Du Contrôle de La Collaboration Humaine Vers Des Contrats de Service Sémantiques Pour La Sécurité de l'information

Gloria Elena Jaramillo, Manuel Munier, Philippe Aniorte

From Human Collaboration Control to Semantic Semantic Service Contracts for Information Security

Gloria Elena Jaramillo[1], Manuel Munier[1], Philippe Aniorte[1]

1. Laboratiore d'Informatique de l'Université de Pau et des Pays de l'Adour
371 rue du Ruisseau – BP 201
{gloriaelena.jaramillorojas, manuel.munier, philippe.aniorte}@univ-pau.fr

RÉSUMÉ. L'architecture orientée services (SOA) est considérée comme le paradigme le plus prometteur au cours des dernières années pour fournir des fonctionnalités et faciliter la coopération commerciale. Dans le SOA, la vision traditionnelle de la sécurité vise à garder des propriétés telles que la disponibilité, l'authenticité et la confidentialité, en protégeant le service Web lui-même. Cependant, dans une telle approche les particularités de l'interaction humaine en ce qui concerne les comportements des parties prenantes de service ont été jusqu'à présent basée seulement sur la confiance. Dans cet article, justifié par la nécessité d'avoir un accord dans le vocabulaire commercial utilisé pour exprimer les politiques de contrôlabilité, nous présentons un modèle formalisant la sémantique des contrats de service. Le formalisme DL est utilisé pour modeler le domaine de connaissance, alors que OWL 2 est utilisé comme la syntaxe concrète.

ABSTRACT. The service-oriented technologies are considered as the most promising paradigm over the last few years for delivering functionalities and allowing business cooperation. In those paradigms, the traditional vision of security aims to keep properties such as availability, authentication, and confidentiality by protecting the web service itself. However, in such an approach, the particularities of the human interaction in regard to the behaviors of the service stakeholders have been until now based only on trust. In this article, justified by the need of having an agreement in the business vocabulary used for expressing controllability policies, we present a model formalizing the semantics of service contracts. The DL formalism is used to model the specific knowledge domain, while OWL 2 is used as concrete syntax.

Mots-clés : contrat, sémantique, modèle, service, SOA, confiance.

KEYWORDS: contract, semantics, model, service, SOA, trust.

DOI:10.3166/RCMA.25.1-n © 2016 Lavoisier

1. Introduction

The dynamics of global markets, free trade agreements and domestic policies of the different countries have changed the way of doing business. In the last years, to compete effectively in the market, organizations have needed to create competitive advantages, which have been usually reflected in a dichotomy between costs and service quality. Enterprises are interested in consolidating clients, offering better services than competitors and expanding their market, but at the same time, they want to reduce costs and increase their profits. In such a context, organizations realized that they cannot work on its own and targeted the collaboration strategies as a key element for accomplishing their business objectives. So, it is not rare that internal policies within organizations have increasingly sought to focus on the core of their business goals while collaborating with external partners by delegating or outsourcing secondary activities.

From the technological point of view, a such interaction has led to changes in the architecture and design of the information systems, which increasingly need to be adaptable, modular and flexible to support different environments, users and the dynamism of the global market tendencies. Given this framework, the Service-based Technologies (SBT) have been proposed as an effective solution to those challenges. On the one hand, thanks to its properties of transparency, integration and loose coupling, the Service Oriented Architectures (SOAs) meet the managerial requirement of a rapid adaptation of the information systems according to the market dynamics. It allows organizations a constant innovation of their processes in terms of the delivered commodity and the partners used. On the other hand, the Cloud aims the virtualization of the technological resources, which allows its users ignoring infrastructure details when using services. So, cloud users can storage, process, create and share data and applications regardless the memory or storage capacity of host devices, which are managed by the Cloud provider.

Basically, in both SOA and the Cloud, new paradigms of interaction between clients and providers mediated by the services are created based on dynamism and trust, where the former is reflected in its properties of elasticity, integration and (semi-) automatic composition, while trust becomes a key aspect to lay the foundations of the relation. Indeed, due to the fact that the external partner is an independent organization, its internal processes are considered as a black box. Therefore, no control is possible, which compels each organization to trust in the behaviors of the external partner. In our work, we define this as the problem of the lack of control. Despite the fact that in the service-oriented technologies, each partner has the freedom of the way in which he internally performs the service, some degree of control is required to guarantee that the external partner behaves as expected and hence to prevent damages due to misbehaviors.

The idea behind our work is to be able to formalize the rules of collaboration that actors must respect. These rules go far beyond simply identifying the access rights or the exchanged data (parameters when invoking the service and the result of the execution of the service). We aim at taking into account aspects related to usage control: metadata of the data provided by the client (constraints, business quality indicators, certificates of conformity, and the like), expected metadata associated to

the returned result, business restrictions (such as subcontracting or allowed partial results). Thus, these contracts define security policies at the organizational level. In the future, the idea will be to design a platform to analyze and get feedback of the stakeholder interactions. It includes to determine the state of each rule, the degree of importance of a violation, and the consequences of non-compliance with a rule. Such a platform will be able to calculate and trace several indicators such as the correct execution of contracts, indicators of trust in both the results and the partners, and the relevance of a contract's rule in the actual interaction. Moreover, thresholds can be defined to alert human operators of improper performance to take corrective actions semi-automatically. It is important to highlight that such an approach is not limited to web services or fully automated information systems, but it can also be applied to an Electronic Document Management, where those mechanisms may relieve operators of some manual validations.

In this article, we present the first results of our controllability method aiming the creation of machine-readable service contracts, which extend the expressiveness of traditional Service Level Agreements (SLAs) with more complex business requirements about the expected behavior of partners regarding to the use of assets. Concretely, we present our proposed formalization of the semantic of service contracts. This article is structured as follows. Our problem of controllability, which justifies the need of having a clear semantic of contracts, is described in detail in Section 2. In Section 3 the state of the art is presented which illustrates the limitations of existing approaches. In Section 4 our followed methodology for the construction of the proposed semantic model is presented. Later, the core of this article, an ontology to formalize the semantics of service contracts is presented. Finally, Section 6 presents the conclusions and future works.

2. Controllability

To trust external partners is needed to the opening of an information system towards the outside world, as when information is collected, stored, or processed by an external organization. In scenarios where the information is framed in collaborative relationships, to measure the probability of failure as well as to have guarantees about the quality of the provided service become relevant. One the one hand, Service Level Agreements (SLAs) are a common strategy implemented within organizations as part of their security plan to specify requirements about the service provision. Thus, the external partner guarantees some performance levels as well as some properties to secure the data exchanged between the client and the provider such as the authorization to access data, confidentiality and data availability. However, such technology-centered guarantees do not consider specific behaviors of the human interaction, which are equally important to protect an organization against business damages such as the loss of clients, loss of reputation or fines due to the non accomplishment of some legal normative. On the other hand, regarding the risk analysis, the standard ISO / IEC 27005 (ISO/IEC, 2005), which is associated to the information security, allows to quantify a risk against a potential threat. However, in the development of this standard the notion of service was not taken into account. Similarly, besides the technical aspect of the provision of a service, the

economical, legal, and business aspects must be considered. Specifically, as for usage policies, from the moment an organizational resource leaves the organization's perimeter, there is no way of knowing if this resource is being used as expected. The consequences of such a loss of control justify the need for methods to control the use of shared resources in the provision of a service. The challenge in this scenario is to ensure that the external partner behaves as expected when the resource is in its domain. This idea already existed in large projects involving several partners, but they aim at the coordination of interfaces between organizations, or about the creation of cross-organizational access policies. On the contrary, our approach aims to reuse the business processes already developed in project management and to (semi-) automate them within a platform by integrating this notion of control in a framework of risk management and information security.

In (Lalanne, 2013) controllability (it was the proposed translation by authors in (Lalanne et al., 2013) from the original French term *maîtrisabilité*) is proposed as a new abstract security property at the same level than confidentiality, integrity and availability. Author defines controllability as the ability of "ensuring total control over the services used" by allowing the information system architect to qualify the level of trust on services. In that proposition, however, the actual controllability implementation is restricted to data, and more concretely to the content of the documents used on the service provision. Aiming that control, authors propose the use of metadata attached to the documents (creating hence self-protected documents following the Digital Rights Management scheme) in order to make the traceability of the communication. In its simple form, the principle behind that proposition is to create policies embedded in the documents exchanged among the service stakeholders. Those rules govern the access to documents according contextual information, while some kind of metadata is collected in order to calculate indicators useful to supervise the information system.

In our work, we agree with the above definition of controllability, as well as its underlying motivation in terms of organizational risks. However, unlike existing works (Lalanne et al., 2013) (Munier et al., 2012) (Munier et al., 2014a) (Munier et al., 2014b) we are not restricted to data, but we consider more general organizational resources, referred to as assets. In our approach it is not only important the result of the service provision, but the way how this result was obtained in terms of the use of assets (behaviors of the service stakeholders) in order to avoid organizational damages. Our controllability approach aims the enforcement of the service provision following some business rules consistent with the risk management process of each organization. Moreover, we improve current works by the formalization, and its implementation in a machine-readable form, of the policies governing the relation between the service stakeholders. This aspect has been left as future work so far.

Basically, the problem of the lack of controllability in the service based technologies can be described as follows. During an external service provision some organizational assets are exchanged from/to both sides of the interaction (clients and providers). When the asset is in the domain of his "owner", policy enforcement mechanisms can be implemented to guarantee a limited set of privileges over its usage. However, when it leaves such a domain, the "owner" has no control over the behaviors of the external partner. The consequences of such loss of control are not

trivial since the way in which the shared resources are used by the external partner may intentionally or unintentionally affects the organization causing monetary fines, loss of customers or lawsuits. Our targeted problem of the control over the use of assets is characterized by:

- Organizational resources exchanged during the service provision (which depends on or are affected by the behavior of the external partner).

- An external partner whose behavior cannot be directly inspected.

- A set of requirements describing the expected behavior of the external partner.

Current controllability approaches regulate the behaviors of partners by the implementation of plain-text agreements. Those documents have legal value (they bind parties to its accomplishment), and so can be used in case of litigation. As it was previously mentioned, to the best of our knowledge current machine-readable SLA only cover a subset of such an agreements, concretely, monetary, security and runtime attributes (availability, response time and execution time). This makes each organization has a restricted view of the overall behavior of its partners, and useful business information is left aside to automating tasks associated to the decision-making process, for instance, decision about whether to change of partner.

Our work aims the creation of service contracts able to represent and verify business requirements aiming the control over the use of assets, where such an usage reflects the expected behavior of the service stakeholders. Our analysis of real cases led us to the following conclusions:

- Since the behaviors of partners cannot be directly observed (inspected), some mechanism should be putted in place to verify the compliance with the contract.

- In regards to business commitments, it should be considered several degrees of non accomplishments, from a complete lack of compliance with some contractual term to a compliance with some deviations of the initially agreed conditions.

- Organizational requirements about the use of assets are expressed as coarse-grained business activities.

The aforementioned issues highlight the importance and justify the need of having an ontology which agrees in the vocabulary used for expressing the expected behaviors of partners. Indeed, due to the fact that each organization creates policies which need to be accomplished by other external organization, an agreement in the vocabulary used is needed to avoid misunderstandings in the expected commitments. This article focuses in the formalization of the semantic of a service contract for the SBT. Following, we present the state of the art clarifying the limitations of the two closer domains of our approach, namely, machine-readable service level agreements and semantic contracts.

3. State of the Art

In order to formalize non functional business requirements governing the service provision, a review of the existing works was done to determine their strengths and limitations. In the following, we present the most relevant approaches used as a

basis for the development of our semantic contract. Those works are grouped into two categories: machine-readable SLAs and ontologies for contracts.

3.1. Machine-readable SLA

A Service Level Agreement (SLA) is a document containing the terms governing the service provision. In general, such a document can be written in natural language or machine-readable language; following, we focus in machine-readable ones since we are interested in determining the expressiveness of current formalizations (models).

SLA is an active area of research in SBT since the guarantees they contain serve as a basis to evaluate the quality of the service and to calculate metrics of reputations and trust. WSLA (Ludwig et al., 2003) and WS-Agreement (Andrieux et al., 2005) are the most known models proposed for defining SLA. They are both based on the specification of templates from XML schemas and they are able to express non-functional aspects associated to penalties, rewards, payments as well as to model the actions to take in case of violations of SLOs. SLA* (Kearney et al., 2010), unlike the two previous approaches, is not tied to any language. It supports general definition of services (not only web services) through the proposition of a domain and language-independent definition of the agreement. In this approach, instead of defining constraints as a relation between two elements (parameter-value), it is defined as a variable bound by a domain, which gives a more abstract expressiveness. Due to the fact that the approach focuses on the syntax, authors of this work present as main limitation its lack of semantic, in particular for the definition of action post-conditions. WSLA+ (Nepal et al., 2008) and SLAng (Lamanna et al., 2003) focus on SLA for inter-organizational environments. The former supports agreements signed by multiple parties while the latter models SLA by splitting the target service provision model into three tiers: applications, middleware, and the underlying resources (network, storage) where each component of the tier is provided by a different organization.

WSOL (Tosic et al., 2002)(Tosic et al., 2005) is proposed to manage single and composed web services. In this approach, the service offering plays the role of a SLA or service contract (both terms are used interchangeably in that work) consisting on a formal representation of a single class of service, together with its constraints and management statements. However, as most of the SLA approaches, it focuses on the syntax, while the semantic is left for defining a vocabulary of metrics and measurement units of the QoS.

Current approaches to represent requirements about the service provision can be classified in functional and non-functional. Due to the fact, we are interested in representing in the contract the expected use of the shared resources, the non-functional requirements were analyzed in depth. Such analysis led us to conclude that current non-functional requirements focuses on technical and runtime features of the service, mainly targeting security and performance properties. Moreover, most of them focuses on the syntax and not the semantics, which is an important aspect for understanding the contractual commitments and their negotiation. As stated in

(Karaenke, Kirn, 2007), current SLAs do not sufficiently fulfill the requirements on business "since they are though with other technical objectives" which is implicitly confirmed in (Paliulioniene, 2013). Moreover, the adaptation of current approaches to support controllability is not suitable since they aim a runtime monitoring to prove compliance with the agreement terms. On the contraty, business requirements are not always observable or measurable. Conversely, for the definition of controllability, we need a machine-readable representation which holds coarse-grained activities as well as any resource over which the organization wants to keep control. Additionally, a clear semantic for the definition of those elements is needed to tackle the use of business specialized vocabularies.

3.2. Contract Ontologies

The use of ontologies in the domain of contracts or SOA is not new. The SOA-O 2.0, proposed in 2014 by The Open Group (Group, 2014) as a standard ontology for SOA, aims to fill the gap between the business and IT vision of a service-oriented architecture. In SOA-O, service contracts are "agreements needed in order to define how to use a service. [...] A service contract is binding on all participants in the interaction, including the service itself and the element that provides it for the particular interaction in question" (Group, 2014). Although this definition agrees with our vision of service contracts, the functional elements (components) of the SOA domain constitute the core of that standard, then no ontology for service contracts is proposed.

Other more general approaches of contract ontologies are proposed in (Cesare, Geerts, 2012)(Ramanauskaitė et al., 2013)(Yan et al., 2006) and (Kabilan et al., 2003). In (Yan et al., 2006) it is presented a contract ontology mainly based on a taxonomy of contract concepts. This work explicitly states the problems that arise in a contractual relation and classify them in pre-contract problems, contract-phase problems and post-contract problems. Since that approach is not focused in service-oriented architectures, the loss of control in assets is not included in the categorization of contractual problems and no analysis is proposed regarding the business perspective of the service provision as part of the contract terms. Authors of this work highlight the importance of creating ontologies based on the expressiveness capability of OWL instead of the expressiveness of the natural language. In (Cesare, Geerts, 2012), authors state that an ontological model should map as accurately as possible the reality of the world, otherwise the model will represent "concepts conceived as human creations". Based on the philosophical view, this work proposes a perdurantist ontology of contracts, according to which an object is defined by attributes partially present during the object'sexistence. In this approach, a contractual party is completely defined by commitments, states and execution events. Despite the fact that some analysis are made addressing the duality of events, the reciprocity of commitments and the description of the economic resources, the exchanged resources (i.e. the economic ones) are not explicitly modeled as a class or type of element. Moreover, no formal definition of the commitments is presented, nor any concrete representation.

In general, to the best of our knowledge, contract ontologies are mainly used to create taxonomies and an agreement in the vocabulary, instead of annotating data with its semantics meaning. Those ontologies do not cover a controllability vocabulary useful for the definition of contractual policies. In short, no current ontology is enough complete for representing the knowledge of controllability of assets in the frame of service contracts, requiring more precise relations which capture the behavior and the dynamism of a service provision as well as the contractual relations. However, it should not be forgotten that one of the design criteria of an ontology is extendability. In particular, from the above described works, it is noted that our model could be integrated to the SOA-O ontology, so improving its expressiveness by adding new concepts which represents a clear understanding of the components of a service contract.

4. Methodology

In our approach, contracts explicitly represent policies governing the relation between clients and providers regarding the use of assets. Ontologies are not used here with the purpose of creating a taxonomy, but a vocabulary for adding a clear semantics to the contractual terms that will be used for describing the policy. For instance, if the term “contractual parties” is used in the policy, its semantics will allow to refers to “the client” and “the provider”. It also has the advantage of avoiding the subjectivity in the interpretation of the policy by supporting the semantics of its vocabulary in a formal model.

We consider therefore service contracts as the specific domain of knowledge to be represented. Following, a model of the knowledge belonging to such domain is presented. In order to determine what specific formalism should be used, we firstly identified our needs of expressiveness according to the following methodology (Jaramillo et al., 2015):

1. Contract base collection. It has been commonly stated in the literature that the creation of a semantic model must be led by the needs of representation in a specific domain of knowledge rather than the expressiveness of the description language. As a first step to identify those needs, a contract base composed of 44 real documents was built and grouped into two categories. The first set corresponds to real documents written in plain English containing business terms about the commitments of each party about services provided in the cloud. Those contracts were analyzed each one belonging to a particular layer of the cloud stack, i.e. data/document storage, infrastructure, and e-commerce applications. The second set of contracts corresponds to more general service contracts, i.e., services which are not provided by the cloud nor mediated by a web service. This second group allows us to identify business policies associated to more general organizational assets. In general, the document collection aims the creation of a model which effectively supports the representation of business needs.

2. Contract tagging. In this step, each contract was manually analyzed and key elements according to a controlled English were identified and tagged. The aim of this step is twofold. Firstly, it allows to define the structure of service contracts.

Secondly, it creates a conceptual definition of the elements represented in the contract. In the tagging process, a contract vocabulary consisting of concepts and definitions is built. More in detail, each statement of the contract was individually analyzed in order to identify contract concepts, specific names (individuals), verbs expressing relations between contract concepts (roles), and basic language symbols such as quantifiers, connectors, modals, and quantifications. To illustrate this step, let us take the following excerpt of the Dropbox's term of service:

> *"Some Services allow Customer to download Dropbox software which may update automatically. Customer may use the software only to access the Services. If any component of the software is offered under an open source license, Dropbox will make the license available to Customer."*

As a result of the tagging process concepts such as *Service, Client, Provider*, *Asset, Activity*, and *Attribute* are identified, to which specific individuals are associated. For instance, in the contract instantiation, Dropbox is semantically defined as an individual of the class *Provider*. Moreover, relations are created to enrich the semantic meaning of contractual concepts. For example, the relation *identifies* allows to describe the fact that the individual *Services* does not represent the service itself but it is a term used through the document to refer to the set of services covered by the terms of service.

3. Formal representation. Once the contractual terms are identified, their semantics is defined by using the Description Language (DL) formalism. In particular, our needs of expressiveness were matched with the constructors proposed by the different subsets of DL. The DL formalism is the core of the definition of ontologies. It allows to add a formal semantic to the specification of concepts while offering mechanisms of inference. It aims decidability due to an expressiveness richer than the propositional logic but more restrictive than the First Order Logic (FOL). Currently, this formalism is composed of several sub-languages with different degree of expressiveness. By following this methodology, the `SROIQ(D)` formalism was used, since it fulfills our needs of representation while keeping decidability.

With those three steps, it is possible to create formal definitions of both the contractual concepts and the terms associated to the control of assets. However, a final step is added to the methodology in order to validate the expressiveness of the chosen logic representation as well as to validate that business requirements written in plain English can be mapped into a machine-readable representation.

4. Machine-readable representation. In this step the abstract model represented by the DL formalism is translated into a concrete ontological representation in XML by using languages and engines supporting the logic representation and inference. In this step, the Protégé tool was used to validate the concrete XML representation of the OWL 2 against the Hermit 1.3.8 reasoner.

Despite the high expressiveness of the `SROIQ(D)` formalism, we strive to make clear that our proposition of machine-readable service contract has been developed under the basis of two restrictions:

– *We assume that contracts are signed only between two parties, one acting as a provider and the other one acting as client*. Consequently, federate contracts (contracts agreed among all the partners involved in a workflow) are not considered in our approach. Instead, we break down the service provision chain in several client-provider relations, and the controllability terms including third parties' obligations are tackled by means of propagation of policies. As a consequence, in case of orchestrated services, several contracts govern the interaction of partners at the same time.

– *The negotiation process is out of the scope of this work*. So, our proposition assumes that the policy governing the service provision has been already agreed between the client and provider.

5. Semantic Contract Model

Following, formal descriptions of the elements (concepts) identified in each of these components is proposed. For each element, a set of axioms is defined for establishing its relations with other elements of the model. However, for the sake of clarity each axiom is presented and analyzed individually. Therefore, the complete definition of the concept is understood as the conjunction of those individuals axioms.

5.1. Service Actors

In order to binding each service stakeholder through contractual obligations, it is needed to identify and classify the actors involved in the service. In our approach, we divide actors in contractual actors (service client, service provider, third parties, and signatory parties) and controllability actors (controller, processor). The former assign roles which make sense in the framework of the service provision, while the latter, assign roles from the controllability perspective.

5.1.1. Contractual Actors

Service provider and Service client are defined in the same way as used in the service oriented architectures. That is to say, the provider is seen as an active actor supplying a service, which is requested by the client. In the modeling of the semantics of a service contract, it was assumed that the contract governs one single service. It implies, the definition of at least two contractual parties, where their role in the contract is defined by means of the property involvesParty.

Third parties include certification authorities, subcontractors of the provider, auditors and suppliers. From the legislation point of view, most of the law of contract around the world state that only contractual parties, i.e. service clients and providers, can enforce a contract. However, specific ordinances, such as those in Hong Kong and UK, give some benefits and rights to third parties. As prescribed by law, in case that a contractual party confers benefits or rights to a third party, this latter should be specifically identified in the contract by its name or as a member of

a class or as answering to a particular description. Consequently, third parties are identified in the tagging process as concepts whose semantics need to be defined in terms of their relation with the service, and more importantly, in terms of their relation with assets.

A signatory party is an entity legally acting on behalf of the client/provider for attesting that the party agrees with the contract. So, even if the client and provider are both the bound parties, who actually signs a contract is not the organization as a whole but an individual actor with the authority for representing the organization.

The modeling of the contractual actors in the DL formalism is formalized as:

$$\begin{gathered} Contract := \{c \mid \forall (c, sp) \in isSignedBy \rightarrow sp \in SignatoryParty\}, \\ \{c \mid \exists (c, ct) \in isComposedOf \wedge ct \in ContractualTerm\}, \\ \{c \mid \forall (c, s) \in governs \rightarrow s \in Service\}, \\ \{c \mid \forall (c, sg) \in isAuthenticatedBy \rightarrow sg \in Signature\}, \\ \{c \mid \forall (c, a) \in involvesParty \rightarrow a \in ContractualActor\}, \\ \{c \mid \#\{p \mid (c, p) \in involvesParty \wedge p \in Provider\} = 1\}, \\ \{c \mid \#\{cl \mid (c, cl) \in involvesParty \wedge cl \in Client\} = 1\} \end{gathered} \tag{1}$$

$$(Provider \cap Client) \cap ThirdParty = \emptyset \tag{2}$$

$$ThirdParty := (ContractualActor \wedge \neg Client \wedge \neg Provider \wedge \neg SignatoryParty) \tag{3}$$

$$Client := \{cl \mid \forall (cl,s) \in requests \rightarrow s \in Service\} \tag{4}$$

$$Provider := \{p \mid \forall (p,s) \in provides \rightarrow s \in Service\} \tag{5}$$

5.1.2. Controllability Actors

In order to represent the relation between assets and their actors, the concepts controller and processor are modeled. Those terms represent the fact that even if assets are used by different actors, they "belong" to one entity, who grants some rights over them to an external actor for a particular purpose. In (European Parliament and the Council of the European Union, 1995), a controller is defined as any "natural or legal person, public authority, agency or other body which, alone or jointly with others, determines the purposes and means of the processing of personal data". Similarly, a "processor means a natural or legal person, public authority, agency or other body which processes personal data on behalf of the controller". In

our model, those definitions are retaken but also extended to tackle a more general set of organizational assets than only data.

We define a controller as an actor establishing business rules about the use of the assets which are under the control of his organization. Similarly, a processor is an actor enforced to respect the business rules established by the controller regarding the use of assets. By signing the contract, the processor is responsible for complying with the contractual rules, i.e. with the use of the assets according to the controller expectations and requirements.

$$Controller := \{ct \mid \forall (ct, as) \in controls \rightarrow as \in Asset\} \tag{6}$$

$$Processor := \{pr \mid \forall (pr, as) \in uses \rightarrow as \in Asset\} \tag{7}$$

$$Client \supseteq (Processor \cup Controller) \tag{8}$$

$$Provider \supseteq (Processor \cup Controller) \tag{9}$$

$$controls \cap uses = \emptyset \tag{10}$$

From the previous formalization, some knowledge such as {Processor employs some Asset}, which is not explicitly asserted in the knowledge base can be inferred by the reasoner because the Processor is inferred to be an actor. It leads to define a subsumption axiom between uses and employs as shown in Eq. 11. The difference between these two relations is that while the latter is a wider concept defining the dependence between two different concepts defined in the contract, the former is a more specific relation which defines the expected use of an external partner with an organizational asset.

$$\forall (x,y) \in uses \rightarrow (x,y) \in employs \tag{11}$$

Regarding the definition of the controllability rules, the concepts Controller and Processor are used to formally represent the relations establishing who sets the rule and who is the beneficiary of the rule defined in the contractual term.

$$Rule \supseteq \{r \mid \forall (r,a) \in hasBeneficiary \rightarrow a \in ContractualActor\}, \{r \mid \forall (r,cp) \in isSetBy \rightarrow cp \in ControllabilityActor\} \tag{12}$$

$$ControllabilityActor \supseteq Controller \tag{13}$$

$$ControllabilityActor \supseteq Processor \tag{14}$$

5.2. Service

Unlike most of the existing SLA where the service is defined as a technological interface, our approach links the semantics of services to a business perspective. Indeed, due to the fact we aim the formalization of business requirements in a

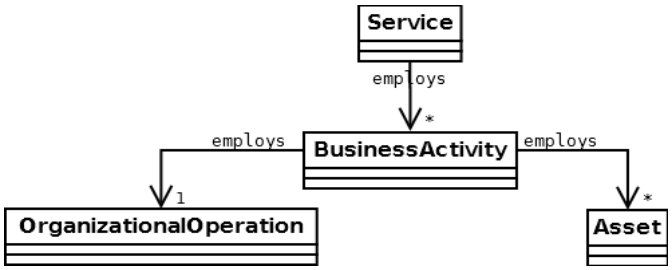
machine-readable form, services are considered in its broad sense, as any process offered by a service provider to a service client, by keeping the properties of transparency, flexibility, and business and technological independence between clients and providers. Thus, the result of the contractual relation between clients and providers is some tangible or intangible commodity, which represents the core of the business logic of the provider, as well as the final "product" of the contractual relation. Therefore, the definition of both service and commodity are highly important since the conformity of the commodity with the client's expectation is part of the successful assessment of the quality of the service. The definition of the service, in the context of a contractual relation, is formalized as:

$$Service \supseteq \{ s \mid \forall (s,com) \in produces \rightarrow com \in Commodity \} \tag{15}$$

5.3. Business Activity

As it has been previously stated, in the contractual relations between clients and providers, organizations rarely reflect requirements in terms of fine-grained actions, instead of that, more coarse-grained activities are described. Precisely, it is due to the lack of solutions allowing to represent and verify those requirements, that they have not been included in the machine-readable SLAs. In order to represent business terms in the controllability requirements, the concept Business Activity is defined. A business activity represents a coarse-grained organizational operation in which the use of one or more assets is involved.

Figure 1. Business Activity



Therefore, a business activity represents the observable use of an asset within an organization. In other words, the operation represents any possible interaction with the asset, while the activity is the organizational term coined for that use. For instance, a notification is a business activity where the contact information represents the asset, and the act of sending is the operation. Note that in our model, a controllability actor (controller or processor) is linked to the asset, instead of the activity. Indeed, we are interested in modeling the fact that for the correct provision of the service a set of activities need to be carried out either by the provider or the client, and for doing so, some assets are used. Business activities can be therefore seen as a concrete representation of the partners' behavior. Individuals belonging to the class BusinessActivity will be used in the definition of the policies governing the contractual relation in order to restrict the specific situations/conditions in which an actor can perform the activity.

$$Service \subseteq \{ s \mid \exists (s,bo) \in employs \wedge\ bo \in BusinessActivity \} \tag{16}$$

$$BusinessActivity \subseteq \{ ba \mid \exists (ba,op) \in employs \wedge op \in OrganizationalOperation \} \tag{17}$$

$$BusinessActivity \subseteq \{ ba \mid \exists (ba,a) \in employs \wedge\ a \in Asset \} \tag{18}$$

5.4. Asset

Taking in mind that the goal of our controllability policies is to explicitly define rules governing the use of organizational resources, the definition of what an asset means and the way in which it relates with the components of the service contract becomes a key aspect of our model. In the contract model, an asset is any resource having some value for an organization and on which some rules regarding its use are established as part of the contractual terms.

According to the ISO/IEC 27005 (ISO/IEC, 2005), organizational assets are divided into two groups, namely, primary assets and supporting assets. In general, the business operations, processes, activities and information sensitive for the organization are considered as primary assets. It means, resources which could compromise the mission of the organization. Whilst to the second group belongs those organizational elements that may compromise the primary assets if their vulnerabilities are exploited. This includes the hardware, software, network, personnel, site and the organization's structure. Our approach relies on the fact that a subset of those assets are affected by the behaviors of the external partner, it comprise physical resources as well as intangible ones such as the reputation.

Concretely, assets are modeled in terms of their relations with the actors during the service provision. Those relations specify the usage in terms of any possible interaction of the actor with the asset. The relations isControlledBy and isUsedBy relate the actor who restricts the activities that are done with the asset, and the external actor who according to the contractual terms is bound to comply with those usage restrictions. Taking in mind this definition of assets, it is clear that our aimed policies cover a wide range of organizational resources than only files or data, as the traditional security policies.

$$\begin{aligned} Asset := & \{ as \mid \forall (as,ct) \in isControlledBy \rightarrow ct \in Controller \}, \\ & \{ as \mid \forall (as,pr) \in isUsedBy \rightarrow pr \in Processor \} \end{aligned} \tag{19}$$

5.5. Attributes

One of the limitations of current approaches is the difficulty of representing complex restrictions about the service provision. Indeed, traditional service guarantees are expressed in the form of parameter-value relations. Such a representation is consistent with a guarantee enforcement based on the collection of measurements which represent runtime attributes. However, it does not allow to

formalize complex guarantees such as restrictions in terms of other guarantees. In our semantic contract, we propose to model the concepts Relational Attributes and Quality Attributes.

Attributes are highly important because they are an essential part of the definition of the contract concepts by describing the properties associated to them. In the FOL, attributes can be represented as *hasAttribute(contractConcept, attributeName, attributeValue)*; however, the theoretical model of the description logic assumes $R = \Delta^I \times \Delta^I$, it means a role is assigned to a pair of individuals in the form of 2-ary operations. Therefore, we reformulate the previous 3-ary relation as:

$$hasValue\left(a, \langle attributeValue \rangle\right) \tag{20}$$

where {a ϵ hasAttribute(<concept>, <attributeName>)}, {concept ϵ Δ^I} and {attributeName ϵ Attribute}. Consequently,

$$\left[\exists e \mid \forall\ att\left(e,att\right) \in hasAttribute \rightarrow att \in Attribute\right] \tag{21}$$

We consider two kinds of attributes according to the knowledge represented by the attributeName (which, in turn, determines the value of the *attributeValue*), namely: quality attributes and relational attributes. The former assigns a data-typed value to an attribute, while the latter defines the value of an attribute as being an individual belonging to the domain of interpretation. Relational attributes are particularly interesting since they overcome the limitation of most of the existing approaches which only represent attributes as a parameter-data value relation.

$$Attribute \subseteq \left[RelationalAttribute \cup QualityAttribute\right] \tag{22}$$

Although according to the DL vocabulary, properties are, in general, expressed as roles, it is not a misrepresentation in our model to describe them as individuals belonging to a class. From the point of view of the contractual policies, attributes are not only metadata associated to a concept, but as it was previously stated some business restrictions can also be expressed in terms of those attributes. For example, a rule about the state of some other rule or about some physical feature of a commodity. We argue that if the behavior of the processors need to be controlled regarding those attributes, then they become organizational assets themselves. Therefore, this knowledge can be captured by considering the attributes as a concept. It has the advantage of allowing to associate more complex descriptions to the properties themselves, and, facilitates the attribute-based query to the knowledge base.

$$\begin{gathered} hasLiteralValue \subseteq hasValue \\ hasClassValue \subseteq hasValue \end{gathered} \tag{23}$$

$$RelationalAttribute \subseteq \left\{ra \mid \forall\left(ra,val\right) \in hasClassValue \rightarrow val \in \Delta^I\right\} \tag{24}$$

$$\left[\forall\ x\ |\left(x,y\right) \in hasLiteralValue \rightarrow x \in QualityAttribute\right] \tag{25}$$

5.6. Contractual Term

A service contract aims to clarify the terms of the relations between clients and providers by covering any issue that may affect the relation between the contractual parties. Usually, those terms are identified by their nature, mainly, payment, guarantee, remedies, claims, liability, exclusion, contract cancellation/modification and legislation (Figure 2). In particular, guarantee terms are statements in which some contractual party ensures the compliance with some commitments regarding the provision of the service. On the other hand, remedies and claims specify rules which are not directly focused on the service provision but on the path to follow (behavior) when some contractual term, including service guarantees, are not respected.

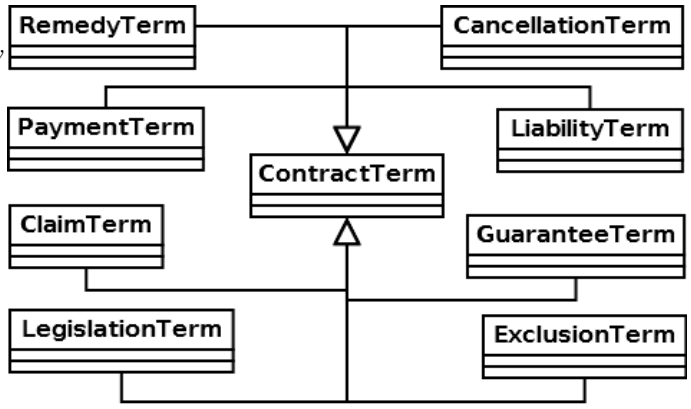


Figure 2. Taxonomy of the Contract Terms

Considering that a contract can be understood as a plan governing the behavior of actors, the events preventing damages to the organization should be explicitly represented in the policy as rules. Consequently, the undesired events which may cause damages are seen as a result of the non-compliance with one or more rules established in the policy, i.e, the non-compliance with some contractual term. In our proposed model, we define any deviation or non-compliance with the contract terms as an infringement. We highlight that an infringement encompasses any non-compliance with a term regardless its nature, it means, it can refer to either the violation of a rule or the ignore of a recommendation. Similarly, note that a damage is an organizational consequence of a breach in the contractual relation, such as loss of reputation, caused by the infringement of a contractual term. In general, damages are usually defined in contracts according to the category they belong, for instance, consequential damage or punitive damage, it allows to hide the actual impact of a contractual breach for the affected organization.

$$Infrigement \subseteq \{ i \mid \exists (i,a) \in affects \wedge a \in Asset \} \tag{26}$$

Similarly, remedies are defined as a compensation for failures to perform the rules governing the service provision or for a deviation of the terms agreed in the

contract. When some infringement occurs, compensations can be putted in place to balance the effect of damages. The explicit inclusion of remedies in the terms agreed between a client and a provider is a current practice in SLAs. However, they are defined in monetary terms, usually such as service credits. From a business perspective, other kind of material or immaterial remedies are also possible such as awards or preferential treatment. Examples are found in airline companies who in case of overbooking offer free hotel rooms, extra travels or some other amenities to passengers that voluntarily cede their seats.

$$Remedy := \{ rem \mid \forall (rem, inf) \in compensates \rightarrow inf \in Infringement \} \tag{27}$$

Figure 4 summarizes the formalization of the semantic contracts.

Given our needs of expressiveness, we use the OWL 2 language which supports semantics based on SROIQ(D). More concretely, we use the OWL/XML syntax, which is an XML serialization whose aim is the interoperability with other XML representations, in particular, those oriented to the web such as WSDL, XPath, XSLT and schema-aware editors. Due to the fact that service contracts are aimed to be attached to the service description, an XML representation addresses the interoperability required in the service oriented architectures. Note that the interoperation between other XML-related technologies and OWL could be leveraged by the use of GRDDL transformations; also, some toolkits exist for the translation of OWL/XML into RDF/XML.

Considering that the way in which the syntax is mapped into the semantics is part of the reasoner implementation, the proposed semantic contract representation was validated against the Hermit 1.3.8 reasoner to guarantee both the decidability in inference tasks and that the represented knowledge is not contradictory. HermiT has been developed to tackle direct semantic, therefore it is fully conformant with the OWL 2 DL. (W3C, 2009) presents the comparison of several OWL 2 engines regarding their performance against some reasoning test cases.

The machine-readable implementation of the semantic contract model was done by mapping the concepts and roles presented in this Section into classes and properties, respectively in OWL 2. The result of such mapping is an ontology described in the SROIQ(D) formalism, composed of 325 axioms and 59 classes. Figure 3 shows the metrics of the contract ontology taken from the Protégé tool.

6. Conclusion

The representation of a specific domain based on ontologies has been widely used for sharing and formalize knowledge in a machine-readable form. In this article, motivated by the need of having an agreement in the vocabulary used to create controllability policies, an ontology of the semantics of service contracts is proposed. The model of the contract is formalized by using a subset of the FOL, specifically, the DL formalism. The OWL 2 language is used as the concrete syntax of the model, which allows its machine-readable representation. The proposed model is the first step towards the implementation of controllability policies in inter-

organizational environments. It also contributes to the current models of both SOA and contract ontology. Despite the contributions of the proposed model some aspects are left as future works, notably the formalization of the negotiation process and the extension of the contract to support multiparty signatures.

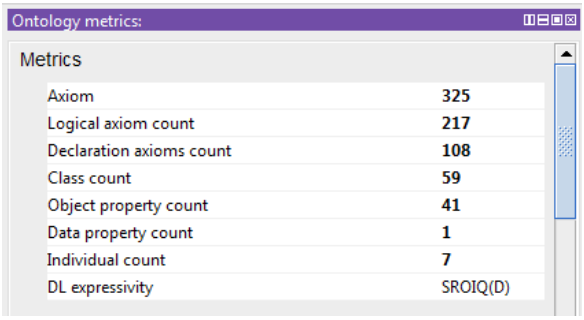


Figure 3. Contract Ontology Metrics in Protégé

In the last years, organizations have understood the need to clarify what each partner can, can not, should or should do in the course of providing services. In this work, our goal was primarily to align the customer's expectations with the actual provided service. This will allow us to evaluate the conformity of the execution with respect to the commitments. It is justified by the fact that in the Business-to-Business context, a "business dependency" is implicitly generated, due to the fact that some assets are shared between customers and suppliers. From a risk management perspective, the way in which assets are used by the external partner can cause organizational damages such as customer loss, fines, loss of reputation, or lawsuits. Although an asset generally refers to any organizational resource, this work has been particularly focused on shared assets because the challenge of this approach is that they move from an organizational domain to other, while ensuring that each organization retains control over its own assets.

We are currently working in the implementation of the proposed semantic contracts (as well as the logs that contain the evidences) within a platform to audit the interaction between clients and providers. Such a platform is able to calculate and trace indicators about the correct execution of contracts, relevance of contract's rules (the frequency of a rule violation gives useful insights about the relevance of the commitment). Those indicators can also be used to propose a model of trust and reputation, which, in terms of risk management is an additional security parameter to consider when choosing a service provider.

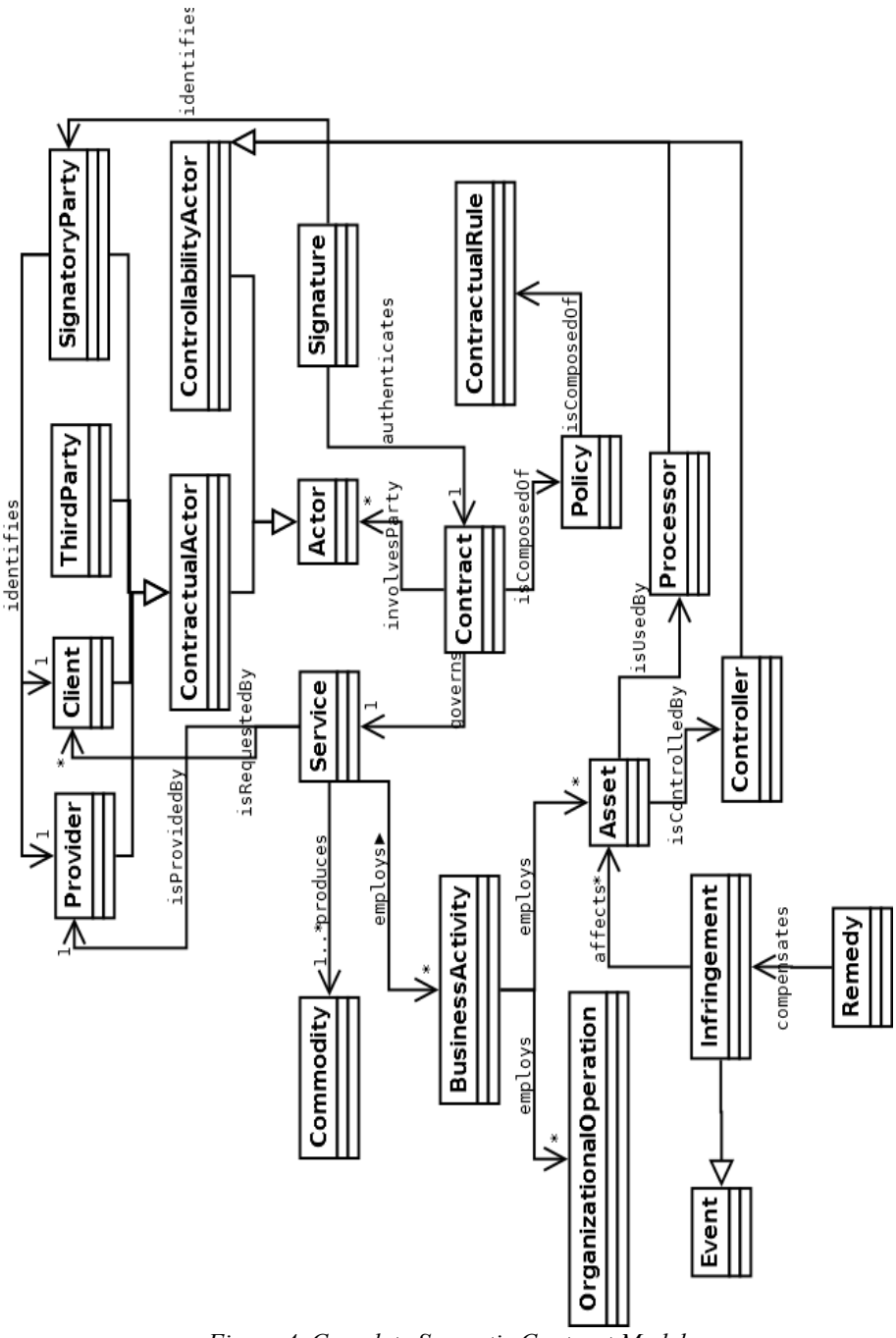


Figure 4. Complete Semantic Contract Model

Bibliography

Andrieux A., Czajkowski K., Dan A., Keahey K., Ludwig H., Nakata T. et al. (2005, septembre). Web Services Agreement Specification (WS-Agreement). Rapport technique. Global Grid Forum, Grid Resource Allocation Agreement Protocol (GRAAP) WG. Consulté sur http://www.ggf.org/Public_Comment_Docs/Documents/Oct-2005/WS-AgreementSpecificationDraft050920.pdf

Cesare S. de, Geerts G. L. (2012). Toward a perdurantist ontology of contracts. In Advanced information systems engineering workshops: Caise 2012 international workshops, gdansk, poland, june 25-26, 2012. proceedings, p. 85–96. Berlin, Heidelberg, Springer Berlin Heidelberg. Consulté sur http://dx.doi.org/10.1007/978-3-642-31069-0_7

European Parliament and the Council of the European Union. (1995). Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. Official Journal of the European Union, vol. L 281, p. 0031–0050.

Group T. O. (2014). Service-oriented architecture ontology, version 2.0. The Open Group Technical Standard no Reference C144. Auteur.

ISO/IEC. (2005). ISO/IEC 27002:2005 - Information technology – Security techniques – Code of practice for information security management. Rapport technique. Auteur. Consulté sur http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=50297

Jaramillo G. E., Ardagna C. A., Anisetti M. (2015, May). A hybrid representation model for service contracts. In 2015 International Conference on Information and Communication Technology Research (ICTRC), p. 246-249.

Kabilan V., Johannesson P., Rugaimukamu D. M. (2003). Business contract obligation monitoring through use of multi tier contract ontology. In On the move to meaningful internet systems 2003: Otm 2003 workshops: Otm confederated international workshops, hciswwa, ipw, jtres,worm, wms, and wrsm 2003, catania, sicily, italy, november 3-7, 2003. proceedings, p. 690–702. Berlin, Heidelberg, Springer Berlin Heidelberg. Consulté sur http://dx.doi.org/10.1007/978-3-540-39962-9_70

Karaenke P., Kirn S. (2007). Service level agreements: An evaluation from a business application perspective. In Proceedings of echallenges.

Kearney K. T., Torelli F., Kotsokalis C. (2010, Oct). SLA*: An abstract syntax for service level agreements. In 2010 11th IEEE/ACM International Conference on Grid Computing, p. 217-224.

Lalanne V. (2013). Gestion des risques dans les architectures orientées services. Thèse de doctorat non publiée, Université de Pau et des Pays de l'Adour.

Lalanne V., Munier M., Gabillon A. (2013, Sept). Information security risk management in a world of services. In Social computing (socialcom), 2013 international conference on, p. 586-593.

Lamanna D. D., Skene J., Emmerich W. (2003, May). SLang: A language for defining service level agreements. In Distributed computing systems, 2003. FTDCS 2003. Proceedings. The 9th IEEE Workshop on Future Trends of , p. 100-106.

Ludwig H., Keller A., Dan A., King R. P., Franck R. (2003, janvier). Web Service Level Agreement (WSLA) Language Specification, v1.0. Consulté sur http://www.research.ibm.com/wsla/WSLASpecV1-20030128.pdf

Munier M., Lalanne V., Ardoy P.-Y., Ricarde M. (2014a). Legal issues about metadata data privacy vs information security. In Data privacy management and autonomous spontaneous security: 8th international workshop, DPM 2013, and 6th International Workshop, SETOP 2013, Egham, UK, september 12-13, 2013, revised selected papers, p. 162–177. Berlin, Heidelberg, Springer Berlin Heidelberg. Consulté sur http://dx.doi.org/10.1007/978-3-642-54568-9_11

Munier M., Lalanne V., Ardoy P.-Y., Ricarde M. (2014b, mai). Métadonnées et Aspects Juridiques: Vie Privée vs Sécurité de l'Information. In 9ème Conférence sur la Sécurité des Architectures Réseaux et des Systèmes d'Information (SARSSI'2014), p. 65-76. Saint-Germain-Au-Mont-d'Or, France. Consulté sur https://hal.archives-ouvertes.fr/hal-01082085

Munier M., Lalanne V., Ricarde M. (2012, June). Self-protecting documents for cloud storage security. In 2012 ieee 11th international conference on trust, security and privacy in computing and communications, p. 1231-1238.

Nepal S., Zic J., Chen S. (2008, July). WSLA+: Web service level agreement language for collaborations. In Services computing, 2008. SCC'08. IEEE International Conference on, vol. 2, p. 485-488.

Paliulioniene L. (2013). On description of contracts and agreements in the context of SOA. Computational Science and Techniques, vol. 1, no 2, p. 171-183.

Ramanauskaite S., Olifer D., Goranin N., Čenys A. (2013). Security ontology for adaptive mapping of security standards. International Journal of Computers, Communications & Control (IJCCC), vol. 8, no 6, p. 813–825.

Tosic V., Pagurek B., Patel K., Esfandiari B., Ma W. (2005). Management applications of the web service offerings language (WSOL). Information Systems, vol. 30, no 7, p. 564 - 586. Consulté sur http://www.sciencedirect.com/science/article/pii/S0306437904001012 (The 15th International Conference on Advanced Information Systems Engineering (CaiSE 2003)The 15th International Conference on Advanced Information Systems Engineering (CAiSE 2003))

Tosic V., Patel K., Pagurek B. (2002). WSOL - web service offerings language. In Web services, e-business, and the semantic web: Caise 2002 International Workshop, WES 2002 Toronto, Canada, May 27–28, 2002 revised papers, p. 57–67. Berlin, Heidelberg, Springer Berlin Heidelberg. Consulté sur http://dx.doi.org/10.1007/3-540-36189-8_5

W3C. (2009). Test Suite Status. https://www.w3.org/2007/OWL/wiki/Test_Suite_Status. ([Online; accessed 13-August-2016])

Yan Y., Zhang J., Yan M. (2006). Ontology modeling for contract: Using OWL to express semantic relations. In 2006 10th IEEE International Enterprise Distributed Object Computing Conference (EDOC'06), p. 409–412.