantly, the program for the Cisco router presents different configuration *modes*, which are absent in the Nortel router configuration program. This in turn means that the engineer must be aware of these differences in the corresponding CLIs. The situation may be even more complicated due to changes that can be introduced in new versions of the operating systems of those routers. This means that the network administrator must translate a high-level specification of the desired functions (e.g., traffic conditioning for gold, silver, and bronze services) into different sets of commands for each vendor.

An ontology-driven interoperability approach (discussed in Sections III and IV) will help us adapt existing automated network management solutions to new network types (e.g., wireless and sensors).

III. Ontology Mapping Driven Interoperability Framework

A. Ontologies and Their Realization in the Semantic Web

Ontology is regarded as an explicit specification of a conceptualization that facilitates knowledge sharing and reuse [12]. It has been used in different domains in the capturing, representing and structuring of knowledge. Ontologies are constructed using generalization relationships to form their taxonomies and using other semantic relationships (e.g., whole-part, function) to capture the meanings of concepts and factual knowledge of a domain. Formality of ontology ranges from those represented by textual description, set of words (e.g., WordNet) to more formal languages such as OWL and Description Logic.

In an intelligent network environment, we use an ontology to provide semantics for building an underlying knowledge base that not only allows agents (manager entities) to communicate, but also to reason with each other, enabling the desired tasks to be performed collaboratively. In fact, different knowledge bases can be regarded as different views of an ontology [3]. The domain knowledge captured in the ontology provides the basis for agent intelligence. Ontology mapping can then be performed between ontologies such that the semantics between concepts can be matched accordingly. The mapping between semantics is used to resolve the semantic interoperability problem.

TABLE I
Summary and Comparison of Standards-Based and Ontology-Driven Interoperability Approach

Feature	Standard-based approach	Ontology-driven approach
Reusability	*Application and domain specific.*	*Separate ontologies can be defined for different applications and domains. A single ontology mapping algorithm is reusable across the ontologies.*
Adaptability	*New standards or updates are required to support new technologies.*	*Before the related semantics of the new technologies is fully incorporated into the existing ontologies, the new technologies can be temporarily supported by analyzing their relationships with the existing semantics embedded in the ontologies.*
Support for intelligent agents	*Standards normally define the format, syntax and structure of information. Their formality supports machine processing, but the lack of semantics (similar to XML) does not allow intelligent processing.*	*The semantics embedded in the ontologies (relationships between ontological concepts, behavioral characteristics), allows agents to formulate intelligent decisions.*
Coverage of the semantic interoperability	*The collaborative effort of standards (data, specification and so on) can be regarded as the act of achieving the semantic interoperability. However, the existence of competing and overlapping standards introduces another dimension of the interoperability problem.*	*Although there exist different understandings of ontology by different communities, ontology as a theory provides an unrestricted form of semantic analysis, reasoning and mapping. Semantic mapping should be able to be performed between any pair of ontologies to solve the semantic interoperability problem.*

We reduce the problem of semantic interoperability of management to the ontology mapping task between semantics of application concepts (Section IV). Table I compares the ontology with standards-based approach for semantic interoperability of management.

B. Semantic Modeling Using Ontology

The use of ontology as a basis for semantic modeling of managed objects and processes is a relatively new research and development activity [15]. While historically the majority of the work on semantic modeling was related to understanding natural language, information retrieval, and machine learning, the recent advances in network and service management, e-enterprise management, and semantic web applications, have significantly increased the interest in semantic modeling generally, and particularly, in those methods that use ontology for knowledge conceptualization.

In our research, we consider few important issues regarding ontology. Despite significant interest in ontologies, ontology mapping is mostly done manually or semi automatically. In order to practically tackle real-world problems, automatic ontology mapping is required. Furthermore, computational effectiveness is critical to solve real-world problems. We keep this in mind and derive a computational effective ontology model (logic based) and a mapping algorithm. Finally, intelligent agents now form part of the distributed and dynamic environment (e.g., Internet). We propose an ontology solution that is adaptable by intelligent agents.

In the domain of network management, the concepts that can be modeled in a network ontology include tangible (e.g., router interface) and intangible (e.g., BGP parameters) *network objects*, and *management operations* (e.g., router configuration commands, security monitoring operations). Semantic relationships far more complex than *is-a* relationship often exist between the concepts. For example, the concept, router configuration command, can be semantically expressed as the state transition of *network objects*. While slots can be used to represent and abstract a single semantic relationship, we suggest the use of logic (with its powerful expressiveness) to represent the more complex semantic relationships.

C. Similarity-Based Ontology Mapping

In this section, we propose the method of automatic ontology mapping based on a semantic similarity function that will be defined between the concepts.

1) Process of Ontology Mapping: The task of ontology mapping can be divided into the following steps.

Step 1) **Ontology acquisition**—Collection of user guides, manuals, expert testimonies, management practices, and other not formalized documents describing the problem domain.

Step 2) **Ontology formalization**—Specification of ontology in RDF, OWL, Ontolingua, Protégé, or any other formal ontology specification language. In our case, we formalize ontology in logic.

Step 3) **Concept similarity estimation**—Evaluation of the semantic content of concepts and calculating the semantic similarity between the matched concepts.

Step 4) **Ontology mapping**—A search procedure of finding the highest similarity match between concepts belonging to the mapped ontologies.

Step 5) **Results presentation**—Presentation of the ontology mapping results in a user-friendly format, or in a form required by other components of the application systems

The research focus of this paper is on steps 3) and 4). We will introduce a novel semantic ontology similarity function and demonstrate how to build an ontology mapping procedure that preserves the highest similarity between the mapped ontologies.

2) Related Work on Ontology Mapping: Ontology mapping has been widely researched, especially in the semantic web [4], [9], [10], [15]. Traditionally, ontology is researched in the linguistic domain [14] and more recently in other domains such as biological [13], and network management [5], [6]. Works on concept similarity assessment such as [14] compute similarity based on a single taxonomy. They rely on corpus statistics (frequency of concepts) which is only suitable for the comparison between words. While existing ontology mapping approaches can be classified as manual, semiautomatic [10] and automatic [8], [9], they can also be categorized technologically. [9] employs machine learning techniques to measure similarity between concepts based on number of shared instances. However, large number of training instances is required. Also, the computation of similarity based on instances is contradictive to the more intuitive approaches that are based on semantic relationships (between generic concepts). References [8] and [13] compute similarity between concepts, presented as set and graph comparison, respectively, based on semantic relationships such as *is-a*, *part-of*, and *function*. Our solution allows comparison between more complex semantics (logic allows more powerful expression). Reference [7] presents a more formal approach that uses propositional logic as the basis for strict exact matching.

Almost all existing works miss (or lack an efficient approach of) Step 4) of the process of ontology mapping (Section III-C1). We propose a search procedure in Step 4) (Section III-C5) that is computationally efficient. Finally, the notions of similarity computation between one-to-many and many-to-many concepts are not adequately addressed by (if any) existing approach. Although they are not addressed in this paper, we are developing such extensions (based on Section III-C5). For example, a result of our case study (described in Section IV) shows the need to explore mapping between sets of commands (many-to-many).

The initial derivation of our works [18] is presented in the context of security management.

3) Method of Estimating Concept Similarity: As it was described in the overall process of ontology mapping in Section III-C1, the prerequisite for ontology mapping is the ability to measure the semantic similarity between the concepts of different ontologies. The task of semantic mapping between the ontologies O_1 and O_2 could be defined as a search procedure for finding for each concept C_1 in ontology O_1 a matching concept C_2 in ontology O_2 so, $S(C_1, C_2)$ has a maximum value, where S is the similarity function. We divide the similarity measurement into three main tasks: 1) translation of the ontology from the initial ontology formalization language into the semantic ontology description language; 2) generation of the semantic content of the concepts; and 3) calculation of the semantic similarity estimates between the concepts.

In the proposed method of ontology mapping, we are using the first-order logic (FOL) calculus [16] as the language for describing the semantics of the domain concepts and objects. Such use of FOL as a semantics specification language has been successful in other research areas, e.g., in describing the semantics of programming languages [17]. In many practical application areas, e.g., network and service management, e-enterprise applications, and semantic web, the semantic content of objects and interobject relations can be represented in FOL and effectively estimated by the complexity of the corresponding FOL expressions. Different aspects of an object, such as its structural components, class references, constraints, and functions, as well as relations between concepts, will be described in FOL as separate logical statements containing elementary FOL statements like $\exists\, x\ Network\text{-}Object(x) \wedge Semantic\text{-}Relationship(x)$. As a direct consequence of using FOL as a semantics specification language, the similarity of concepts can be evaluated by comparing the corresponding FOL expressions.

Prior to the definition of our similarity function, we will introduce the notion of an *aspect* of a concept. Each concept of a domain (domain ontology) is a mathematical abstraction of entities of a physical or virtual world of interest. Entity in those worlds may have complex internal structures they may participate in different interactions, or exhibit various behavior in time and space. In order to capture these complexities at an abstract concept level and take them into account while estimating the semantic similarity between the concepts, the notion of an aspect of a concept is introduced. For example, two *network objects* such as *OSPF*- and *RIP-Routing Table* can be semantically similar regarding their structural (st) aspects (both have parts—source IP, metric fields and so on), but dissimilar regarding their class references (cr) aspect (one is a subconcept of *OSPF Resource* and one is a subconcept of *RIP Resource*). Depending on the application domain, each aspect of a concept has a different level of importance to the overall semantics of the concept. In order to take this into account, we use a weight function $\omega(i), \Sigma\omega(i) = 1$, to modulate the contribution of each aspect i. The assignment of the specific values for $\omega(i)$ is the task of domain experts, and is beyond the scope of this paper.

Throughout this paper $L_i(C)$ denotes the logical semantics of an aspect i of a concept C expressed in FOL. The semantic similarity S between two concepts C_1 and C_2 will then defined as the sum of the weighted logical similarities S_L between the $L_i(C_1)$ and $L_i(C_2)$ over all aspects i.

Definition 1: $S(C_1, C_2) = \sum_i \omega(i) * S_L(L_i(C_1), L_i(C_2))$.

It is important to note that we will define the similarity function only for common aspects. Therefore, if the aspect i is not presented in both comparable concepts, then $\omega(i) = 0$.

The next step in the semantic similarity computations is to determine the nature of the function S_L. First of all, we assume that the logical expressions $L_i(C_1)$ and $L_i(C_2)$ will be represented in their conjunctive normal form (CNF), where each constituent (assertion) is a disjunction over elementary predicates. The CNF representation permits us to reduce the task of computing the logical similarity between the expressions $L_i(C_1)$ and $L_i(C_2)$ to the task of estimating the degree to which the assertions as shared by $L_i(C_1)$ and $L_i(C_2)$ (we denote the total amount of assertions shared in common as their ***commonality***). Following this, the logical similarity function S_L is defined as follows:

Defintion 2: Similarity Function S_L Between Logical Expressions $L_i(C_1)$ and $L_i(C_2)$:

$$S_L(L_i(C_1), L_i(C_2)) = \frac{\min\left(\sum_{A\in L_i(C_1)} \partial(A), \sum_{A\in L_i(C_2)} \partial(A)\right)}{|\{L_i(C_1)\} \cup \{L_i(C_2)\}|}$$

where A is an assertion; $\{L_i(C)\}$ denotes the set of assertions from $L_i(C)$; and the function ∂ measures the degree (in the range of $[0, 1]$) to which assertion A is shared between the expressions $L_i(C_1)$ and $L_i(C_2)$. The definition of the function ∂ will be given in the Section III-C4.

Before calculating the function ∂ over the assertions in $L_i(C_1)$ and $L_i(C)$, we need to know what assertions of $L_i(C_1)$ could be inferred from $L_i(C_2)$ and *vice versa*. We propose to use the logical selected, linear, definite (SLD) resolution (see Appendix I) procedure to infer such $L_i(C_1)$ and $L_i(C_2)$ assertions. While there are different functions we can employ to combine the asymmetric results of ∂ returned for both directions such that S_L satisfies the symmetrical property, we select the minimum function to model the pessimistic measurement for worst case scenario.

4) Generation of Semantic Content (SLD Resolution): This section describes our strategy for measuring the degree of ***commonality*** between FOL statements using SLD resolution. Since an assertion can be composed of more than one predicate, it can be partially resolved during SLD resolution. Furthermore, there may exist more than one way in which a predicate is resolved. We derive an approach to quantify how much a (partially or totally) shared assertion contributes to the ***commonality*** of $L_i(C_1)$ and $L_i(C_2)$.

We observe that general concepts tend to exhibit broad semantics (i.e., the more general the concept, the larger the set of specific concepts it is subsuming). The diversity of meanings embedded in the subsuming set of concepts contributes to the broad semantics. For example, in the network security domain, the generic concept of reconnaissance attack can be structurally explained as the (more specific concepts of) reconnaissance attacks that are performed over transmission control protocol (TCP) or user datagram protocol (UDP) connections. In a FOL statement, broad semantics is indicated by the existence of logical operator $\vee$ within its composite assertions. The notion of broad semantics (exhibited in generic concepts) is considered in our concept similarity assessment, mainly to facilitate the ontology traversal strategy presented in Section III-C5. According to our strategy, similarity assessment between specific and generic concepts is inevitable on the search path from the ontological root to the final matching node (see example in Fig. 4). As a result, matching between different broadness of semantics becomes part of our concept similarity measurement. For example, consider a generic router concept A that configures Boolean- $\vee$ numeric-value OSPF variables, and concepts B and C that configure only numeric-value OSPF variables. Our similarity measurement has to quantify the intuition that $S(B, C) > S(A, B)$ or $S(A, C)$. This intuition is true simply because B and C have the potential to be identical, while the broad semantics exhibited in A (Boolean $\vee$ numeric) eliminates such potential. Similar to the principle of information theory, the broader the semantics, the less meaningful the concept is (i.e., the concept, *network object*, is not as meaningful as *OSPF Resource* because the former can be a *BGP or an OSPF Resource, or even something completely different*). Therefore, we suggest the use of information theory to estimate the broadness of semantics.

A FOL statement in CNF contains a conjunction of assertions that in turn contain disjunction of predicates. The task of measuring the ***commonality*** between two FOLs is then divided into the steps of measuring the resolution qualities of predicates and of assertions.

We now consider the task of measuring the resolution quality of predicates. Let us consider the scenario of SLD resolution of $L_i(C_1) \models L_i(C_2)$ ($\models$ implies that all assertions from $L_i(C_2)$

can be inferred from $L_i(C_1)$). A negated predicate $\neg p(x)$ (that belongs to $\neg L_i(C_2)$) can be SLD-resolved in different ways such as by assertions (from $L_i(C_1)$): $p(y)$, $h(y)$ (assume that $h(y)$ is related to $p(x)$ by axiom $p(x) \rightarrow h(y)$), $p(y) \vee \cdots q(z)$, and $h(y) \vee \cdots q(z)$. We first estimate the broadness of semantics of an assertion using the well-known formula $\sum(\alpha(e) * (-\log_2(\alpha(e))))$, referred to as the *Entropy* of a series of events e, where α is the probability distribution of event occurrence. However, α is not always available. In such situation, we estimate the semantic broadness of $p(y) \vee \cdots q(z)$ as its *Entropy* by setting $\alpha(e)$ as $1/n$, where $e = p(y)\ or\ \ldots q(z)$ and n denotes the number of predicates in $p(y) \vee \cdots q(z)$. We then determine the similarity between the two predicates that are resolved ($\neg p(x)$ and the corresponding predicate from the corresponding assertion from $L_i(C_1)$). The resolution quality of $p(x)$ is defined as S weighted by $e^{-Entropy(A)}$ (models the idea of the broader the semantics in A, the lower the resolution quality).

Defintion 3: $\mathbf{\Omega}(p, A') = e^{-Entropy(A')} * S(C_p, C_{p'})$, where p is a predicate, A' an assertion, and the predicate p being SLD-resolved by another predicate p' from assertion A'. S, defined in Defintion 1, is used recursively to measure the similarity between C_p and $C_{p'}$, where C_p and $C_{p'}$ are concepts denoted by p and p' respectively.

Now that we have a measure $\mathbf{\Omega}$ for the task of measuring the resolution quality of predicates, we select the maximum $\mathbf{\Omega}$-value among the predicates of an assertion to represent the resolution quality of the assertion. We define a function ∂ for such measurement.

Definition 4: $\partial(A) = \mathrm{Max}_{p \in A}(\Omega(p, A'))$ where p is a predicate from assertion A and A' is the assertion that is used to resolve p.

Note that SLD resolution is intuitively asymmetric. Consider the two logical statements $p(x)$ and $p(x) \vee q(y)$, the successful SLD resolution of $p(x) \models p(x) \vee q(y)$ does not guarantee SLD resolution of $p(x) \vee q(y) \models p(x)$. Since $\mathbf{\Omega}$ and ∂ are based on SLD resolution, they are inevitably asymmetric in nature. However, as mentioned in previous section, the min function used in Definition 1 ensures that our definition of similarity is symmetric.

Due to the nontrivial amount of technical details presented on our concept similarity measurement, we illustrate an example on the usage of our defined ideas (see Appendix II).

5) Ontology Mapping Procedure: We now propose the ontology mapping procedure [Step 4) in Section III-C1)]. The procedure includes two main components, namely, a classification scheme and an ontology traversal algorithm. The former categorizes matching results according to their similarity values, and the latter employs the categorization scheme to guide the search through target ontology during an ontology mapping process.

Matching results can intuitively be classified into the classes shown in Table II.

TABLE II
TRADITIONAL AND INTUITIVE CLASSIFICATION OF MATCHES

Class	Formalism
Equivalent concepts	$C_1 = C_2 \Leftrightarrow \forall i\ [L_i(C_1) \models L_i(C_2)\ \&\ L_i(C_2) \models L_i(C_1)]$; $S(C_1, C_2)=1$
Sub concept	$C_1 \subset C_2 \Leftrightarrow \forall i\ [L_i(C_1) \models L_i(C_2)]$; $0 < S(C_1, C_2) < 1$
Associated concepts	$C_1 \approx C_2 \Leftrightarrow \exists i\ [\models L_i(C_1) \cap \models L_i(C_2)] \neq \varnothing$; $0 < S(C_1, C_2) < 1$
Dissimilar concepts	$C_1 \otimes C_2 \Leftrightarrow \forall i\ [\models L_i(C_1) \cap \models L_i(C_2)] = \varnothing$; $S(C_1, C_2)=0$

Since $S_L(L_i(C_1), L_i(C_2))$ can still be computed even when neither of the $\models$ directions holds, we relax the strict membership constraints (e.g., $L_i(C_1) \models L_i(C_2)$ has to hold in subconcept) of the classification scheme in Table II. We observe that if $C_1 \subset C_2$, $|\{L_i(C_1)\}| > |\{L_i(C_2)\}|$ and or $\sum_{A \in L_i(C_1)} \partial(A) < \sum_{A \in L_i(C_2)} \partial(A)$. The observation can be implemented in our similarity measurements: we denote $S(C_1 \rightarrow C_2)$ as a variant of Definition 1, where $\sum_{A \in L_i(C_2)} \partial(A)$ and $|\{L_i(C_2)\}|$ replace $\min\left(\sum_{A \in L_i(C_1)} \partial(A), \sum_{A \in L_i(C_2)} \partial(A)\right)$ and $|\{L_i(C_1)\} \cup \{L_i(C_2)\}|$, respectively, in the S_L component. The observation can then be generalized as $S(C_2 \rightarrow C_1) < S(C_1 \rightarrow C_2)$ for $C_1 \subset C_2$. Our classification scheme is then defined accordingly as follows.

We now present our strategy for traversing an ontology during the process of concept mapping. The strategy is based on the classification scheme defined above. Let C_2 and C_1 be the source and target concepts in their respective ontologies O_2 and O_1. The strategy is then defined accordingly in Table III

Step 1) Start with $c =$ root of O_1, and an empty cache H that is used to store concepts.

Step 2) Push the search downwards to children of c. If c is a leaf node, cache c in H and go to Step 5). Else, go to Step 3).

Step 3) Select amongst the children of c, a node c' that produces highest S between c' and C_2. Set $c = c'$. If c' is more general than C_2, repeat Step 2) on c'. Else if c' is more specific than C_2, go to Step 4). Else, go to Step 6).

Step 4) Cache c in H. Select amongst the siblings of c, a node c' that produces highest S and c' is more general than C_2. If $S(c', N) > S(\text{parentof } c', N)$, repeat Step 2) on c' with $c = c'$ Else, go to Step 5).

Step 5) Select amongst the concepts in cache H, a node c', that has highest S between c' and C_2. Return c' as the closest match for C_2.

Step 6) If $S(c, C_2) = 1$ i.e., c is an exact match of C_2, return c as the closest match for C_2. Else compute S values between C_2 and every child c' of c. If $\exists\ c' S(c', C_2) > S(c, C_2)$, perform Step 2) on c. Else cache c in H and go to Step 5).

TABLE III

Class	Condition	Remark
Exact	$S(C_1,C_2) = 1$	C_1 and C_2 have *equivalent* meanings
General *C_1 is a Super-concept of* C_2	$S(C_2 \quad C_1) > S(C_1 \quad C_2)$	C_1 has *partial* meaning of C_2.
Specific *C_1 is a Sub-concept of* C_2	$S(C_2 \quad C_1) < S(C_1 \quad C_2)$	C_1 contains *all of,* plus *unrelated* meanings, of C_2.
Overlap	$0 < S(C_1,C_2) < 1$	C_1 has *overlapping* meanings with C_2.

Although standard algorithms such as breath first search (stop searching downwards when more specific nodes are reached) can be used, exhaustive searching is computationally expensive. Our strategy serves as a heuristic for more efficient ontology searching.

IV. Case Study: Router Configuration Management Interoperability Resolution

This section describes the application of the proposed solution presented in Section III to the router configuration domain. We selectively study Cisco and Nortel. In this domain, we are interested in mapping between application concepts, router configuration commands.

A. Cisco and Nortel Ontologies

During the mapping process of router configuration commands, we consider two semantic aspects, namely: 1)*sp*(*states-prior-to*) and 2) *sa* (*states-after*), where states refer to the states of the *network objects* that are involved in the command execution. These two aspects separately capture the semantics of precondition and postcondition of the command execution, and together capture the semantics of state transition of the *network objects*.

However, in most cases, the differences in preconditions are the result of varying programming models (e.g., Cisco has different configuration modes: *config, config-router*, and *config-if*). Furthermore, although most commands cause state transition of *network objects*, the *states-prior-to* is often irrelevant to the command execution itself (for example, it does not matter what state/value *OSPF-HelloInterval* is in prior to a command, such as *ip ospf hello-interval*, that reconfigures it).

Even though the aspect *sp* is not important for the measurement of similarity between individual commands (one-to-one mapping), both aspects *sp* and *sa* are critical for the similarity measurement between command sequences (one-to-sequence and sequence-to-sequence mappings). For example, in searching for a match for the Nortel command that sets the router's Ethernet interface media type, a sequence of Cisco commands (e.g., *configure term, interface ethernet 2,* and *media-type 10baset*) can be returned. Note that this one-to-sequence match relies on matching between series of *network object* states from both aspects *sp* (i.e., *admin-mode* maps to ⟨ sequence ⟩ ⟨ *config, config-if* ⟩) and *sa* (i.e., *media-type* maps to a new value).

During the comparison between *network objects* and between states, we consider the class reference (cr) aspect, i.e., $S(C_p, C_{p'})$ from Definition 3 is computed on the cr aspect.

We illustrate in Fig. 4 some examples on the modeling of the aspect *sa* of commands in FOL. The ontological taxonomies of application concepts—router configuration commands, associated *network objects* and their possible states are illustrated in Fig. 3.

B. Cisco and Nortel Command Translation as Ontology Mapping

In this paper, we only present our experiments on the mapping between individual commands. Hence, the simple weight function $\omega(sa) = 1$ (i.e., only consider semantic aspect sa) is used for the similarity function S.

Following the ontology traversal strategy presented in Section III-C5, we illustrate in Fig. 4 an example of the search for possible matching commands from the Cisco ontology for the Nortel command *define ip base ripCompatibility interface* (denoted $\mathbf{N}$):

In Step A, (results of executing Step 2) in Section III-C5) the search is pushed downwards onto nodes b, c, and d because the root is more general than N. In Step B, (results of executing Step 3) node b is selected amongst b, c, and d since $S(b,N) > S(c,N)$ or $S(d,N)$. Also, $S(N \rightarrow b) > S(b \rightarrow N)$. Therefore, according to our classification scheme, b is more general than N. Step 2) is then performed on node b. In Step C, (results of executing Step 3) node e and f are compared against node N. Node e is selected amongst node e and f since $S(e,N) > S(f,N)$. Again, $S(N \rightarrow e) > S(e \rightarrow N)$. Therefore, e is more general than N. Step 2) is then repeated on node e. In Step D, (results of executing Step 3) nodes g, h, i, and j are compared against N. Nodes g and h are selected amongst nodes g, h, i, and j since they have higher similarity scores. Node g or h is more general than N because $S(N \rightarrow g \text{ or } h) > S(g \text{ or } h \rightarrow N)$. Therefore, Step 2) is performed on g or h. However, g or h is a leaf node and, therefore, is cached in H. As a result of executing Step 5), node g or h is selected amongst the nodes in H, where $H = \{g \text{ or } h\}$. Hence, g or h is returned as the closest matching Cisco command to the Nortel command N.

C. Evaluation

In our case study, experiments were performed mainly to conceptually validate our ontology mapping driven interoperability solution (not an exhaustive implementation). Keeping in mind that there exists no semantic approach like ours, without comparison, we can only evaluate the sensibility of the overall mapping results. Our ontologies were constructed using Protégé (http://protege.stanford.edu), while the ontology mapping strategy was implemented in Java. Approximately, 250 Cisco commands and 200 Nortel commands were analyzed, modeled,

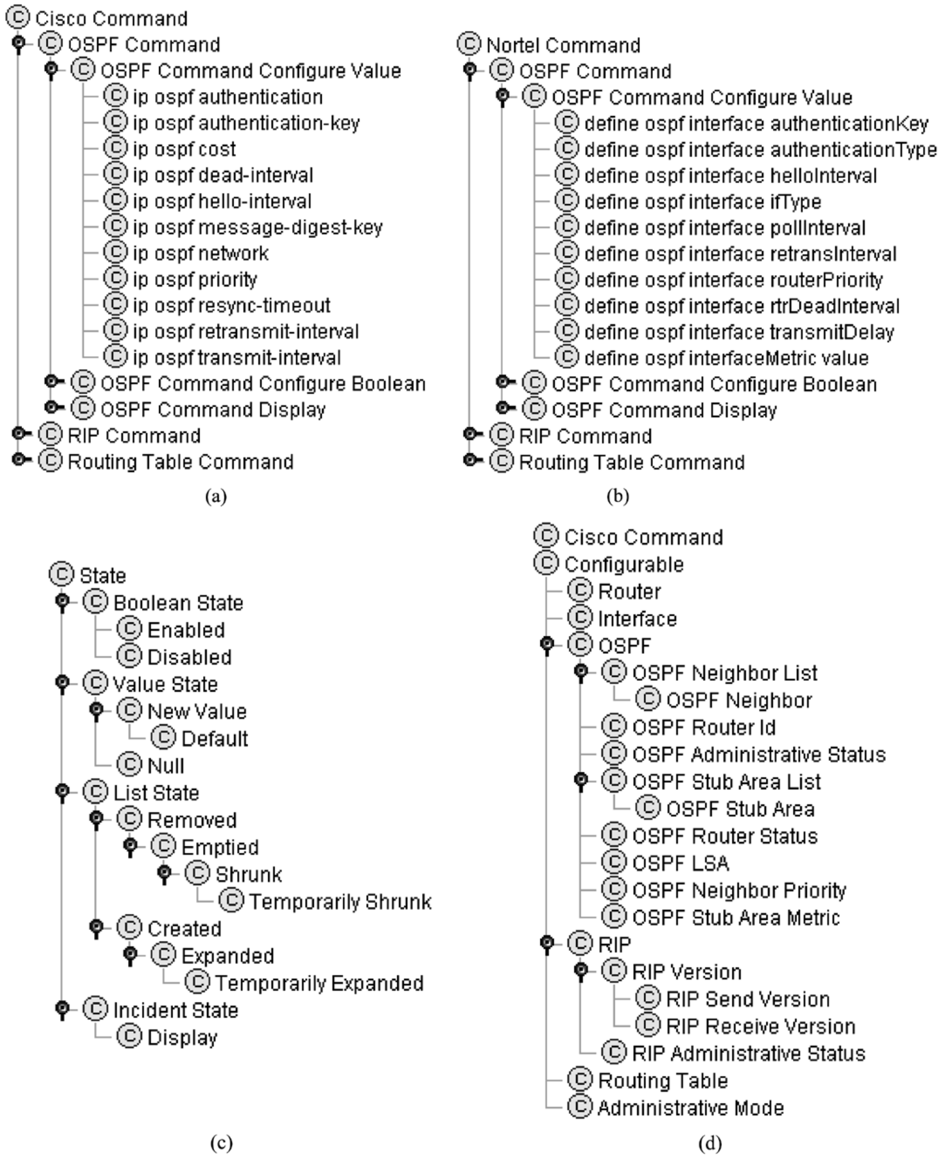


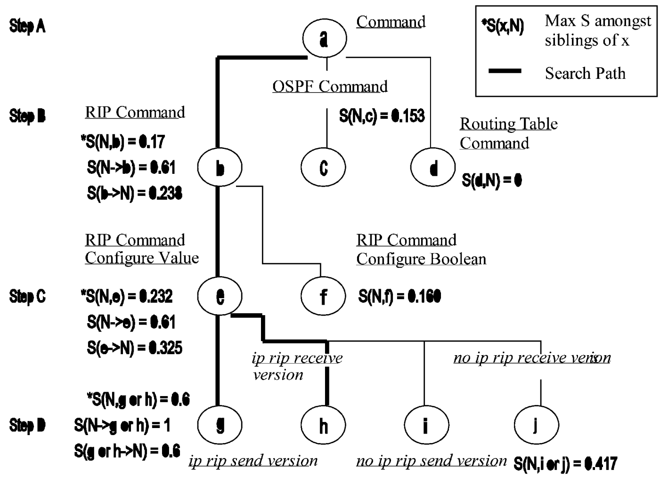
Fig. 3. Ontological taxonomies of (a) Cisco command, (b) Nortel command, (c) state, and (d) configurable.

and classified into two different ontologies. We applied our mapping strategy to a set of selected commands. We observed that Cisco commands were more general than Nortel commands, and that there was rarely an exact match if the semantic aspect *sp* is also considered. Each Cisco command achieves less than what a Nortel command does, and Cisco's programming model differs from Nortel's (due to the different Cisco administration modes).

Our mapping solution is not without its limitations. One problem is that due to the presence of term e^{-1} in $\mathbf{\Omega}$, the density of numeric values computed by $\mathbf{S}$ are unevenly distributed. Thus, the similarity measures of less similar concepts are distinguished by small numerical differences. Further studies are needed to refine the definition of similarity measures and their computational methods so that they capture more effectively the intuitive notions of similarities and dissimilarities amongst concepts in this domain. Another problem is related to the limitations of the theorem proving of FOL. In scenarios where SLD resolution does not promptly return a conclusion on the equivalence between two logical statements, the case where the resolution process enters into a loop due to nonequivalence between the statements is not distinguishable from the case where a prolonged amount of time is required to prove the equivalence between the statements. However, this problem should be minimized when the mapping approach is applied in carefully studied domains such as network management where the complexity of the FOL statements can be managed.

D. Discussion

We have demonstrated our ontology driven interoperability framework in the router configuration example in the TMN EML layer. The framework can definitely be reused and applied in other TMN layers once we identify the concepts that require integration (e.g., an ISP may need to sign SLAs with a company with two divisions—healthcare and finance), as shown in Fig. 5. A solution to the semantic interoperability problem across different domains such as healthcare and finance will be useful in computing generic variables such as costs, penalties, resource availability, etc. These ontologies could then be linked to the underlying network and element layers for integrated management applications. Application of ontology mapping



N	∃x,y,z(Interface(z) ∧ RIP-Version-Receive(x,z) ∧ RIP-Version-Send(y,z)) ∧ NewValue(x) ∧ NewValue(y)
b	∃x,z(Interface(z) ∧ (RIP(x,z) ∨ RIP(x))) ∧ (Boolean-State(x) ∨ Value-State(x) ∨ List-State(x) ∨ Incident-State(x))
e	∃x,z(Interface(z) ∧ (RIP(x,z) ∨ RIP(x))) ∧ Value-State(x)
g	∃x,z(Interface(z) ∧ RIP-Version-Send(x,z)) ∧ New-Value(x)
h	∃x,z(Interface(z) ∧ RIP-Version-Receive(x,z)) ∧ New-Value(x)

Fig. 4. Illustrative example of command mapping via ontology mapping.

methodology to network management (e.g., MIBs and DMI) has been reported in [6].

V. Conclusion and Future Work

This paper has presented an ontology-driven interoperability approach for solving the semantic interoperability problem and illustrated it with heterogeneous router configuration management. The problems due to multiple semantic models, terminologies and meanings are endemic in all areas of IT management due to the tendencies of commercial firms to invent "proprietary" concepts to differentiate their products in the market place. Our ontology-driven solution framework mainly involves a logical model of ontology and a similarity- based ontology mapping strategy. It is a generic solution that is reusable in domains within and across the TMN layers.

This research lays the foundation for a more substantive work in the management of survivable networks and service, especially in the context of the growing threat of cyber-terrorism. Intelligent solutions for these domains will need agents to adapt management strategies based on ontologies of different types of networks and management configurations.

Appendix I
SLD Resolution

Resolution in logic refers to a mechanism in proving statements in first order logic. The mechanism is applied to two clauses in a sentence. Through unification, resolution eliminates a literal that occurs as positive in one clause and negative in the other clause. Unification is the process of identifying the most general unifier (m.g.u) θ, such that θ serves as a substitution that makes two atomic formulae identical. A unifier θ_1 is considered as more general than another unifier θ_2 if for some substitution τ, $\theta_2 = \theta_1\tau$.

Examples of Unification:

1) unifier of $Predicate(x)$ and $Predicate(y)$ is $\{x/y\}$;
2) Unifier of $Predicate(x, y, z)$ and $Predicate(z, z, w)$ is $\{x/y, z/y, w/y\}$.

Consider the scenario of proving $L(C_1) \models L(C_2)$. A logic program is the set of assertions from $L(C_1)$ and a goal G contains the negated assertions from $L(C_2)$. Lets denote the following.

1) $Q_0 : Assertion_1, Q_1 : Assertion_2 \ldots Q_m : Assertion_m$, where $Assertion_{1\ldots m} \in L(C_1)$.
2) $G_0 : \neg L(C_2), G_1 : G_0Q_0\theta_0 \ldots G_n : G_{n-1}Q_{n-1}\theta_{n-1}$, where θ_m is the m.g.u of G_m and Q_m.

SLD resolution can be formally described as:

Given a logic program P in first order language and a goal G. The derivation consists a sequence $G_0, G_1, \ldots, G_m$ of negative clauses from G, associated with a sequence $Q_0, Q_1, \ldots, Q_m$ of variants of clauses from P, and a sequence of substitution $\theta_0, \theta_1, \ldots, \theta_m$. G_i and Q_i resolve into G_{i+1}, and $G_0, G_1, \ldots, G_m$ yields the corresponding computed substitution $\theta_0, \theta_1, \ldots, \theta_m$.

Appendix II
Illustrative Example of the Usage of Concept Similarity Function

Let us consider the example ontology of network resources depicted in Fig. 6. In order to demonstrate as many features of our approach as possible, we strategically select to illustrate the mapping between the concepts "OSPF Routing Table" and "RIP Resource."

Assume that we are interested in three particular semantic aspects during the process of concept mapping, namely class reference, structure and resource type (i.e., Boolean, numeric or list resource). The *class reference* (cr) semantics of two matching concepts C_1 and C_2 can be modeled as the paths from the node that is shared in common by C_1 and C_2 to the nodes C_1 and C_2 themselves. Note that since C_1 and C_2 are part of the paths, during the comparison between the path nodes, C_1 and C_2 (being the predicates) will be recursively compared again according to Definition 3. To avoid such a loop, let p and q be the respective predicates of C_1 and C_2, we assign a value of $S(C_p, C_q) = 1$ if C_p and C_q are equal, otherwise $S(C_p, C_q) = 0$. Furthermore, we induce a heavy similarity penalty on concepts whose common node is the root node by removing any occurrence of the root node from the path. The cr semantics of nodes OSPF Routing Table $(path : \{Network\ Object, OSPF\ Resource, OSPF\ Routing\ Table\})$ and RIP Resource $(path : \{Network\ Object, RIP\ Resource\})$ can be modeled in FOL as: L_{cr} (OSPFRouting Table) $= OSPF\text{-}Resource(y) \wedge OSPF\text{-}Routing\text{-}Table(z)$ and L_{cr} (RIP Resource) $= RIP\text{-}Resource(y)$. The *structural* (st) semantics can be modeled as: L_{st} (OSPF Routing Table) $= Source\text{-}IP(x) \wedge Destination\text{-}IP(y) \wedge \cdots \wedge Metric\text{-}Cost(z)$ and L_{st} (RIP Resource) $=$ *empty*. The *resource type* (rt) semantics might be modeled as: L_{rt} (OSPF Routing Table) $= List\text{-}Resource(x)$, L_{rt} (RIP Resource) $= List\text{-}Resource(a) \vee Numeric\text{-}Resource(b) \vee Boolean\text{-}Resource(c)$.

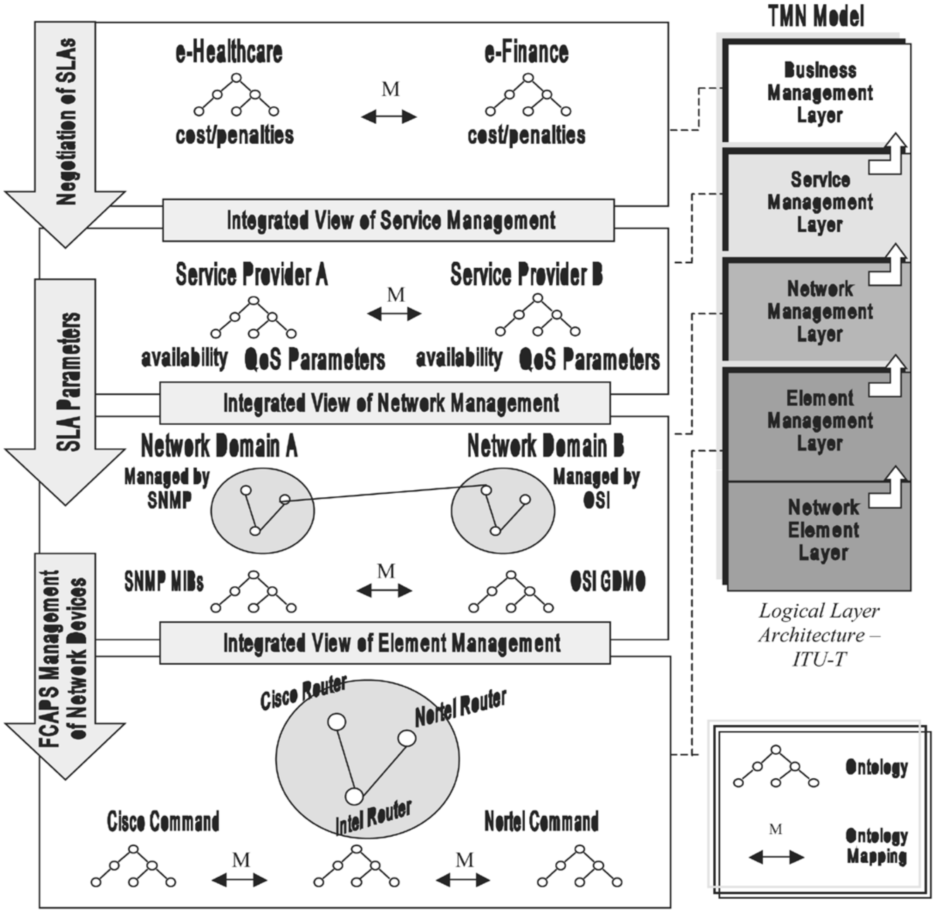


Fig. 5. Ontology mapping driven interoperability solution (TMN view).

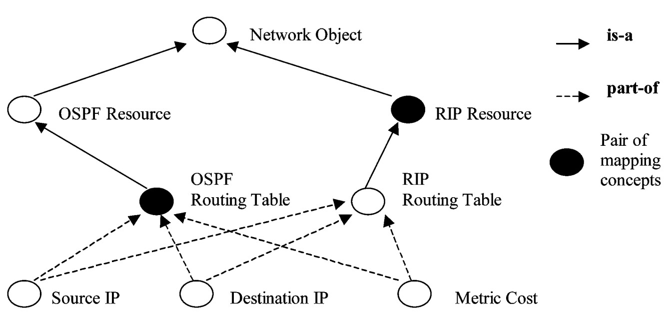


Fig. 6. Example ontology of network resources.

Assume that we choose a weight function $\omega(i) = 1/3$ where $i \in \{$ class reference, structure, resource type $\}$. The similarity between concepts C_1:"OSPF Routing Table" and C_2:"RIP Resource" can then be determined as follows (involve series of similarity computations).

A. Mapping Between Concepts C_1:"OSPF Routing Table" and C_2:"RIP Resource"

Class Reference: $\{L_{cr}(C_1)\} = \{A1: OSPF\text{-}Resource(x), A2: OSPF\text{-}Routing\text{-}Table(y)\}$, $\{L_{cr}(C_2)\} = k\{B1: RIP\text{-}Resource(a)\}$, $\{L_{cr}(C_1)\} \vee \{L_{cr}(C_2)\} = \{A1, A2, B1\}$.

In the direction of $L_{cr}(C_1) \models L_{cr}(C_2)$:

$\partial(B1, A1) = S(C_{B1}, C_{A1}) = S(C_3, C_4) = 0.67$ (see point B below for mapping between C_3 and C_4) Note that $B1$ is resolved by $A1$ and not $A2$ because $\partial(B1, A1) > \partial(B1, A2)$.

In the direction of $L_{cr}(C_2) \models L_{cr}(C_1)$:

$\partial(A1, B1) = 0.67$, $\partial(A2, -) = 0$.

Hence, $S_L(L_{cr}(C_1), L_{cr}(C_2)) = \text{Min}(0.67, 0.67)/3 = 0.223$.

Structure: Since C_2 is a generic concept that does not have a definite structure, the structural semantics is not available. Since nothing can be used to resolve the predicates from $L_{cr}(C_1)$, the resolution quality of the direction $L_{cr}(C_2) \models L_{cr}(C_1)$ becomes 0. Hence, $S_L(L_{st}(C_1), L_{st}(C_2)) = 0$.

Resource Type: $\{L_{rt}(C_1)\} = \{A1: List\text{-}Resource(x)\}$, $\{L_{rt}(C_2)\} = \{B2: List{-}Resource(a) \vee Boolean\text{-}Resource\ (b) \vee Numeric\text{-}Resource\ (c)\}$, $\{L_{rt}(C_1)\} \vee \{L_{rt}(C_2)\} = \{A1, B2\}$.

In the direction of $L_{rt}(C_1) \models L_{rt}(C_2)$:

$\partial(B2, A1) = \text{Max}(\mathbf{\Omega}\, List\text{-}Resource(a), A1), \mathbf{\Omega}\ (Boolean\text{-}Resource(b), -), \mathbf{\Omega}\,(Numeric\text{-}Resource(c), -))$, where $\mathbf{\Omega}(List\text{-}Resource(a), A1) = e^{-Entropy(A1)} * S(List\text{-}Resource\ (a), A1) = e^{-0} * 1 = 1$. Therefore, $(B2, A1) = \text{Max}\ (1, 0, 0) = 1$.

In the direction of $L_{rt}(C_2) \models L_{rt}(C_1)$:

$\partial(A1, B2) = \mathbf{\Omega}(A1, B2) = e^{-Entropy\ (B2)} * S_p(A1, List\text{-}Resource\ (a)) = e^{-1.58} * 1 = 0.2$.

Hence, $S_L(L_{rt}(C_1), L_{rt}(C_2)) = \text{Min}\ (1, 0.2)/2 = 0.2/2 = 0.1$.

Finally, $S(C_1, C_2)$ is computed as $1/3 * 0.223 + 1/3 * 0 + 1/3 * 0.1 = 0.108$.

B. Mapping Between Concepts C_3:"OSPF Resource" and C_4:"RIP Resource"

Class Reference: $\{L_{cr}(C_3)\}=\{A1{:}OSPF\text{-}Resource(x)\}$, $\{L_{cr}(C_4)\} = \{B1 : RIP\text{-}Resource(a)\}$, $\{L_{cr}(C_1)\} \vee \{L_{cr}(C_2)\} = \{A1, B1\}$.

In the direction of $L_{cr}(C_3) \models L_{cr}(C_4)$:

$\partial(B1, A1) = S(C_{B1}, C_{A1}) = 0$ (as mentioned earlier, $S(C_{B1}, C_{A1}) = 0$ if C_{B1} is not equal to C_{A1} to avoid recursive loop).

In the direction of $L_{cr}(C_4) \models L_{cr}(C_3)$:

$\partial(A1, B1) = S(C_{B1}, C_{A1}) = 0$.

Hence, $S_L(L_{cr}(C_3), L_{cr}(C_4)) = \text{Min }(0,0)/3 = 0$.

Structure: Since both concepts do not have a specified structural semantics, we assign a similarity value of $S_L(L_{st}(C_3), L_{st}(C_4)) = 1$ to reflect the irrelevance of the structural semantics during the similarity comparison.

Resource Type: $\{L_{rt}(C_3)\} = \{A1 : List\text{-}Resource(x) \vee Boolean\text{-}Resource(y) \vee Numeric{-}Resource(z)\}$, $\{L_{rt}(C_4)\} = \{B2 : List\text{-}Resource(a) \vee Boolean\text{-}Resource\ (b) \vee Numeric\text{-}Resource(c)\}$, $\{L_{rt}(C_1)\} \vee \{L_{rt}(C_2)\} = \{A1\}$.

Since both $L_{rt}(C_3) \models L_{rt}(C_4)$ and $L_{rt}(C_4) \models L_{rt}(C_3)$ can be deduced, the similarity value $S_L(L_{st}(C_3), L_{st}(C_4)) = 1$.

Finally, $S(C_3, C_4)$ is computed as $1/3{*}0 + 1/3{*}1 + 1/3{*}1 = 0.67$.

References

[1] P. Kalyanasundaram and A. S. Sethi, "Interoperability issues in heterogeneous network management," *J. Netw. Syst. Manage.*, vol. 2, no. 2, pp. 169–193, Jun. 1994.

[2] P. Ray, *Integrated Management From E-Business Perspective*. Norwell, MA: Kluwer, 2003.

[3] D. E. O'Leary, "Using AI in knowledge management: Knowledge bases and ontologies," *IEEE Intell. Syst.*, vol. 13, no. 3, pp. 34–39, May/Jun. 1998.

[4] D. Fensel, F. V. Harmelen, I. Horrocks, D. L. McGuinness, and P. F. Patel-Schneider, "OIL: An ontology infrastructure for the semantic web," *IEEE Intell. Syst.*, vol. 16, no. 2, pp. 38–45, 2001.

[5] J. E. L. Vergara, V. A. Villagra, J. I. Asensio, and J. Berrocal, "Ontologies: Giving semantics to network management models," *IEEE Network, special issue on Network Management*, vol. 17, no. 3, pp. 15–21, May/Jun. 2003.

[6] J. E. L. Vergara, V. A. Villagra, and J. I. Berrocal, "Applying the web ontology language to management information definitions," *IEEE Commun. Mag.*, vol. 42, no. 7, pp. 68–74, Jul. 2004.

[7] Y. Kalfoglou and M. Schorlemmer, "Information-flow based ontology mapping," in *Proc. Cinfederated Int. Conf. DOA, CoopIS and IDBASE*, 2002, pp. 1132–1151.

[8] M. A. Rodriguez and M. J. Egenhofer, "Determining semantic similarity among entity classes from different ontologies," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 2, pp. 442–456, Mar.–Apr. 2003.

[9] A. Doan, J. Madhavan, P. Domingos, and A. Halevy, "Learning to map between ontologies on the semantic web," in *Proc. 11th Int. Con. World Wide Web*, 2002, pp. 662–673.

[10] N. Silva and J. Rocha, "Ontology mapping for interoperability in semantic web," in *Proc. IADIS Int. Conf. WWW/Internet 2003 (ICWI'2003)*, Algarve, Portugal, Nov. 2003, pp. 603–610.

[11] J. Strassner, "Modeling and managing the lifecycle of telecom systems," in *Proc. Int. Conf. Cybern. Inf. Technol., Syst., Applicat.*, 2004, pp. 174–180.

[12] T. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," in *Proc. Padua Workshop Formal Ontology*, Mar. 1993. [Online]. Available: http://ksl-web.stanford.edu/KSL_Abstracts/KSL-93-04.html.

[13] R. Culmone, G. Rossi, and E. Merelli, "An ontology similarity algorithm for BioAgent," in *Proc. NETTAB Workshop Agents Bioinform.*, Bologna, 2002. [Online]. Available: http://www.bioagent.net/WWW-Publications/Download/NETTAB02P1.pdf.

[14] J. J. Jiang and D. W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," in *Proc. Int. Conf. Res. Comput. Linguistics*, 1997, pp. 19–33.

[15] J. Hoppenbrouwers. Semantic modeling. [Online]. Available: infolab.uvt.nl/~hoppie/research/pub/berngv03.pdf

[16] M. Fitting, *First Order Logic and Automated Theorem Proving*, 2nd ed. New York: Springer-Verlag, 1996.

[17] S. Rura and B. S. Lerner, "Flexible static semantic checking using first-order logic," in *Proc. EWSPT*, vol. 2786, 2003, pp. 143–153.

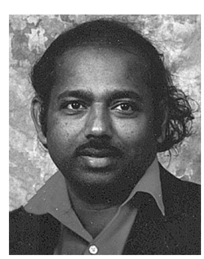
[18] A. K. Y. Wong, N. Paramesh, and P. Ray, "Similarity and logic based ontology mapping for security management," in *Proc. FLARIS*, Clearwater Beach, FL, May 15–17, 2005, pp. 653–659.

Alfred Ka Yiu Wong received the B.S. degree (First Class Honors) in software engineering from the University of New South Wales, Sydney, Australia, in 2004. Currently, he is working towards the Ph.D. degree in computer science at the University of New South Wales.

His research activities began when he was completing his scholarship-based honors thesis with CMCRC in 2003. His research interests include ontology construction, ontology mapping, network management, and in particular, security management.

Pradeep Ray (S'93–M'01–SM'05) is a Senior Member of the Academic Staff in the School of Information Systems, Technology and Management, University of New South Wales, Sydney, Australia. His research interests include the cooperative management of enterprise networks and services in various business areas, such as healthcare and telecommunications. He has more than 90 publications including two research books.

Dr. Pradeep has been the Symposium Chair of the IEEE GLOBECOM 2004 Symposium on Security and Network Management, GLOBECOM 2002 Symposium on Service Infrastructure for Virtual Enterprises (SIVE). He was the coeditor of the *International Journal of Network and Systems Management* (Special Issue on E-Business Management) in March 2003. He has been the Chair of the IEEE Technical Committee on Enterprise Networking (EntNet http://www.comsoc.org/~entnet/) that launched the flagship IEEE annual international event called the International Conference on Enterprise Networks, Applications, and Services in 2001. He is the founder and the Advisory Committee Chair of IEEE Healthcom, an annual event that brings together IT and health/medical sciences professionals and researchers in different parts of the world.



N. Parameswaran is a Senior Lecturer in the School of Computer Science and Engineering, University of New South Wales, Sydney, Australia. He carries out research in the areas related to agent technology and applications in the areas of problem solving in dynamic situations. He is currently involved in the design and implementation of dialog-based agents in enterprise applications.

John Strassner is the founder of Directory Enabled Networking (DEN) Technology. He currently serves as a Fellow of Motorola Laboratories, where he directs their autonomic computing efforts. He also works on identity management, policy management, and seamless mobility. He has authored *Directory Enabled Networks* and *Policy Based Network Management*. He currently chairs the TeleManagement Forum's (TMF) Metamodel, Policy, and Shared Information and Data model working groups, and is an Advisor to the TMF Board of Directors. He is also active in the ITU-T's NGN focus group. He was previously the CSO for Intelliden and a former Cisco Fellow. Currently, he is a Visiting Professor at Waterford's Institute of Technology. He has over 70 refereed conference papers.