

Ontology-based Engineering for Self-Managing Communications

David Lewis, Declan O'Sullivan, Kevin Feeney,
John Keeney, Ruaidhri Power

Knowledge & Data Engineering Group (KDEG),
Centre for Telecommunications Value Chain Research (CTVR),
Department of Computer Science, Trinity College Dublin, Dublin, Ireland.
{Dave.Lewis, Declan.OSullivan, Kevin.Feeney,
John.Keeney, Ruaidhri.Power}@cs.tcd.ie

Abstract. Ontology-based semantics support encoding and mapping between separately authored and thus heterogeneous knowledge, and is expressed in widely accepted standards (e.g. W3C's OWL). It has been suggested that ontology-based semantics will bring benefits to the management of a diversity of systems, ranging from conventional communication services to future autonomic communication services. This paper examines the state of the art in the application of ontological modeling to a range of concerns of interest in the engineering of communication services. In particular the role of ontology modeling for the modeling of services, policies, context, management information and semantic mappings will be examined.

1 Introduction

Within the Semantic Web initiative it has been widely observed that ontological reasoning techniques will only become beneficial once a sufficiently large number of available services have been semantically marked-up. Similarly in the context of management, ontology-driven self management will only be of use for communication systems once services and networks possess ontological representations. To arrive at a situation where ontology-based semantics can be fruitfully employed in network operations, we must first move from the current state of the art in communications management technology which are relied upon by Operational Support Systems (OSS).

For this reason we believe it is timely to review the state of the art with respect to ontological modeling with reference to some key aspects of adaptive communication components that are emerging. In section 2 we overview a reference model for adaptive communication components and use this model as a means to decompose the problem domain and discuss the state of the art in section 3. We present our analysis of the state of ontological modeling under the headings of: modeling services; modeling policies; modeling management

information; modeling context and modeling of semantic mappings. Finally in section 4 we provide some conclusions and future work.

2 A Reference Model for Adaptive Autonomic Components

The move to self-managing systems implies that management decision making is delegated away from the human administrators using manager applications and towards the components being managed. The most common approach to delegating such management decision making is through the use of policy-based management, where a declarative rule that embodies the management decision is executed as close to the managed resource as possible [1].

Thus we have previously proposed that a suitable component model for autonomic systems should combine semantic web services with existing management information semantics and policy rules defining adaptive behavior [2]. Figure 1 below depicts a reference model showing the relationship between the aspects of an adaptive service element.

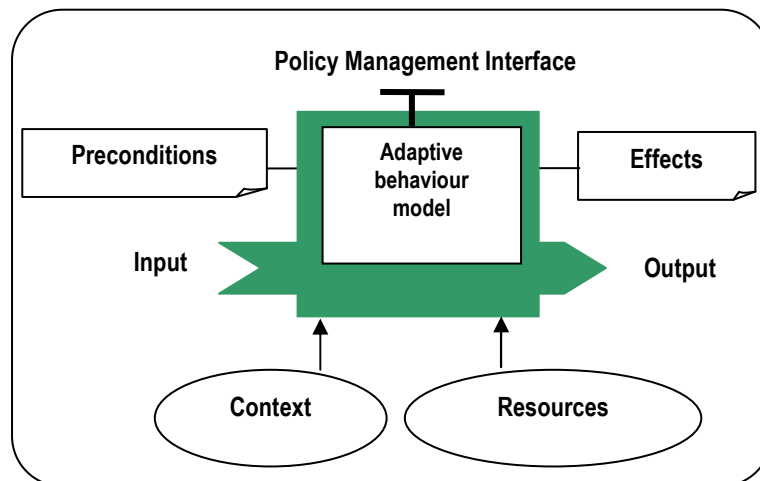


Fig. 1. Reference model for an adaptive service element

Here resource components are managed by presenting their management functions as a semantic web service. It is aware of and controls its own specific set of resource components, which may be modeled as a further set of services. Policies modify the behavior of the service offered by components, based on the state of the component's resources and the external context. An approach where semantics are shared between definitions of the service, the resources, the context and the policies could offer advantages of increased cohesiveness and reduced cognitive load when engineering autonomic systems from such components. We foresee ontologically based modeling of semantics as ideal to

achieve this and in the following section examine the current state of the art of ontology modeling in each of the key aspects of the adaptive communication component reference architecture: modeling services; modeling policies; modeling management information; modeling context and modeling of semantic mappings.

3 Semantic Language Support for Ontological OSS Engineering

This section discusses the current state of the art in ontological languages with respect to the reference model introduced in section 2.

3.1 Modeling Services

The popularity of Service-Oriented Architectures in integrating distributed systems, and the recent standardization of description logic languages for describing ontologies under the World Wide Web consortium's Semantic Web initiative [3] has resulted in intense research into languages for expressing and manipulating Semantic Web Services. These typically aim to integrate with existing web service languages such as the Web Service Description Language (WSDL) and thus aim to exploit the array of WSDL compatible service execution technologies. Semantic Web Service languages typically incorporate composite service features from existing web service languages such as BPEL4WS [4]. This allows them to express complex service interactions between a web service provider and its service consuming client, a modeling approach termed choreography. Alternatively, composite services may be expressed using a business process abstraction which describes how one service is provided by control and data flows between a set of constituent services, each of which may be in turn further decomposed. This latter approach to modeling composite services is termed service orchestration.

Semantic Web Service languages also introduce the modeling of conditional expressions detailing the state of the world in which the service is executed, before and after the services invocation. Conditional expressions can also operate on the knowledge taken in and emitted by a service, i.e. its input and outputs.

Applying ontology-based semantics to web service descriptions offers the possibility of exploiting automated reasoning using off-the-shelf logic engines to assist in service discovery and service composition [5]. This offers the possibility of automating or semi-automating what are currently human-led engineering tasks. For instance in discovering and selecting a service, ontological queries can match terms in a service request to terms in service descriptions that are defined in an ontology to be sub-classes or super-classes of

the requested terms. Alternatively, existing AI planning and situation calculus techniques have been applied to sequential composition of services based on their semantic descriptions.

Currently, as this activity is led by the W3C, web-based e-commerce is seen as the primary profitable application area. Thus there has been relatively little attention paid to the application of semantic service models to non-web services. Nevertheless several researchers have highlighted that the above benefits could also be exploited in other areas where service-oriented solutions are sought amid a level of service heterogeneity. Alternative applications include: pervasive computing [6], telecommunication networks or enterprise networks [7], where hardware and software from multiple vendors need to be integrated rapidly to respond to changing value chain requirements. Such applications require groundings from semantic service descriptions to other service oriented mechanisms than those of the Web, e.g. CORBA, JXTA, or any of the many application layer communication protocols deployed directly on networks, e.g. SS7 or SIP. Fortunately, though much research on ontology-based service semantics focus on WSDL groundings, the languages typically do not exclude grounding to multiple service mechanisms, though few alternatives have been addressed in practice.

These non-web application domains reveal a further interesting requirement for the modeling of semantic services in that they often represent services on specific to certain types of physical devices. These device types are characterized by physical resources, e.g. toner level in a printer or routing tables in an IP router, that typically play a role both in delivering the value of services offered by the device and in administrating the operation of the device by its owner. This differs from the models recently being examined for the management of web services, e.g. by OASIS Web Service Distributed Management Technical committee (www.oasis-open.org), in that these resources are related to the operation of the device rather than to the operation of the Web Service. The increasing tendency to operate web services from sophisticated, commoditized server farms means that the management of the computing and network resources underlying the service is not closely integrated to the semantics of the service itself. Grid technologies, due to the specialized nature of the services offered, sometimes provide a higher level of integration in the view of the service offered and how it is managed [8]. However, when considering the operation of individual devices on a network, the value of the service that device offers is more closely linked to the resources that characterize the device, rather than being a web service using a pool of generic computing resources on a server farm. In other words, the devices in which we are interested offer specific resources that underpin the value provided by the device's service, rather than general purpose computing and storage resources used to delivery a range of web services. The significance of this is that the latter resources can (and are) being standardized, e.g. by Open Grid Service Architecture (www.globus.org/ogsa/) and the Web Services Distributed Management technical committee at OASIS (www.oasis-open.org). However, service-specific device resources will continue to demonstrate a higher degree

of heterogeneity due to their specialized nature. It is therefore important when considering the operation of devices that offer services, that the engineering of the management of that device is closely integrated to the semantics of the service being offered.

Recent years have seen an explosion of research into Semantic Web languages and frameworks. Semantic Web languages were originally intended to provide semantically rich, machine-processible descriptions of web-based content and services. The Semantic Web Service working group of the W3C has identified a number of semantic web service languages and frameworks. Some, such as WSDL-S, simply enable the referencing of external semantic files from within WSDL [9]. This for example allows an ontological description of a service parameter to be defined separately using the W3C's standardized Web Ontology Language (OWL) [10]. Two other approaches that appear to offer a more comprehensive approach to working with semantic web services are OWL for Services (OWL-S) and the Web Service Modeling Ontology (WSMO).

OWL-S

OWL-S is an OWL ontology for describing services, thus reflecting the W3C approach of building more advanced Semantic Web features upon a 'stack' of standards [11]. OWL-S aims to support the automated discovery, invocation, composition and management of web services. It consists of a number of interlinked models:

- the service profile which is used in advertising and selecting web services,
- the service model which is a process-oriented view of how services can be composed (or orchestrated) in a nested manner;
- the grounding model that defines how the ontological service model is mapped onto a concrete communications mechanism (though only a WSDL grounding has been defined to date)
- a resource model offering shared semantics for underlying resources.

OWL-S defines a service in terms of its input and output parameters and in terms of preconditions that must be true before the service is invoked and effects which may become true once the invocation is completed. For the conditional terms, OWL-S requires an additional rule language, and currently allows a number of languages to be used while awaiting the standardization of the Semantic Web Rule Language (www.w3.org/Submission/2005/01/) by the W3C. Similarly, conditional expressions are required in the service model in several of the process flow specification primitives used to define process models, e.g. if-then and while-do control flow constructs.

WSMO

WSMO builds on a previous non-semantic web service framework and is more focused on service discovery and service interoperability [12]. It therefore explicitly includes the modeling of the goals of a service user, against which service offerings are matched. WSMO also includes a range of mediation types that can be used in binding semantic expressions between services, goals, ontologies and groundings. At its most basic WSMO describes services in terms of pre-conditions and post-conditions that apply to information that passes in

and out of the service. It separately defines assumptions and effects, which express pre-invocation and post-invocation conditions that must apply to the environment, or world model, in which the service exists. These expressions are described using ontologies. However, WSMO does not subscribe the W3C stack approach to defining semantic languages, so although its ontology language can be mapped to OWL, it provides direct support for the expression of axioms and rules.

OWL-S vs. WSMO

In modeling complex, composite services, WSMO has focused on supporting choreography, i.e. the externally visible behavior of a service, rather than, as in OWL-S, upon orchestration, where the emphasis is on modeling the internal breakdown of a service into sub processes. The relative merits of OWL-S and WSMO are currently a topic of intense debate in the W3C Semantic Web Services working group and elsewhere. However, there has been little consideration as to how OWL-S and WSMO may be used to integrate semantic service models with semantic models of the resources that underpin them. Though OWL-S attempts to model resources, it is primarily with the aim to managing the sharing of resources between service invocations. It is currently the least developed part of the OWL-S specification, with little guidance and few examples on its use. WSMO does not aim to explicitly model the underlying resource of a service, but it does use the concept of an abstract state machine to model service choreographies. It thus supports state-oriented semantics which we will examine in more detail below for its potential for modeling physical resources.

3.2 Modeling Policies

Typically, in networked devices the management of resources has been handled using the manager-agent paradigm where the resources are modeled as a set of managed objects that can be manipulated by a managing application via a well defined management protocol (e.g. through CMIP, SNMP etc.). Such management information modeling has largely been a manual task requiring good knowledge of the services (typically expressed as communication protocols) offered by the device.

Increasingly, however, the availability of increased computing power on an individual device means that management decision making can be delegated to the device itself, without recourse to remote managing applications, and the architectural centralization and communication overhead that it typically entails. Where the required management actions for the occurrence of a particular operational state, e.g. a partial failure or performance dip, is well understood, the binding between that state and the action that needs to be taken can be encoded in a declarative policy rule, which can be down loaded to the device for local evaluation.

A wide range of rule languages have been defined based upon semantic web languages such as WSML [13], RuleML [14], SWSL [15], SWRL [16], Common Logic [17] and TRIPLE [18]. These are primarily academic efforts; all but RuleML are concerned explicitly with knowledge representation, mainly or only for the Semantic Web (except CL). Rules in these languages can be generally described as taking the form if condition then condition, rather than the standard form of policy rules: if condition then action. The W3C has been gathering data and explore options for establishing a standard web-based language for expressing rules. Its workshop on Rule Languages for Interoperability in April 2005 found “significant interest in establishing a standard language for expressing rules” [19] but the wide range of different requirements for rule languages means that there are important differences between basic concepts as to what a rule should be. Terminology differences make it difficult to even discuss the differences in a clearly defined way. As a result of this workshop, a Rule Interchange Format (RIF) Working Group has been formed in order to investigate the possibilities for defining a format for rules, so they can be used across diverse systems, support a diverse range of requirements and build upon the existing technologies and standards in the area, namely XML, RDF, SPARQL and OWL [19].

Although the W3C efforts to standardize rule languages are only beginning, there have been several proposals for policy based management languages which are constructed on top of Semantic web languages. We describe the policy systems which are based on these languages below.

Rei

Rei [20] is a policy language, originally based on RDF-S, since updated to OWL-Lite, that allows policies to be specified as constraints over allowable and obligated actions on resources in the environment. Rei also includes deontic logic-like variables giving it the flexibility to specify relations like role value maps that are not directly possible in OWL. This deontic-logic-based policy language allows users to express and represent the concepts of rights, prohibitions, obligations, and dispensations. These concepts correspond, respectively, to the conditions of positive and negative authorization, and positive and negative obligation in other policy specification languages. Rei allows users to extend the basic ontology with additional domain dependent ontologies to express concepts and resources that are peculiar to certain domains. For instance, if there is a need to model the specific action of printing a file on a local printer, the general action class of the Rei basic ontology can be customized to include more contextual information about specific printing options. Rei includes meta policy specifications for conflict resolution, speech acts for remote policy management and policy analysis specifications like what-if analysis and use-case management. It is designed for deployment in ubiquitous computing environments and its main goal is to address the issue of governing autonomous entities in constantly evolving distributed environments [21]. The Rei engine, developed in XSB, reasons over Rei policies and domain knowledge in RDF and OWL to provide answers about the current permissions and obligations of an entity, which are used to guide the entity's behavior.

Rei policies are associated with agents, called subjects by means of the has construct: has(Subject, PolicyObject)

The subject of Rei policies can either be a URI identifying an agent or a variable, allowing all agents who satisfy the conditions to be associated with the policy object to possess the policy object. This allows role based or group based policies to be defined by using has with a variable and specifying the role or group, which are application dependent, as part of the condition of the policy object. In this way, policies can be individual, role, group - based, or any combination of the three.

As Rei is designed for highly flexible environments, such as ubiquitous computing, flexible and dynamic mechanisms are important for administration, in particular for distributing permissions throughout the system. Thus, delegation forms a central part of Rei's administration model: there are three types of inter-related rights associated with each action, out of which the last two give certain delegation rights.

- Right to execute : Possessing this right allows the agent to perform the action: has(Agent, right(Action, Condition)), where Action is the action and Condition are the conditions on execution
- Right to delegate execution: If an agent possesses the right to delegate the execution of an action, it can delegate to other agents the right to perform the action, but it cannot perform the action itself. This is similar to the appointments role in the OASIS system described above.
- Right to delegate delegation right: The agent can delegate to another agent or a group of agents the right to further delegate the right to perform the action and delegate this right. This right gives the possessor the right to delegate the previous right, the right to delegate execution and the right to delegate delegation itself.

There are generally two types of delegation, while-delegations and when-delegations. A while-delegation forces all following delegators to satisfy its conditions in order to be true. A when-delegation requires the immediate delegator to satisfy its conditions only at the time of the delegation and not after. For example, consider a when-delegation which gives Jane the right to delegate when she is an employee. All the delegations that Jane made while she was an employee hold even after she leaves. On the other hand, a similar while-delegation will fail once the delegator leaves the company. The while delegation is known as the default delegation type and is suitable for the temporary transfer of access rights from one individual to another in order to fulfill a specific task, similar to the concept of delegation in Ponder, described above. When delegation, on the other hand is equivalent to administration of the system, since the rights delegated through this type of delegation are permanent.

KAoS

KAoS [22][23], like Rei, incorporates semantic web languages, in this case being entirely specified in DAML. KAoS is divided into policy and domain services and was originally designed for constraining the behavior of agents in a wide variety of operational settings. The KAoS Policy Ontologies (KPO) define basic ontologies for actions, actors, groups, places, various entities related to

actions (e.g., computing resources), and policies. The actor ontology distinguishes between people and various classes of software agents that can be the subject of policy. Groups of actors or other entities may be distinguished according to whether the set of members is defined extensionally (i.e., through explicit enumeration in some kind of registry) or intentionally (i.e., by virtue of some common property such as a joint goal that all actors possess or a given place where various entities may be currently located). This allows a variety of different grouping mechanisms to be applied to the agents.

A KAoS policy is a statement enabling or constraining execution of some type of action by one or more actors in relation to various aspects of some situation. In DAML, a policy is represented as an instance of the appropriate policy type (i.e., positive or negative authorization, positive or negative obligation) with associated values for properties: priority, update time stamp and a site of enforcement. The most important property value is, however, the name of a controlled action class. Usually, a new action class is built automatically when a policy is defined. Through various property restrictions, a given policy can be variously scoped, for example, either to individual agents, to agents of a given class or to agents belonging to a particular group, etc. Additionally, action context can be precisely described by restricting values of its properties. KAoS is administered with the KAoS Policy Administration Tool (KPAT), a graphical interface which hides the complexity of the DAML policy representation from users. However, access control to administration of the system is not modeled by the system.

Overall, both Rei and KAoS provide considerable flexibility in terms of the grouping abstractions that can be incorporated into them and both are extremely expressive – a consequence of the inherent flexibility and extensibility of semantic web languages. An ontology-based description of the policy enables the system to use concepts to describe the environments and the entities being controlled, thus simplifying their description and facilitating the analysis and the careful reasoning over them. In addition, ontology-based approaches allow the possibility of dynamically calculating relations between policies and environment, entities or other policies based on ontology relations rather than fixing them in advance. It is possible to access the information provided by querying the ontology according to the ontology schema. This is an advantage in comparison to traditional languages that provide only pre-defined queries to access information and static representations of policy. As they are designed for the interchange of semantic information between autonomous domains, ontologies can simplify the sharing of policy knowledge thus increasing the possibility for entities to negotiate policies and to agree on a common set of policies in heterogeneous policy environments. On the other hand, the semantic web languages used for ontology representation still present a complex syntax, long declarative description, and hyperlinks and references to external resources that make the code very difficult to read, even more so than XACML. Furthermore, the high level nature of these languages can mean that the policies specified in them can be difficult to implement – a process which can not be

entirely automated, but requires some human programming to translate policies into the particular capabilities of the target platform.

From the point of view of the organizational model embodied in the policy system, the flexibility of ontologies allows us to use practically any structural model of the organization. Both Rei and KAoS support essentially arbitrary grouping abstractions, however, the flip-side of this flexibility is that the onus for maintaining a particular consistent approach to modeling the organization of resources and autonomous agents (or users) in the system falls onto the administrators. Rei provides a flexible delegation mechanism that allows this administration to be distributed through the organization; however, this flexibility comes at the expense of the complexity of constraining the propagation of rights through the organization through policies. Tonti et al. [24] provide a comparison of Rei, KAoS and Ponder.

3.3 Modeling Management Information

The predominant paradigm in network management has been the manager-agent model. Here, the OSI Management and Internet Management represent the two main standards bodies, using the GDMO and SMI languages respectively. Both of these languages, though being potentially generic profiles of ASN.1, were shaped in their usage by the features of the protocols that accompanied them, CMIP and SNMP respectively. In the 1990s the Distributed Management Task Force defined the Common Information Model schema that was a principled attempt to define management information models for the manager-agent paradigm, but in a way that was independent from the protocol used. This proved successful, quickly becoming a focus for management information modeling standardization effort, especially in the enterprise management sphere, with support added for a number of protocol bindings including DCE, XML/HTTP and LDAP. The modeling approach was highly object-oriented, yet also incorporated a number of ontological modeling concepts, such as making associations first class concepts with domain and range bindings to classes and allowing class and instance definitions to be freely mixed. More recently Jorge de Vergara and Victor Villagra [25] have shown directly the value of modeling management information models in OWL, and how this can be used to ease the interoperation between models originally conceived in different MIB languages, i.e., GDMO, SMI, CIM.

In parallel, the engineering of service and business layer OSSs for the telecommunication market began to adopt the service-oriented and n-tier component architectures that had come to dominate enterprise computing. At the forefront of attempts to reach industry agreement on modeling such architectures for communications management was the TeleManagement Forum's NGOSS initiative [26]. This is attempting to stimulate an open market in telecoms business software component by forming agreements on management information exchanged between business processes and service definitions, via which inter-process invocations can be made. The former

encompasses network and element level MIB information as well as service and business level information typically captured in corporate databases. Such business objects also increasingly become the subject of business-to-business e-commerce agreements, e.g. ebXML. This has a natural synergy with the enterprise management model of the DMTF, and the two organizations are now collaborating closely on information modeling. The NGOSS initiative seems ripe for an ontological approach, provided suitable methodologies and tools emerge [7].

3.4 Modeling of Context

Strang and Linnhoff-Popien survey the multitude of context modeling approaches in [27]. They classify these models into the following groups. **Key Value** models are the simplest form of markup each particular context attribute is represented as a key, and the application simply reads the value associated with that key to retrieve the result. **Markup scheme** models are hierarchical models, defining both attributes and content for each tag. Because of this, they are often expressed in XML or another SGML variant. **Graphical** modeling applies existing modeling approaches such as UML and Object-Role Modeling (ORM) to context information. The generic nature of these approaches allows them to be easily extended to include features such as dependencies between context facts. **Object-oriented** models apply the traditional benefits of object-oriented software design such as encapsulation and reusability to context. Objects are used to hide the details of context acquisition, exposing a context interface at an abstraction level that is useful to an application. This allows these context objects to be upgraded in the future without affecting the applications that use them. **Logic based** models represent context as a set of facts and concluding expressions that are true in the environment. A formal system is used to apply rules which allow additional facts and expressions to be derived. Finally, **ontology-based** models also exploit formal models of context to express concepts and relations between them.

Of the published ontology models for context, clearly the most influential is SOUPA [28]. The SOUPA project began in November 2003 as part of the Semantic Web in Ubicomp Special Interest Group. The SOUPA ontologies are freely published online, and are frequently cited as a good example of ontologies for context. The SOUPA ontologies consist of two sets of separate but interlinked ontologies that form SOUPA core and SOUPA extension. While SOUPA core is used to model fundamental concepts such as Person, Action, Space and Time, SOUPA extension models higher-level concepts such as Schedule, Meeting, Contact Preference and Conditional Belief. These ontologies are designed to be used separately if required, so that application developers may choose to make use of only some of the ontologies in their application, to reduce complexity. The designers of SOUPA elected to borrow terms from other ontologies, but not to import them directly. SOUPA references terms from a number of ontologies such as the Friend-Of-A-Friend ontology (FOAF) [29],

the spatial ontologies in OpenCyc [30], COBRA-ONT [31] and the MoGATU BDI ontology [32]. The SOUPA ontologies have been used in a number of projects, for example Fuchs et. al. [33] describe an implementation of an intelligent answering machine application whose ontology maps to the SOUPA ontology and the FOAF ontology [29], allowing interoperation with other applications which also map to these ontologies.

SOUPA has also been extended in the CoBrA-ONT ontology [31] (by the same authors) to allow the CoBrA system to manage smart meeting rooms. The CoBrA-ONT ontology defines some of the common relationships and attributes that are associated with people, places and activities in a pervasive computing space. The CoBrA system uses this ontology, along with instance data provided to it, to answer questions such as “Is X currently in a meeting place in building Y?” or “Is X the speaker of meeting Z?”. At the highest level, the CoBrA-ONT ontology describes “Person”, “Place” and “Intention”. For example, the Person class defines properties of people such as their name and e-mail address, while more specific subclasses of Person such as “Speaker” or “PersonInBuilding” are used to define additional properties such as the building the person is in. The “Intention” class defines user intentions such as the speaker’s intention to give a talk. This class is defined as the union of all its subclasses, as it is the collection of all defined user intentions.

Independently from SOUPA, the MoGATU BDI ontologies [32] are used in the MoGATU system, and have a slightly different focus. The Agent ontology represents human users or intelligent software entities. Agents can express their Beliefs, Desires, Intentions and Goals. The statements that express their beliefs can be unconditional statements, which always hold true or conditional statements which are asserted if a particular condition statement is true. An agent’s desires can conflict with other desires, but its goals are a set of non-conflicting achievable desires. An agent can assert a set of intentions, the actions it will perform to achieve its goals. These actions can be combined into an ordered sequence which is called a plan. Agents can express the priority of these desires, goals, intentions, etc. using a weighting system. The MoGATU ontologies also allow the expression of time, both time instants and time periods. This allows the agent to specify when and how often to initiate an action plan. By reasoning over the ontology information available, the agent can take intelligent action based on the current state of its environment and the other agents (human or software) taking part in it.

The CoDAMoS ontology [34], as presented by Preuveneers et. al., is another generic upper-level ontology which aims to provide a basis for the most important aspects of context information. It defines a User (which contains a user’s preferences, profile and current activity), Environment (containing time and location information as well as environmental conditions such as lighting), Platform (a hardware and software description of a device) and Service (software which provides a service to a user). User tasks can be broken down into the activities that the user performs in order to accomplish the task. As well as tasks, Users have Profiles which express static information about them, Roles which express the kind of actions they perform, and Mood stores extra more

dynamic information such as current communication preferences. The Platform section of the ontology provides a description of the software that is available on the device for the user and other services to interact with, and the hardware resources of the device. This includes elements such as the software installed on the device, the operating system used, the virtual machines available to execute software and so on. Each Platform is part of an Environment. The environment specifies physical properties such as its location and environmental condition (temperature, humidity, etc.)

Some less-cited models include SOCAM and CONON. Gu et al. present a Service-Oriented Context-Aware Middleware (SOCAM) [35] based on a context model with person, location, activity and computational entity (such as a device, network, application, service, etc.) as basic context concepts. The Context Ontology (CONON) [36] is an upper-level ontology for context which is designed to be extended with domain-specific ontologies for particular tasks. Location, user and activity are taken as the most fundamental elements of context. Computational entities are also considered as first-class elements of the top-level ontology. The authors use data modeled with this ontology to reason about facts such as the location of users.

Finally, the Context Ontology Language (CoOL) [37] is an ontology-based context modeling approach, rather than a particular model. It uses the Aspect-Scale-Context (ASC) model where each aspect (e.g. distance) can have several scales (e.g. meter scale or feet scale) to express some context information (e.g. 10). Mapping functions exist to convert context information from one scale to another.

3.5 Modeling Semantic mapping

The promise of ontologies is in the sharing of an understanding of a domain that can be communicated between people and application systems [38]. However, ontologies are defined from a particular perspective. Web service sequencing or composition is a typical example of where this difference in perspective is causing a problem. WSMO and OWL-S attempt to annotate web services with more semantic information so that they can be more easily used and discovered. Realistically however, an annotation of any one web service will take place from a particular perspective and using a particular selection of ontologies. Thus there is still a need to reconcile these different perspectives so that combined use of the ontologies can be achieved when composing or sequencing web services drawn from several sources together. Ontology mappings are seen as the way in which such reconciliation and combination can be enabled.

Mappings between elements in ontologies are usually expressed as pairs of related entities in some mapping expression. This mapping expression can range over simple equivalences and complex correspondences. An example of a simple equivalence is where Paper in the one ontology could be considered equivalent to ConferencePaper in a second ontology. An example of a complex correspondence would be that a has-page-numbers property of one ontology is

equivalent to the lastPage property minus the firstPage property of a second ontology. These mapping expressions are normally output as a separate document. The advantage of a separate document for the mappings is that mappings can be managed independently of the ontologies. Most state of the art mapping systems express mappings in a proprietary format typically aligned with the technology used by the mapping system. This is one reason why direct comparison of ontology mapping tools has been a difficult exercise [39]. For example, the OntoMerge system [40] uses bridging axioms written in first order logic language to express the translation rules between the concepts in the ontologies, and then runs a theorem prover optimized for ontology translation over the ontologies and the axioms. Another example is the MAFRA system [41] that includes a formal representation to specify the mappings. The formalism that is used to describe the Semantic Bridges is based on an ontology specified in DAML+OIL, called the Semantic Bridging Ontology (SBO).

Increasingly the need for an open mapping format is being recognized and proposals have begun to emerge [42][43][44][45]. For example, XML based formats to enable comparison of the output of a variety of matching tools were developed for the I3CON contest [46] and EON contest [47]. In order to participate, the entrant systems needed to adapt their output to a given mapping format. Systems from Lockheed Martin, AT&T, Teknowledge, INRIA and University of Karlsruhe took part in the I3CON contest. Systems from Stanford University Fujitsu, INRIA, University of Montreal and University of Karlsruhe took part in the EON contest. Experience from these contests proved positive [48] and led to the development of the INRIA ontology alignment format [43]. The format can also be rendered into different formats (SWRL, OWL etc.) for the purposes of interpretation. In contrast deBruijn et al. [44][45] have proposed a generic mapping language that must be grounded in a declarative logical language and thus requires a reasoner. Initial groundings to OWL (Description Logic-based language) and WSML-Flight (a Logic Programming-based language) have been developed.

It is useful that the research community has begun over the last few years to address the issue of a common way to specify the results of matching algorithms and/or mapping systems. Unfortunately it is too early to determine whether one of the two prominent contenders (that is from INRIA and from deBruijn et al.) will emerge as the basis of a standard format, whether another will be proposed or whether the common Rule Interchange Format (RIF) emerging from the W3C might be sufficiently expressive. The advantage of the INRIA format is that it can be used for representing results of match algorithms and results of mappings, which can be rendered into different mapping languages. The advantage of the deBruijn et al. format is that it has a formal basis. What is clear is that further and wider evaluations of the formats are required and that several issues remain to be addressed. One such issue is the manner in which strength/similarity/confidence in a match or mapping should be expressed. This is particularly important when combining the results of matchers from different vendors together or when sharing mappings between systems. Another issue that needs to be explored is whether a match or a mapping can be annotated with

information that indicates whether or not the match/mapping is valid for particular application contexts. Another key issue that has started to be explored is the efficient sharing of mappings, with peer to peer approaches [49] and content based network approaches [50] both showing promise. Finally an issue that has yet to be explored is the issue of integrating mappings that have been shared, into a node such that conflicts can be identified and opportunities for new mappings based on transitive relationships can be examined. In summary, the desire for a common format to express ontology matches/mappings in a manner that would be open to rendering into specific system or technology formats has only recently gained momentum.

5 Conclusions and Further Work

It is clear from the state of the art and our own work in the highlighted areas, that the application of ontology modeling holds promise as we move more towards systems that exhibit self-managing behavior, including networking [51], pervasive computing [52], and distributed system [53] environments.

As the application of ontology modeling begins to gain more widespread acceptance, the research challenge is beginning to move towards issues related to engineering of ontology based systems in performance demanding communication environments. There are a wide range of issues still to be resolved, some of which we have already started to address, such as: how to benchmark ontology-based systems [54]; what is the performance of ontology based reasoners for communications intense environments [55][57]; how can ontology mappings bridge management information heterogeneity [56]; and how to integrate policy based directives into semantic web services using existing language features [57].

Acknowledgements

This work has been partly sponsored by Science Foundation Ireland through the Centre for Telecommunications Value Chain Research, and partly by the Irish Higher Education Authority through the M-Zones program.

References

1. Strassner, J., (2004) "Policy-based Network Management – Solutions for the Next Generation", Elsevier.
2. Semantic Interoperability for an Autonomic Knowledge Delivery Service, D. Lewis, D. O'Sullivan, R. Power, J. Keeney, in Proc of 2nd IFIP WG6.6 International

- Workshop on Autonomic Communication - Autonomic Communication Principles, Oct. 3-5, 2005, Vouliagmeni, Athens, Greece.
3. Berners-Lee, T., Hendler, K., Lassila, O. (2001), 'The Semantic Web', Scientific American, pp 35-43, Issue 284 (3), 17th May 2001.
 4. Business Process Execution Language for Web Services, version 1.1, May 2003.
 5. McIlraith, S.A., Son, T.C., Honglei Zeng, H. (2001), 'Semantic Web Services', IEEE Intelligent Systems, 16(2), March/April 2001.
 6. Masuoka, R., Labrou, Y., Parsia, B., Sirin, E., "Ontology-Enabled Pervasive Computing Applications", IEEE Intelligent Systems, Sept/Oct 2003, pp 68-72.
 7. Duke, A., Davies, J., Richardson, M., Kings, N., "A Semantic Service Oriented Architecture for the Telecommunications Industry", in proc of IFIP Int'l Conf on Intelligence in communications Systems, Bangkok, Thailand, Nov 2004 (INTELLICOM 2004), Springer LNCS 3283, pp 236-245.
 8. Foster, I., Kesselman, C., Nick, J., and Tuecke, S., "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration." Open Grid Service Infrastructure WG, Global Grid Forum, June 2002.
 9. Akkiraju, R, et al, "Web Service Semantics - WSDL-S", W3C Member Submission 7 November 2005, Version 1.0.
 10. OWL Web Ontology Language Reference, available at: <http://www.w3.org/TR/owl-ref/>.
 11. Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N., Sycara, K., "OWL-S: Semantic Markup for Web Services", W3C Member Submission 22 November 2004
 12. D3.1v0.1 WSMO Primer, WSMO Final Draft April 2006.
 13. de Bruijn, J. Lausen, H., Krummenacher, R., Polleres, A., Predoiu, L., Kifer, M., Fensel, D. "The Web Service Modeling Language WSML." WSML Deliverable D16.1v0.2, 2005. <http://www.wsmo.org/TR/d16/d16.1/v0.2/>
 14. Grosz, B., Poon, T. "Representing agent contracts with exceptions using xml rules, ontologies, and process descriptions." In RuleML-BR-SW'02, 2002.
 15. Grosz, B., Kifer, M., Martin, D. L. "Rules in the Semantic Web Services Language (SWSL): An Overview for Standardization Directions." Position Paper for W3C Workshop on Rule Languages for Interoperability. Available at: <http://www.w3.org/2004/12/rules-ws/paper/124/>.
 16. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosz, B., Dean, M. SWRL: "A Semantic Web Rule Language Combining OWL and RuleML." W3C Member Submission 21 May 2004. Available from <http://www.w3.org/2004/SUBM-SWRL-20040521/>.
 17. ISO/IEC JTC 1/SC 32 N xxxx 2006-06-01 ISO/FDIS 24707 ISO/JTC 1/SC 32/WG2 ANSI "Information technology - Draft Standard - Common Logic (CL) - A framework for a family of logic-based language".
 18. Sintek, M., Decker, S. "TRIPLE - A Query, Inference, and Transformation Language for the Semantic Web." First International Semantic Web Conference 2002: 364-378.
 19. Hawke, S., Tabet, S., de Sainte Marie, C. "Rule Language Standardization Report from the W3C Workshop on Rule Languages for Interoperability", April 2005 <http://www.w3.org/2004/12/rules-ws/report/>
 20. Kagal, L., Finin, T., Joshi, A., "A Policy Language for A Pervasive Computing Environment", IEEE 4th International Workshop on Policies for Distributed Systems and Networks, June 04, 2003.

21. Patwardhan, A., Korolev, V., Kagal, L., Joshi, A. "Enforcing Policies in Pervasive Environments" In proc International Conference on Mobile and Ubiquitous Systems: Networking and Services, August 2004.
22. Bradshaw, J., Uszok, A., Jeffers, R., Suri, N., Hayes, P., Burstein, M., Acquisti, A., Benyo, B., Breedy, M., Carvalho, M., Diller, D., Johnson, M., Kulkarni, S., Lott, J., Sierhuis, M., and Van Hoof, R., "Representation and reasoning for DAML-based policy and domain services in KAOs and Nomads." In proc of the Autonomous Agents and Multi-Agent Systems Conference (AAMAS 2003). Melbourne, Australia, New York, NY: ACM Press, (2003).
23. Uszok, J. Bradshaw, R. Jeffers, N. Suri et al.: "KAoS Policy and Domain Services: Toward a Description-Logic Approach to Policy Representation, Deconfliction, and Enforcement". In proc of IEEE 4th International Workshop on Policies for Distributed Systems and Networks (POLICY 2003), Lake Como, Italy, (2003).
24. Tonti, G., Bradshaw, J.M., Jeffers, R., Montanari, R., Suri, N., Uszok, A., "Semantic Web Languages for Policy Representation and Reasoning: A Comparison of KAOs, Rei, and Ponder". In proc of 2nd International Semantic Web Conference (ISWC2003), October 20-23, 2003, Sanibel Island, Florida, USA.
25. López de Vergara, J.E., Villagrà, V.A., Berrocal, J., "Applying the Web Ontology Language to management information definitions", IEEE Communications Magazine, Vol. 42, Issue 7, July 2004, pp. 68-74. ISSN 0163-6804.
26. Fleck, J, "An Overview of the NGOSS Architecture", TeleManagement Forum Whitepaper, May 2003.
27. Strang T., and Linnhoff-Popien, C., "A context modeling survey". In proc of Workshop on Advanced Context Modeling, Reasoning and Management as part of UbiComp 2004, September 2004.
28. Chen, H., Finin, T., and Joshi, A., "The SOUPA Ontology for Pervasive Computing." Whitestein Series in Software Agent Technologies. Springer, July 2005.
29. Brickley, D., and Miller, L., "Foaf vocabulary specification", <http://xmlns.com/foaf/0.1/>, January 2006.
30. Lenat, D., and Guha, R. V., "Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project". Addison-Wesley, February 1990.
31. Chen, H., Finin, T., and Joshi, A., "An ontology for context-aware pervasive computing environments." Knowledge and Engineering Review, Special Issue on Ontologies for Distributed Systems, 2003.
32. Perich, F., "Mogatu BDI ontology", <http://mogatu.umbc.edu/bdi/>, January 2004.
33. Fuchs, F., Hochstatter, I., Krause, M., and Berger, M., "A metamodel approach to context information." In PERCOMW '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops, pages 8–14. IEEE Computer Society, 2005.
34. Preveneers, D., Van den Bergh, J., Wagelaar, D., Georges, A., Rigole, P., Clerckx, T., Berbers, T., Coninx, K., Jonckers, V., and De Bosschere, K., "Towards an extensible context ontology for ambient intelligence." Lecture Notes in Computer Science: Second European Symposium on Ambient Intelligence., vol 3295:148–159.
35. Gu, T, Wang, X. H., Pung, H. K., and Zhang, D. Q., "An ontology-based context model in intelligent environments." In proc of Communication Networks and Distributed Systems Modeling and Simulation Conference, San Diego, California, USA, January 2004.
36. Wang, X. H., Gu, T., Zhang, D. Q., and Pung, H. K., "Ontology-based context modeling and reasoning using owl". In proc of Modeling and Reasoning Workshop at PerCom 2004, 2004.

37. Strang, T., Linnhoff-Popien, C., and Frank, K., "CoOL: A Context Ontology Language to enable Contextual Interoperability". In Jean-Bernard Stefani, Isabelle Dameure, and Daniel Hagimont, editors, LNCS 2893: Proceedings of 4th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems (DAIS2003), volume 2893 of Lecture Notes in Computer Science (LNCS), pages 236–247, Paris/France, November 2003. Springer Verlag.
38. Fensel D., "Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce", 2nd edition, Springer-Verlag, Berlin, 2003.
39. Noy N. and Musen M., "Evaluating ontology-mapping tools: requirements and experience", Proceedings of 1st Workshop on Evaluation of Ontology Tools (EON 2002), EKAW '02, 2002.
40. Dou D., McDermott D., Qi P., "Ontology translation by ontology merging and automated reasoning", Proceedings of EKAW2002 Workshop on Ontologies for Multi-Agent Systems, pages 3-18, 2002.
41. Maedche A., Motik B., Silva N., Volz R., "MAFRA - A Mapping FRamework for Distributed Ontologies in the Semantic Web", Workshop on Knowledge Transformation for the Semantic Web (KTSW 2002), ECAI 2002, pages 60-68, Lyon, France, 2002.
42. Euzenat J., "An API for ontology alignment", International Semantic Web Conference (ISWC 2004), LCNS 3298, pages 698-712, Springer, Berlin, Germany, 2004.
43. Euzenat J., "An API for ontology alignment (version 1.3)", Available at <http://co4.inrialpes.fr/align/align.pdf>, June 2005. Last visited: August 2005.
44. deBruijn J., Martin-Recuerda F., Ehrig M., Polleres A., Predoiu L., "Ontology Mediation Management", IST SEKT project deliverable, 4.4.1, February 2005.
45. deBruijn J., Foxvog D., Zimmerman K., "Ontology Mediation Patterns Library", IST SEKT project deliverable, 4.3.1, February 2005.
46. EON 2004 contest, Proceedings of the 3rd Evaluation of Ontology-based tools (EON), Available at: <http://km.aifb.uni-karlsruhe.de/ws/eon2004/>, Last visited: August 2005.
47. I3con contest, NIST Performance Metrics for Intelligent Systems (PerMIS) Workshop, August 2004, <http://www.atl.external.lmco.com/projects/ontology/i3con.html>, Last visited: August 2005.
48. Euzenat J., "Evaluating ontology alignment methods", Semantic Interoperability and Integration, Dagstuhl Seminar Proceedings, Germany, September, 2004.
49. Conroy C., "Wildflower: P2P Sharing of Ontology Mappings", M.Sc. Dissertation, Trinity College Dublin, May 2005.
50. Lynch D., "A Proactive Approach to Semantically Oriented Service Discovery", M.Sc. Dissertation, Trinity College Dublin, September 2005.
51. Smirnov, M., "Autonomic Communication: Research Agenda for a New Communications Paradigm", Fraunhofer FOKUS, November 2004. http://www.fokus.gmd.de/web-dokumente/Flyer_engl/Autonomic-Communicatin.pdf
52. McCann, J. A., and Huebscher, M. C., "Evaluation issues in autonomic computing". In Proceedings of Grid and Cooperative Computing Workshops (GCC), LNCS 3252, 597-608. Wuhan, China. October 21-24, 2004.
53. Kephart, J., Chess, D., "The Vision of Autonomic Computing", IEEE Computer, Jan 2003, pp 41-50.

54. Lewis, D., O'Sullivan, D., Keeney, J., "Towards the Knowledge-Driven Benchmarking of Autonomic Communications", in proc of the Second International IEEE WoWMoM Workshop on Autonomic Communications and Computing (ACC 2006), Niagara-Falls / Buffalo, New York, USA, 26 June 2006.
55. Keeney, J., Lewis, D., O'Sullivan, D., Roelens, A., Boran, A., Richardson, R., "Runtime Semantic Interoperability for Gathering Ontology-based Network Context", in proc of the IEEE/IFIP Network Operations and Management Symposium (NOMS 2006), Vancouver, Canada. 3-7 April 2006. pp56 - 66
56. Lewis, D., Keeney, J., O'Sullivan, D., "Policy-based Management for Resource-Specific Semantic Services", in proc of the 1st Annual Workshop on Distributed Autonomous Network Management Systems, Co-located with the IEEE International Conference on Autonomic Computing (ICAC 2006), Dublin, Ireland, 16 June 2006.
57. Lewis, D., Keeney, J., O'Sullivan, D., Guo, S., "Towards a Managed Extensible Control Plane for Knowledge-Based Networking", to appear in proc of the 17th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management Large Scale Management, (DSOM 2006), at Manweek 2006, Dublin, Ireland, 23-25 October 2006.