Chapter 13
Reconciliation of Contractual Concerns of Web Services

Hong-Linh Truong
Vienna University of Technology, Austria

G.R. Gangadharan
IBM Research India, India

Marco Comerio
University of Milano-Bicocca, Italy

Vincenzo D'Andrea
University of Trento, Italy

Flavio De Paoli
University of Milano-Bicocca, Italy

Schahram Dustdar
Vienna University of Technology, Austria

ABSTRACT

There exist many works addressing service contracts fully or partially. They often mention the same notion with different languages and terminologies. This causes several problems in the specification, negotiation, and monitoring of contractual concerns in service-oriented environments, in particular in the Internet-scale and cloud computing environments. With the objective of reconciling contractual concerns, in this chapter, we will analyze the strengths and weaknesses of existing languages and standards for describing service contracts. We will present our research efforts for dealing with multiple contract specifications and semantics mismatching when identifying, specifying, negotiating, and establishing service contracts for service composition in the Internet and cloud computing environments. We will explore the issues of service contracts compatibility and present our solutions. Furthermore, we will analyze crucial points in monitoring and enforcement emerging contractual terms for Internet-based and cloud-based services that so far have not been in the research focus.

DOI: 10.4018/978-1-61350-432-1.ch013

INTRODUCTION

Web services aim at simplifying the interoperability and integration of services, developed by independent providers, for complex business processes by discovering and composing services distributed over the Internet. The building of such complex and valuable processes of multi-provider Web services relies on the efficiency of discovering the services and composing them. The Web services *discovery* will locate machine-processable descriptions of Web services that meet certain criteria. The Web service *composition* consists in combining the functionality of several discovered Web services in order to define composite services (Kuno et al, 2003). The increasing availability of Web services in the Internet and cloud computing environments that offer similar functionalities requires mechanisms to go beyond the pure functional discovery and composition of Web services. As a result, researchers have developed many solutions for enhancing Web service discovery and composition with the evaluation of multiple types of non-functional properties (NFPs) and applicability conditions associated with a Web service. These solutions play a crucial role in the establishment of mutual understanding about the business transaction between a provider (seller) and a consumer (buyer).

Currently, the mutual understanding between the service provider and the service consumer is established by several approaches across various application domains. Contracts, policies, licenses, and service level agreements (SLAs) are the most common approaches for expressing this understanding. Though there are some differences among contracts, policies, SLAs, and licenses, the common denominator is an identification for a belief of a business transaction between the service provider and the service consumer and thereby, commonly considered under the umbrella term '*Service Contracts*' or simply by '*contracts*'.

A contract is a legally binding exchange of promises or agreement between parties that the law will enforce. The contents of contracts may vary as the definition of contract is very broad in scope. A contract can include quality of service terms (e.g., response time and availability), legal terms (e.g., fair use and copyrights), intellectual rights terms (e.g., allowing or denying composition), and business terms (including financial terms such as payment and tax). Typically these terms are described in forms of conditions established on the basis of non-functional parameters. Both the provider and consumer can specify non-functional parameters associated with their services and their requests, in general, but contract terms will include specific conditions of non-functional parameters that are agreed by the provider and consumer for particular business transactions. As there is no single formal ontology or unique way of describing these terms, service providers and service consumers can represent these terms as they wish. This causes strong ambiguity and redundancies in terms descriptions, preventing the right interpretation of contract terms in multi-provider service-oriented environments.

In the literature, there exist many works fully or partially addressing service contracts, but they often mention the same concepts with different languages and terminologies. This requires guidelines and approaches for reconciling them to better collaboration between service providers, service consumers and service integrators. In most cases, current works do not consider the negotiation and establishment of contractual terms modeled by different specifications. However, such negotiation and establishment should be supported as service consumers are increasingly composing and utilizing services provided by different providers, each imposes a different meaning on contractual terms. In parallel, the trend of providing Software-as-a-service (SaaS) (Viega, 2009) and Data-as-a-Service (DaaS) (Dan et al., 2007;Truong et al., 2009) in the cloud demands a strong support for establishment, monitoring and enforcement of diverse contractual terms as in the SaaS/DaaS model typically each software usage is bound to a service contract. Addressing service contracts for composite services is important but this work is under-researched.

This chapter, based on an early discussion in (Truong et al., 2008), summarizes our observations, research outcomes and experiences in and discusses some future directions for dealing with the above-mentioned issues. We analyze the complexity of current contractual concerns, identify issues preventing the establishment of contractual concerns in the service composition, and propose guidelines for reconciling existing approaches to better collaboration between service providers, service consumers, and service integrators. While existing languages and techniques for service contracts are mainly designed for Web services, we will consider ``services'' in a generic view in which services are offered by software applications that can be invoked via software or services are performed by a human via a software interface. With this assumption, we will discuss common and differences in service contracts for generic services built based on Web services technologies.

The chapter is organized as follows: we will discuss an overview of service contracts in the next section. Then we will analyze the complexities of current contractual concerns and the possible issues due to these complexities. We present several research issues in the reconciliation of service contracts. Next, we present some of our research efforts that aim at tackling these research issues. Finally, we conclude the chapter and discuss future work.

SERVICE CONTRACTS: AN OVERVIEW

In the literature, the mutual understanding between providers and consumers is typically established by specifying policies, service level agreements, licenses, and contracts. We overview them in the following.

Policies establish a relationship between involved parties, specifying obligations and authorizations. Obligations specify the set of activities that an object must or must not perform on target objects and authorizations specify the set of activities that an object is permitted or prohibited to perform on target objects (Lupu & Sloman, 1997). Policies provide the means for specifying and modulating the behavior of a feature to align its capabilities and constraints with the requirements of its users (Kamoda et al, 2005).

A Service Level Agreement (SLA) contains technical data relating to the operation of services that implies the objectives with regard to a service consumer (Muller, 1999). An SLA, e.g. described by WSLA (Web Service Level Agreement) or WS-Agreement specifications, is a bilateral statement signed between a service provider and a service consumer, over the agreed terms and conditions of the given service. An SLA describes the minimum performance criteria a provider promises to meet while delivering a service and typically sets out the remedial action and any penalties that take effect if the performance falls below the promised standard. Thus, an SLA specifies the expected operational characteristics of a service in business oriented terms between a provider and a consumer, so that the characteristics can be measured, monitored, and managed (Lewis & Ray, 1999, Sahai et al, 2002). An SLA is often custom-made and negotiated between a consumer and a provider of a service. Templates for SLA specifications may vary with service providers. SLA negotiation can be a complex process as several terms and conditions related to performance criteria may involve in an SLA.

Service licensing includes all transactions between the licensor (e.g., service provider) and the licensee (e.g., service consumer) in which the licensor establishes the rights granted to the licensee when using some specific services for a specific tenure under predefined terms and conditions (Gangadharan et al, 2007). Licensing is perceived as a method for supporting technology transfer and as a mechanism for getting financial benefits. The objectives of a service license are as follows.

- To define the extent to which the service can be used, on the basis that any use outside the terms of the license would constitute an infringement.
- To have a remedy against the service consumer where the circumstances are such that the acts complained of do not constitute an infringement of copyrights.
- To limit the liability of service providers in case of failure of the service.

Optionally, a service license can include SLA terms. Thus, a service license can be broader than an SLA, protecting the rights of service providers and service consumers. The agreement between the service provider and the service consumer is bound to comply with license clauses, but in general, a license itself is a unilateral act of the provider and is not part of the negotiation.

In general, policies, SLAs, and licenses serve as a common denominator for specifying normative aspects of services and making business relationships between providers and consumers. Policies are commonly used for access control, quality of service, or other management tasks. Service licenses reflect the rights of providers to control how the service is distributed. Although policies and licenses are similar in that they govern what a service does, they are not the same. Typically, a SLA is a bilateral statement signed between a service provider and a service consumer, over the agreed terms and conditions of the given service. A SLA is often custom-made and negotiated between a consumer and a provider of a service. In case of a service license, the service provider and the consumer are bound to comply with license clauses, but the license itself is generally not part of the negotiation. As the literature describes several terms in a synonymous manner with less or more differences, we use a common terminology for referring to these diverse normative aspects describing non-functional properties. We describe a *contractual term* as an instance built as a constraint based on non-functional properties characterizing concerns associated with services and their data. We consider the terminology ***contract*** as a broader set including the terms of SLAs specified by WSLA or WSOL, the clauses of licenses specified by ODRL-S and the properties of policies specified in WS-Policy or WSPL (detailed in the later parts of this chapter).

Contractual concerns of services refer to functional/non-functional properties and business/management information of services. A service contract includes normative aspects that are agreed between the service consumer and the service provider who acted in compliance with the contract. A service contract is a complicated artifact surrounded with interdisciplinary research efforts. In general, the making of a service contract consists of the following phases.

- *Specification of (provisional) contractual terms:* Service providers and/or consumers can specify their provisional contractual terms. Contractual terms are fundamental aspects for service contracts. They basically describe the associated non-functional parameters of a service, covering QoS, business, legal, and intellectual rights issues related to the service usage. A provisional contractual term is basically a provisional constraint on a non-functional parameter; for example, the price for a pay-per-use is 5 Euros.
- *Negotiation and establishment of a contract:* During this phase, provisional contractual terms are negotiated and agreed contract terms are established between providers and consumers. In some specific situations, renegotiation of contracts is allowed.
- *Monitoring and enforcement of the contract:* In this phase, an agreed contract is

monitored and enforced. A set of metrics for measuring and evaluating contractual terms are considered by providers and/or consumers.

As there are several concerns associated with a service, the number of non-functional parameters can be very large or very domain-specific as well as the type of non-functional parameters can be diverse. Therefore, the specification and management of these non-functional parameters can be complex, making the specification, negotiation and establishment, monitoring and enforcement of service contracts complex. Furthermore, when we utilize several services from different providers, we will not deal with a single service contract described in a single specification, but multiple contracts represented in multiple specifications. Therefore, the number of non-functional parameters, types of non-functional parameters, the number of specifications to be used, etc., are increased and diverse. As a consequence, dealing with service contracts from multiple service providers will need techniques that are capable of managing diverse contractual terms, mapping them together, evaluating their compatibility as well as monitoring and enforcing these terms in one-to-multiple or multiple-to-multiple service interaction models. In the following, we will discuss the complexity of current contractual concerns.

COMPLEXITY OF CURRENT CONTRACTUAL CONCERNS

With an illustrative scenario inspired by the COMPAS project (The COMPAS Project, 2010) shown in Figure 1, we intend to give a sample of the complexity of contractual terms of today's multi-provider Web services and the requirement for reconciliation of contractual concerns. Assume that each Web service has a provisional contract (represented by $\{C_{XS}\}$) and a contract (represented as C_{XY} diagrammatically in a shaded square) is established by through the contract negotiation in order to make two Web services to be compatible. The provisional contract specifies contractual terms that the service provider would like to have, but the contract is the final agreement between the provider and the consumer, based on the negotiation of provisional contracts. In Figure 1, the process flow is represented by arrowed dotted dash lines and contractual negotiations and establishments are represented by arrowed black lines.

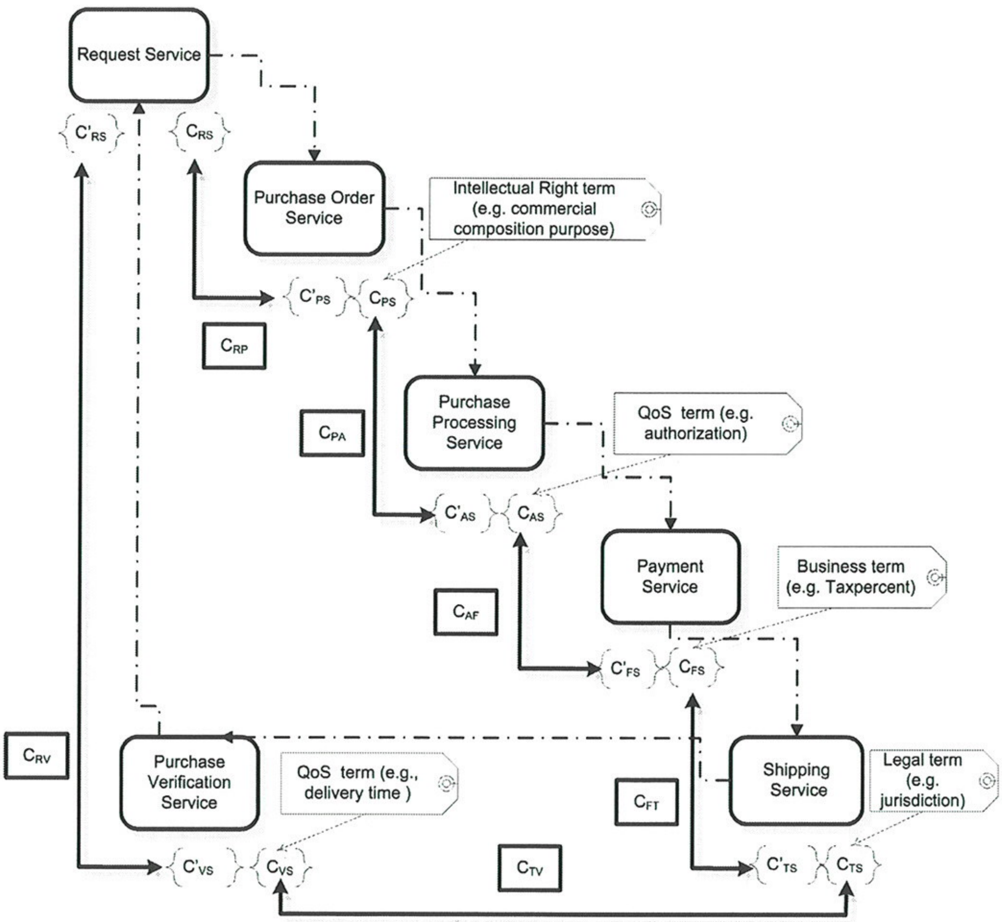
We are considering a supply chain management scenario which involves multiple Web services from different providers. A *Request Service* issuing a purchase request to a *Purchase Order Service*. The *Purchase Order Service* has a provisional contract that includes an intellectual right term represented in $\{C_{PS}\}$. The order sent by this service is being processed by a *Purchase Processing Service* which has a provisional contract $\{C'_{AS}\}$. These contracts are negotiated and agreed upon the terms defined as C_{PA}.

Assume that a security-related QoS term has been associated with the *Purchase Processing Service* which has an impact on the provisional contract in $\{C_{AS}\}$. When the *Purchase Processing Service* interacts with the provisional contract $\{C'_{FS}\}$ of *Payment Service*, a new contract is established as C_{AF}.

A *Shipping Service* processes the *Payment Service* by establishing a contract C_{FT} satisfying contractual terms of $\{C_{FS}\}$ and $\{C'_{TS}\}$. The *Shipping Service* gives certain information about the transferred goods to a *Purchase Verification Service* by mutual contracts establishment. Similarly, the *Purchase Verification Service* sends the verified information about the goods ordered to the *Request Service*, to complete the process of requisition of order to receiving of goods.

In the above-mentioned scenario, there are several service providers. Each service provider has its own services, offers different provisional contracts, and is responsible for the negotiation of the contracts for its own services. On the other

Figure 1. Complexity of contractual concerns in services - a scenario based on COMPAS supply chain management



hand, there is a service integrator/consumer which would like to compose different services into a service composition. While the service providers might not be aware of each other, the integrator/consumer must deal with multiple providers. As in the Internet-scale and cloud computing environments each provider will have its own ways of specifying, negotiating and enforcing service contracts and its own ways will not be the same with that of others, to the service consumer there is a high level of complexity in representing and managing contracts for multi-providers services in the illustrated scenario. First, the current non-uniformity in contractual term specifications raises the question of how to represent contractual terms so that they can interact. For example, in our scenario, the *Payment Service* specifies business term - including financial term - in $\{C_{FS}\}$. As today there is no common ontology for these terms of a contract, the financial term can be represented in the set of quality term by the *Shipping Service*. Second, as these terms can be represented in several XML-based languages, there exist several silos between contracts, SLAs, licenses, and policies. Consider that *Payment Service* represents the provisional contract in the WSLA framework and *Purchase Processing Service* represents the provisional contract in the SLAng. In this case, there are no defined regulations on how these contracts can communicate and establish a new contract. While some works, such as (Keller & Ludwig, 2002), proposed a solution for managing contractual terms across organizations, they mostly assume that all organizations use the same language/specification in describing contractual terms. Here we further examine cases in which organizations use different languages to model the contractual terms.

Issues on Contract Specification Languages and Standards

In the world of Web services, the Web service Description Language (WSDL) (Christensen et al, 2001) and its annotation are the standard way to describe what a service does. From a given WSDL specification and its annotation, the location of the service, the protocols supported by the service, and the functionalities provided in terms of incoming and outgoing messages can be known. Research focusing on languages for service contracts that aims to enhance and complete service description provided by WSDL is continually in progress. These languages/standards are mostly complementary to WSDL by addressing contractual concerns with varying levels of details.

We analyze major existing works, including the Web Service Level Agreement (WSLA) framework (Keller & Ludwig, 2003), WS-Agreement (Andrieux et al, 2007), SLAng (Skene et al, 2004), WSOL (Tosic et al, 2003), WS-Policy (Vedamuthu et al, 2007), Web Services Policy Language (WSPL) (Anderson, 2004), OASIS ebXML Collaboration Protocol Profile (CPP) and Collaboration Protocol Agreement (CPA) (OASIS, 2005), ODRL-S (Gangadharan et al, 2007), OWL-S (Martin et al, 2004) and WSMO (De Bruijn et al, 2005).

The Web Service Level Agreement (WSLA) framework (Keller & Ludwig, 2003) describes the complete life cycle of a Service Level Agreement (SLA) including SLA establishment by negotiation (signing of an SLA by signatory parties for a given service offering), SLA deployment (checking the validity of the SLA and distributing it), Service level measurement and reporting (configuring the run-time system to meet a set of SLAs and comparing measured SLA parameters against the thresholds defined in the SLA), Management actions (determining SLA violations and corrective management actions to be taken), and SLA termination (specifying the conditions for termination). The WSLA framework enables to specify and monitor a wide variety of SLAs for Web services. Based on XML, the WSLA language defines a type system for the various SLA artifacts. An SLA in WSLA is comprised of parties (identifying all the contractual parties), service description (specifying the characteristics of the service and the observable parameters like service availability, throughput, or response time), and obligations (defining various guarantees and constraints to be imposed on SLA parameters).

The Web Services Agreement (WS-Agreement) (Andrieux et al, 2007) supports for establishing agreements between service providers and consumers, using (1) an extensible XML language for specifying the nature of the agreement, and (2) agreement templates to facilitate discovery of compatible agreement parties. The specification consists of three parts which may be used in a composable manner: a schema for specifying an agreement, a schema for specifying an agreement template, and a set of port types and operations for managing agreement life-cycle, including creation, expiration, and monitoring of agreement states.

SLAng (Skene et al, 2004) is a XML based language, for describing Service Level Specifications in the domain of distributed systems and e-business. This language has been modeled by Object Constraints Language (OCL) and Unified Modeling Language (UML) in order to define SLA precisely. SLAng formally defines SLA vocabulary in terms of the behaviour of the services and clients involved in service usage, with reference to a model of service usage.

The Web Service Offering Language (WSOL) (Tosic et al, 2003), a language for specifying constraints, management information, and service offering, provides different service levels defined by several classes of services. The same WSDL description with differing constraints (functional, non-functional, and access right) and managerial statements (price, penalty, and responsibility) is referred as "classes of service" of a Web service in WSOL. Consequently, different classes of services could vary in prices and payment models. WSOL offers several reusability elements to enable easier derivation of a new service offering from the existing offerings.

The WS-Policy (Vedamuthu et al, 2007) provides a general framework to specify and communicate (publish) policies for Web services. It is a model for expressing the capabilities, requirements, and general characteristics of a Web service as policies. The WS-Policy provides a base set of constructs that can be used and extended by other Web services specifications to describe a broad range of service requirements, preferences, and capabilities. The WS-Policy defines a policy as a collection of policy alternatives. In turn, each policy alternative comprises a collection of policy assertions. Each policy assertion indicates an individual requirement, capability or other property of a behavior.

The Web Services Policy Language (WSPL) (Anderson, 2004) is a policy language for expressing various aspects and features related to authentication, authorization, quality of service, quality of protection, reliable messaging, privacy, and application specific service options. WSPL specifies option choices for Web services in basic communication areas (reliable messaging, privacy, security, and so on). A WSPL policy is specified as a sequence of one or more rules, where each rule is an acceptable choice for satisfying the policy. WSPL policies list rules in the order of preference, with the most preferred listed first.

The OASIS ebXML specifies the contractual terms of a party by a Collaboration Protocol Profile (CPP) and the contractual agreement between two parties by a Collaboration Protocol Agreement (CPA) (OASIS, 2005). A CPP defines the capabilities of a party to engage in electronic business with other parties. In other words, a CPP defines what the parties can do (their services and activities) and how they can do. A CPA defines the capabilities that two parties need to agree upon to enable them to engage in electronic business for the purposes of the particular CPA. The CPA can merge the collaboration parties' CPPs into what they have selected to collaborate on and how they intend to do it.

The Open Digital Rights Language for Services (ODRL-S) (Gangadharan et al, 2007), a profile of the ODRL for Service Licensing, is a comprehensive language to represent exclusively the licenses for services, in a machine interpretable form. A license in ODRL-S describes information regarding the service being licensed, the extent to which the service could be used, accessed, and value added, on the basis that any use outside the scope of license would constitute an infringement, payment and charging terms, delivery terms (regarding quality of services and performance), acceptance terms, warranties, and limiting the liability of providers in case of failures, and the rights over future versions and over evolved services.

Current standards for semantic descriptions of services (e.g., OWL-S and WSMO) only marginally cover the specification of service contracts, limiting the support to the specification of sets of NFPs. The Web Ontology Language for Services (OWL-S) (Martin et al, 2004) supports only the specification of NFP in the *Service Profile* and their formalization in OWL. The Web Service Modeling Ontology (WSMO) (De Bruijn et al, 2005) approach allows the specification of a predefined set of NFPs. A closer look at the NFP set shows that many of them are used to express information about the description itself and not about the service. Moreover, these NFPs are not included in the WSMO logical model and thus reasoning on them is not possible.

Several research papers focus on solutions to overcome the current OWL-S and WSMO limitations. (Toma et al., 2007) propose to overcome the WSMO limitation by specifying NFP through axioms. (Kritikos et al, 2006) propose to fill the OWL-S gap by complementing it with a rich and extensible QoS model. (Giallonardo et al, 2007) define a model that allows service providers to advertise on the QoS offered, and service consumers to specify QoS requirements associated with an OWL-S profile.

Table 1 summarizes current support on service contracts with respect to QoS (e.g., performance, security, and dependability terms), business (e.g., financial terms), legal (e.g., jurisdiction place, warranty and limit of liability), and intellectual rights (e.g., share alike, commercial distribution, free). As shown in Table 1, existing approaches have addressed only few aspects (concerns and phases) associated with service contracts. Also, unfortunately, many of them address similar problems but speak in different languages and are not interoperable.

Following the umbrella term *'contractual concerns'*, the terms of a contract can describe quality of service, business, legal and intellectual rights conditions. Without a formal ontology or unique way of describing these terms, services can represent them as they wish, causing redundancies and misinterpretations in term description and leading to several issues in handling these contract terms. To give examples, let us illustrate the complexity in representing contractual terms in the said different languages with respect to our previous scenario.

Based on our classification of contract languages (Comerio, 2009), we group theses languages (see Table 2) and compare the complexity in representation among languages in the respective groups.

In our illustrated scenario, as different providers are participating, these providers could term contractual concerns in different ways.

When *Purchase Verification Service* specifies these terms, it introduces problems in negotiation and establishment of contracts with other services using WSLA or WSOL as these terms are not understandable and not supported by Type B or Type C or other Type A languages.

Though these two services specify similar quality of service conditions, there would be problems in automatic negotiations and composition of these services because these provisional contracts differ in language semantics and syntax.

Table 1. Current languages supporting contractual terms of services

Languages	Specifying			
	Quality of Service Terms	Business Terms	Legal Terms	Intellectual Rights Terms
Web Service Level Agreement (WSLA)	+	+		
WS-Agreement	+	+	+	
Web Service Offerings Language (WSOL)	+	+	+	
SLAng	+	+		
Web Services Policy Language (WSPL)	+			
WS-Policy	+			
Collaboration Protocol Profile and Agreement (CPP/CPA)		+		
Open Digital Rights Language for Services (ODRL-S)	+	+	+	+
Web Ontology Language for Services (OWL-S)	+			
Web Service Modeling Ontology (WSMO)	+			

Table 2. Current languages supporting contractual terms of services

Languages	Type and Description
ODRL-S, CPP/CPA	*Type A*: languages allowing the specification of predefined properties.
WSLA, WS-Agreement, WS-Policy, WSPL, SLAng, WSOL	*Type B*: languages allowing the specification of user-defined properties.
OWL-S, WSMO	*Type C*: languages allowing the specification of properties defined in user ontologies.

During the establishment of a contract, there arises misinterpretations due to non-unified/non-standard approaches in expressing contractual terms of the given two Web services.

Issues on Contract Negotiation and Establishment

In a business environment, a service provider specifies certain contractual terms which describe functional and/or non-functional properties of services. Service consumers also specify their expectation about functional and/or non-functional properties of their requested services. In a win-win business transaction, a provider and a consumer should agree upon contractual terms by negotiation and thereby establishing a contract.

Business and QoS are negotiable terms of a contract. Generally, consumers negotiate on these aspects with providers, using pre-defined templates. Negotiation refers to the comprising exchanges of offers and requests between the participating actors. The theoretical bases of SLA negotiation are provided in (Demirkan et al, 2005), identifying the requirements of a negotiation support system. A common understanding for the contractual terms among the negotiating parties is a critical issue in negotiation. To resolve this semantics issue in (Michael, 2002), template based negotiation is suggested (Grosof et al, 2002). The Policy-driven Automated Negotiations Decision-making Approach (PANDA) (Gimpel et al, 2003) automates the process of decision making within negotiation.

In our illustrated supply scenario, there are certain terms (such as legal terms like "non-commercial use") which cannot be measured by any means. However, these terms are significant in establishing a contract in the Internet and cloud environments. The definition of techniques to support negotiation between providers and consumers on legal and intellectual right terms is an open research challenge.

Exhibit 1. (contractual terms for intellectual rights and legal issues): Assume that Purchase Verification Service allows composition and offers indemnification rights. This can be expressed in ODRL-S as follows in the provisional contract of Purchase Verification Service.

```
<o-ex:permission>
       <sl:composition/>
</o-ex:permission>
<o-ex:requirement>
        <sl:indemnity>
           <sl:thirdpartyinfringementsclaims/>
        </sl:indemnity>
</o-ex:requirement>
```

Exhibit 2. (contractual terms for QoS): Assume that the provisional contract of Payment Service is specified in SLAng as follows

```
<Performance response_time="3"
successful_transactions="99"/>
Consider that Purchase Processing Service specifies QoS in WSOL as follows.
<wsol:QoSconstraint name=''MaxResponseTime''>
      <wsol:QoSname qname=''QoSns:responsetime''/>
      <wsol:QoStype typeName=''QoSns:max''>
      <wsol:qValue> 3 </wsol:qValue>
</wsol:QoSconstraint>
<o-ex:requirement>
          <sl:performance>
              <sl:responsetime> 3 </sl:responsetime>
          </sl:performance>
          <sl:reliability>
              <sl:availabilityrate> 99 </sl:availabilityrate>
          </sl:reliability>
</o-ex:requirement>
```

Issues on Contract Compatibility Evaluation and Composition

In the current service composition landscape, it is not so difficult for service consumers to compose different services based on published service interfaces. Existing composition tools allow service consumer to combine different services, potentially characterized by different service contracts. However, in order to define legal processes, there is a need to ensure that the service compositions do not include conflicting service contracts. This assurance cannot be given by a single provider and currently is not available in existing composition tools.

Incompatibilities among QoS, business, legal and intellectual rights terms specified in contracts of services involved in the composition have a strong impact on composite services and their execution. As an example, services offering incompatible copyrighted data and data distribution or in service coverage can cause inefficient composite service execution since their data are protected by different rules and their coverage is limited to different world regions. Besides the evaluation of compatibility among service contracts, a unified contract for the composite service must be defined composing the contracts offered by the services involved in the composition. This unified contract specifies a recommended set of properties offered by the composite service. However, service contract compatibility and composition are complex activities since they must be evaluated according to the structure of the service composition. This is related to not only the control flow (i.e., the sequence in which the services are invoked) but also the data flow (i.e., the exchange of data between services) of the service composition.

While certain works (Zeng et al, 2004) (Aggarwal et al. 2004) address QoS-based compatibility for control flows, currently there is not a good understanding of how to check contract compatibility and composition for data, the input/output of services, whose contract terms are not always the same to that of the services. The consideration of both control and data flows is essential to perform an efficient Web service contract compatibility evaluation and composition.

Let us consider our scenario described in Figure 1 again, focusing on the interactions between the *Purchase Processing Service*, the *Payment Service* and the *Shipping Service*. The services follow a sequential execution and data are exchanged between the *Purchase Processing Service* and the *Payment Service* and between the *Purchase Processing Service* and the *Shipping Service*. As shown in Figure 1, C_{AS}, C_{FS}, and C_{TS} are the provisional contracts associated with the *Purchase Processing Service*, the *Payment Service* and the *Shipping Service*. To evaluate the compatibility of the availability time range term (i.e., the time range in which the service is available) in C_{AS}, the same term in only C_{FS} must be considered since the *Purchase Processing Service* and the *Payment Service* are executed one after the other. Vice versa, to evaluate the compatibility of the data ownership term (i.e., a license term stating how the data produced by the service are protected) in C_{AS}, the same term in C_{FS} and C_{TS} must be considered since the *Purchase Processing Service* data are managed by both the *Payment Service* and the *Shipping Service*.

Furthermore, past research has not focused on tools and algorithms dealing with contract compatibility evaluation when combining different services from different providers. Typically, they deal with only contract negotiation between consumer and service in a point-to-point manner. As a consequence, the definition of tools and algorithms dealing with service contract compatibility and composition evaluation considering data and control flows of the service composition appear to be an innovative research challenge.

Issues on Contract Monitoring and Enforcement

Consider a scenario where a service consumer is required to pay for a use of particular service through credit card. A message would be delivered to the service provider about payment from the customer. However, unless the credit card consortium informs about transaction, the provider cannot know whether it is paid and the amount of money paid for the use of a service by a customer. Business terms, being one of the important contractual terms, can be measured through an approach involving a third party who informs the provider on the moment of payment by a customer.

In general, many QoS terms are measurable. QoS terms can be measured at the consumer side to confirm whether the received QoS is in conformance with the QoS offered by providers. There can be contradictions in measurement, claiming the differences when a single QoS term is measured at both consumer side and provider side. There should be a consensus in measuring and monitoring of QoS and should be specified in a contract in indemnification clauses. However, there are some contractual terms (e.g., non-commercial use) which are difficult to monitor, if not impossible.

Most of the works in the area of service contracts focus on specifying and establishing contracts. The area of monitoring and enforcement of service contracts is given less attention

Table 3. Monitoring techniques for contractual terms

Contractual Terms	Methods of Measurement	Active Party
QoS terms	Sampling at regular intervals of time, querying through direct communication between provider and consumer	Consumer, Provider
Business terms	Notification through trusted third party approved services	Provider
Legal terms	Requires human involvement	Consumer, Provider
Intellectual rights terms	Requires human involvement	Consumer, Provider

by the community and is focused mainly on QoS terms and partially business terms for individual consumer-to-provider scenarios, not for cross-organizational scenarios. Current proposals on monitoring contractual terms are summarized in Table 3.

In the pioneering work of WSLA (Keller & Ludwig, 2003), the way of specifying, measuring, and monitoring the SLA parameters are described. Upon receipt of an SLA specification, the WSLA monitoring services are automatically configured to enforce the SLA. An event calculus based approach for monitoring SLAs in a utility computing scenario is presented in (Farrell, 2005). The recent work in (Skene et al, 2007) focuses on determining elements of SLAs that are monitorable at which degree. However, none of these approaches are sufficient to monitor completely contractual terms described by different specifications and associated with different services and to make decision strategies based on the outcome of monitoring.

RESEARCH ISSUES IN SERVICE CONTRACTS RECONCILIATION

Based on our analysis on service contracts, we have identified the following key research questions:

Research Issue 1: What Would be the Best Way to Manage Non-Functional Parameters and Service Contracts Associated with Service Instances Separately from WSDL?

Existing tools tend to assume that contracts can be annotated with service descriptions to facilitate the service discovery. However, a contract, if deployed in real business, will be associated with service instances. In particular, the concept of SaaS introduces various challenges to manage contracts associated with services because each customer, in principle, would have a different contract for each service instance.

Research Issue 2: Can we Have a Single Language to Represent All Contractual Concerns? When there is no Unified Language, do the Existing Languages/ Standards Satisfy the Requirement of Consumers for Representing Contractual Concerns?

Generally, all specification languages/standards focus on terms and conditions to be agreed by providers and consumers. Every language describes certain properties of services entirely. Unfortunately, at the time of writing, there is not a single language that fully supports all contractual terms. In the existence of multiple languages to represent contractual concerns, a unified set of standards for contractual terms is needed. To deal with multiple specifications, we can start from the consumer's point of view: we should provide a common language for the consumer to specify the requesting contractual terms. Such a common language can be used together with common ontologies for business, legal and intellectual terms. Based on consumer-specified contractual terms, we apply data integration techniques, such as schema mapping, meta-model and domain-specific languages, to query, evaluate and compare contractual terms given by different specifications/languages.

Interestingly, most of the present languages describing contractual concerns fail to represent hierarchical consumer preferences in contracts. For example, consider the following scenario where a consumer is interested in consuming a service with lesser cost and better response time. The consumer wishes to specify the order of preferences for the service by stating price as the priority term, i.e. the consumer wants to select a service with lower cost followed by higher response time. The present languages express contractual terms in a single level. Representing contractual terms in a hierarchical form (e.g., by a tree data structure) would be one simple solution but can enhance the description of consumer preferences and enable algorithms working on contractual terms to be more efficient. In this sense, the concept of constraints hierarchies (Borning et al, 1992) can be used to specify different preferences – mandatory and optional conditions - for selecting services (Guan et al, 2006).

Research Issue 3: Can we use Contract Negotiation/ Compatibility Algorithms, even though we do not have a Unified Specification Approach?

Presently, negotiation is mainly a manual process, and thus a full or partial automated contract negotiation is needed. Furthermore, in the case of composing services associated with different contracts, compatibility should be checked. However, this has so far attracted little attention.

Research Issue 4: Is Real Time Monitoring and Enforcement of the Contractual Concerns, in Particular, Legal and Intellectual Right Terms, of Dynamic Web Services Possible?

Though automated contract management and enforcement is highly desirable, it is obvious that many contractual terms cannot simply be monitored by the consumer and the provider in a fully automatic fashion, as assumed in most current work. A third party is needed in many cases, as discussed in (Keller & Ludwig, 2002). Moreover, for legal and intellectual right terms, besides third parties involvement, manual monitoring and enforcement will be required for contract enforcement. As a matter of facts, to support contract monitoring and enforcement in semi-automatic processes, the gap between technical contractual terms in business (human processing) level and in operational (machine processing) level, as discussed in (Arenas & Wilson, 2008), has to be addressed.

TOWARDS SERVICE CONTRACTS RECONCILIATION

Addressing the said questions will need a lot of effort from the service community. There are some initial results on the reconciliation of service contracts. In the next sub-sections, we discuss some of our research efforts on addressing the said research issues.

Management of Non-Functional Parameters and Service Contracts

A common way to manage non-functional parameters and service contractual terms in Web services is to annotate service descriptions with these parameters and terms. Such parameters and terms will help to foster the service discovery and contract negotiation and selection. However, with the complexity of service and data concerns (Truong et al., 2009) and as these concerns evolve, this way of annotation is not scalable and cannot support runtime change of concerns well.

One way to deal with this problem is to consider the publishing and management of non-functional parameters and service contractual terms in the evolution of service changes. The SEMF (Service Evolution Management Framework) (Treiber et al., 2008), for example, proposed to manage different types of non-functional parameters and service concerns by using a data representation based on hierarchal Atom feeds. This model allows us to describe catalog of services where each service can have several types of concerns, such as QoS, licensing, interfaces, etc., managed by feeds. Furthermore, feeds, information entries, will be associated with temporal information and can link to external sources, thus allowing not only different types of parameters but also different specifications to be included. For example, domain-specific models can be described in RDF and OWL and linked to the Web service information. Exhibit 3. shows an example of data concerns linked with Web service description using SEMF.

While the model used by SEMF can foster a flexible way to describe contractual terms, it can, however, describe such terms at the service level only: such terms are specified for the whole service. Therefore, it will not be suitable for the management of contractual terms that are associated with service operations or individual data resources provided by services. This issue is particular important for DaaS (data-as-a-service) which can be used to provide several data resources, each has a different contractual terms, e.g., like the Infochimps service (The Infochimp, 2010). This calls for another way to access and manage non-functional parameters and contractual terms to support also at the level of service operation and data resources.

Service Contract Mapping

In order to make service contracts comparable, techniques to map different service contracts described in different specifications and terminolo-

Exhibit 3. Example of using SEMF to managing different service and data concerns

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
    <id>urn:uuid:c7433422-49d6-4588-816d-c001cf00e9df</id>
    <updated>2008-03-25T16:28:15+01:00</updated>
    <title>USAddressVerification Service</title>
    <entry>
        <id>urn:uuid:4c44c6f7-7ee8-4b90-8348-931c1ef3d97e</id>
        <updated>2008-03-25T16:28:15+01:00</updated>
        <published>2008-03-25T16:28:15+01:00</published>
        <title>Interface</title>
        <summary>WSDL Interface </summary>
        <category label="Web Service Description" scheme="http://www.dmoz.org/
Computers/
          Programming/Internet/Service-Oriented_Architecture/Web_Services/
WSDL" term="Interface"/>
        <content type="application/wsdl+xml" src="http://ws.strikeiron.com/
USAddressVerification5?WSDL"/>
    </entry>
    <entry>
        <title>DaaS Concerns</title>
        <summary>Data Concerns</summary>
        <category label="Data Concerns"  term="DaaSConcern"/>
        <content type="application/xml" src="http://www.infosys.tuwien.ac.at/
prototyp/SOD1/

dataconcerns/samples/USAddressVerificationConcerns.xml"/>
    </entry>
</feed>
```

gies must be developed. In our view, the mapping of service contract specifications is not a static, but a dynamic process because specifications and terminologies as well as knowledge about them change over the time.

A possible solution to service contract mapping consists in performing semantic mediation based on ontology matching techniques (Euzenat & Shvaiko, 2007). However, the heterogeneity that characterizes service contract descriptions makes these techniques difficult to be applied. As a matter of fact, in some cases they cannot be applied since particular service contract descriptions (e.g., ODRL-S licenses) are not based on ontologies.

An alternative solution consists in performing the wrapping of service contract using a reference meta-model. An example of this solution is the Policy-Centered Metamodel (PCM) Wrapper (Comerio et al., 2009) that has been designed to support the semi-automatic mapping of ODRL-S, WSLA and WSOL contracts to PCM-based service contract descriptions. The PCM (De Paoli et al., 2008) has been chosen as common meta-model since it offers (1) expressive descriptions addressing qualitative contractual terms by means of logical expressions on ontology values and quantitative terms by means of expressions including ranges and inequalities and (2) structured descriptions by using the concept of *Policy* that aggregates different term descriptions into a single entity with an applicability condition. Moreover, as shown in (Palmonari et al. 2009), PCM-based service contracts allow for semantic mediation between contractual terms based on multiple ontologies (the evaluation and comparison of the PCM with other meta-models is out of the scope of this chapter.).

The PCM Wrapper provides support for two main activities:

- *Modeling and mapping service contract terminologies into a reference ontology:* since different contracts often utilize different terminologies, the definition of a reference ontology is needed. This ontology will contain semantic description of structures of service contract terms, their allowed values and the relationships among them. The description of techniques used by the PCM Wrapper to create the reference ontology is described in (Comerio et al., 2009).
- *Wrapping ODRL-S, WSLA and WSOL specifications to PCM Policy:* different techniques for performing the wrapping of a service contract in each type of language to PCM Policies are developed. For example, the wrapping of ODRL-S specifications is directly performed by applying fixed mapping rules since ODRL-S is characterized by a fixed profile model describing all the terms that can be in a service contract. For what concern specifications in WSLA and WSOL, the wrapping activity is more complicated since there is the necessity to handle the absence of knowledge (i.e., mapping rules) on specified terms, thus the service providers must define the mapping between their contract terms (i.e., text labels for WSLA and ontological concepts for WSOL) and concepts available in the reference ontology. Furthermore, lexical databases like WordNet could be integrated in order to support service providers to define mapping rules for identifying synonyms between text labels and ontological concepts defined in the reference ontology. Different types of ontology alignment tools could be also used to support the wrapping of WSOL specifications: (1) tools for defining a mapping between concepts in two different ontologies by finding pairs of related concepts (e.g., ANCHORPROMPT (Noy et al., 2003)) or by evaluating semantic affinity between concepts (e.g., H-MATCH (Castano et al., 2003)) and (2) tools for defining mapping rules to relate only relevant parts of the

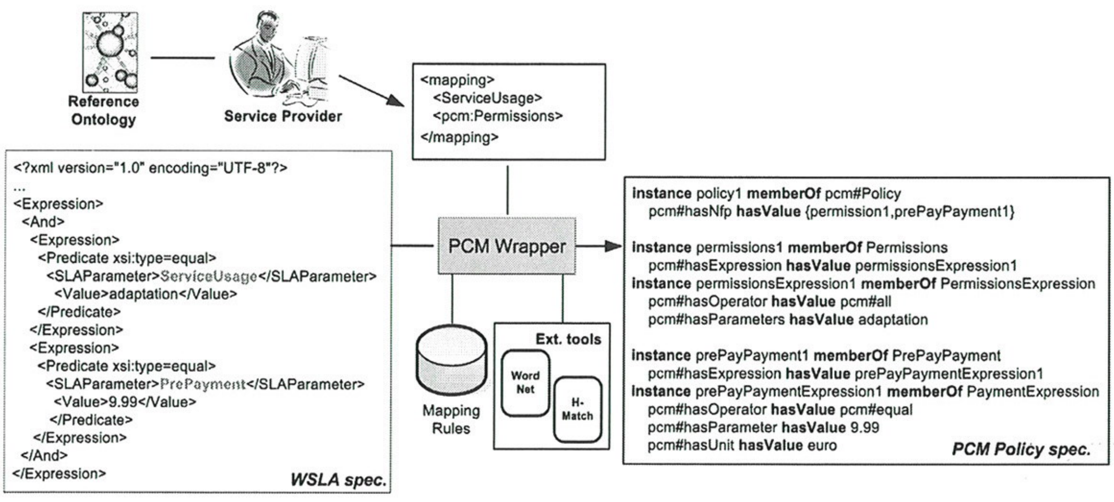
source ontologies (e.g., ONION (Mitra et al., 2001)).

Let us consider how the wrapping of a WSLA specification is performed: (1) parsing the specification in order to detect contract terms (i.e., SLAParameters); (2) searching the previously defined mapping rules related to the detected terms; (3) if mapping rules are not identified, use WordNet to identify a possible mapping between the SLAParameters and concepts available in the reference ontology and ask confirmation about the correctness of the mapping to the service provider; and (4) if the mapping is not correct or not available, ask to the service provider to perform the mapping manually.

Figure 2 illustrates the above-mentioned steps when wrapping the WSLA-based contract associated with the *Purchase Processing Service* of our scenario in Figure 1 that includes the following terms: *PrePayment = 9.99 Euros* and *ServiceUsage = "adaptation"* where *ServiceUsage* is a legal terms that can assume the value *adaptation* (i.e., the right of allowing the use of service interface only), *composition* (i.e., the right of service execution with the right of interface modification) or *derivation* (i.e., right of allowing modifications to the service interface as well as to the implementation of the service). In this example, a mapping rule for *PrePayment* already exists. On the contrary, the term *ServiceUsage* is not known and no rules are available. Moreover, no synonym relations are specified in WordNet between *ServiceUsage* and terms defined in the reference ontology. In order to handle this absence of knowledge, the service provider can navigate the ontology and map the SLAParameter *ServiceUsage* to any ontological concept. The result is that *ServiceUsage* is mapped to *Permissions*.

After this preliminary activity, the mapping proceeds considering the *Expressions* defined in each Service Level Objective of the WSLA specification. Each *Expression* follows the first order logic, including predicates and logic operators. According to the logic operators, different mapping rules can be applied. For example, In Figure 2, the logic operator *"And"* is used to specify the aggregation of two plain predicates stating conditions on *PrePayment* and *ServiceUsage*. The mapping to a PCM Policy consists of defining the concept instances related to all the plain predicates. The final result for the considered

Figure 2. Mapping between WSLA and PCM policy



example is a PCM Policy containing: (1) an instance of Permissions characterized by an expression stating that the value *adaptation* is assumed (i.e., *pcm#hasOperator hasValue pcm#all; pcm#hasParameters hasValue adaptation*) and (2) an instance of PrePayPayment stating that the amount is equal to 9.99 Euros (i.e., *pcm#hasOperator hasValue pcm#equal; pcm#hasParameter hasValue 9.99; pcm#hasUnit hasValue euro*).

Although techniques proposed by the PCM Wrapper are promising for dealing with the heterogeneity of service contract languages, several limitations must be addressed in order to make the PCM Wrapper a practical and widely accepted solution for service contract mapping. First, the current version supports the mapping of a limited set of contract specification languages (i.e., ODRL-S, WSLA and WSOL). Second, currently the PCM Wrapper can be used only by service providers that are familiar with ontologies defined in the Web Service Modeling Language (WSML). Therefore a user-friendly support for ontology navigation and multi-language ontology definition are still open research issues. Finally, lexical databases and ontology alignment tools need to be integrated in a next version of the framework.

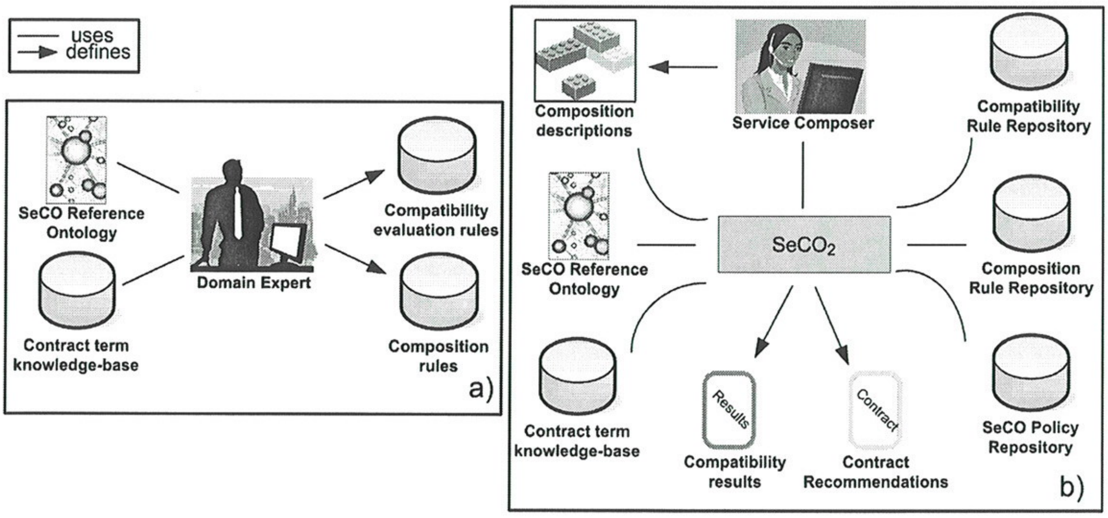
Service Contract Compatibility

With techniques to map different service contracts, it is possible to evaluate the compatibility between heterogeneous contracts associated with services to be composed into a composition. Our solution based on the SeCO2 framework (Comerio et al., 2009) also supports service composers to deal with service contracts in service composition. This framework takes as input a *composition description* (i.e., services involved in the composition, data and control flows) and produces the *compatibility results*: a list of identified incompatibilities among service contract terms. A contract is compatible with another contract if all contract terms are compatible. In order to produce compatibility results, SeCO2 utilizes the following resources:

- **SeCO Reference Ontology:** An extensible ontology contains semantic description of service contract terms, their allowed values, and the relations among them. The *SeCO Reference Ontology* is built based on the Policy-Centered Metamodel (PCM).
- **Contract Term Knowledge-Base:** A repository that contains additional information about properties described in the reference ontology. Among others, this repository contains the specification of the influences between each service contract term and *control* and *data flows*.
- **SeCO Policies:** Representing PCM-based WSML descriptions of service contracts including contract terms defined using the *SeCO Reference Ontology*.
- **Compatibility Evaluation Rules:** Rules for checking the compatibility between *SeCO Policies* of services involved in the composition. Each rule specifies how to evaluate the compatibility of a specific term defined in the reference ontology.
- **Composition Rules:** Rules for the service contract property composition. Each rule specifies how to evaluate a recommended value for a specific term to be included in a composite service contract.

Figure 3 shows (a) *setup-time* and (b) *run-time* activities supported by the SeCO2 framework. At setup-time domain experts are supported in the definition of rules to be applied in the compatibility evaluation and composition processes. The *Compatibility evaluation and composition rule definition* consists in defining a compatibility evaluation rule and a composition rule for each property specified in the *SeCO reference ontology*. This activity is performed considering data and control flows, as well as composition patterns, that can be associated with a service composition and using existing information in the reference ontology and in the contract term knowledge-base.

Figure 3. Setup-time (a) and run-time (b) activities in the SeCO2 framework



At run-time SeCO2 supports service composers offering the following functionalities: (1) *Composition description analysis*: analysis of the composition descriptions provided by the *Service Composer* in order to identify the services involved in the composition and data and control flows of the service composition; (2) *SeCO policy retrieval*: after having identified the services involved in the composition, their associated *SeCO Policies* are retrieved from the *SeCO Policy Repository*; (3) *Service contract compatibility evaluation*: contract terms are extracted from each *SeCO Policy* and evaluated according to the related *Compatibility evaluation rule* on the basis of the data and control flows of the service composition. Incompatibilities between service contracts are pointed out to the *Service Composer*; (4) *Composite service contract definition*: for each contract terms included in at least one of the retrieved *SeCO Policies*, the related *Composition rule* is executed considering data and control flows of the service composition. The result of each rule is included in the contract for the composite service.

Let us consider the provisional contracts C_{PS} and C_{AS} associated with the *Purchase Order Service* and the *Purchase Processing Service* of our supply chain management scenario in Figure 1. Let us assume that the two contracts present different *ServiceUsage* (i.e., the extent to which the service could be used, accessed, and value added, on the basis that any use outside the scope of license would constitute an infringement) terms. C_{PS} and C_{AS} specify *composition* and *derivation* as offered *ServiceUsage* terms respectively. The algorithm of SeCO2 will determine that C_{PS} and C_{AS} are compatible since *derivation* subsumes *composition*. For a complete compatibility analysis, the algorithm must know about the possible subsumptions, which imply a match that should occur, if the given contract clause is more permissive (accepts more) than the corresponding element in the other contract. For example, the subsumption rules for *ServiceUsage* are shown in Table 4.

There could also be a scenario when analyzing the compatibility of contracts where one of the contracts contains terms that the other contract does not. In certain cases, the absence of one or several of these terms does not affect the compatibility with the other contract. Further details about how the SeCO2 framework performs the contract

Table 4. Subsumption rules over ServiceUsage terms

Element 1	Element 2	Comparison	Redefinition
Derivation	Adaptation Composition	Derivation ⊃ Adaptation Derivation ⊃ Composition	Derivation Derivation
Adaptation	Composition Derivation	Composition ⊃ Adaptation Derivation ⊃ Adaptation	Composition Derivation
Composition	Adaptation Derivation	Composition ⊃ Adaptation Derivation ⊃ Composition	Composition Derivation

compatibility evaluation are in (Comerio et al., 2009).

The SeCO2 framework is an ongoing work in which service contract compatibility evaluation and composition rules are manually defined by domain experts. The support given by SeCO2 for automating this activity needs to be developed. Furthermore, only SeCO Policies defined in WSML can be evaluated. In order to manage different specifications, an enriched version of the PCM Wrapper (see the previous section) will be integrated into the framework.

CONCLUSION

In a dynamic market environment, the usage of Web services is based on contracts specifying the terms and conditions of using and provisioning of services. This chapter discusses the strengths and weakness of existing languages and standards for describing contracts in service-oriented computing. When the vision of software and data as a service is realized, service contracts are of paramount importance and we need to consider the interoperability of service contracts provided by different providers. Based on our study, we have suggested some guidelines and presented our solutions to reconcile the contractual concerns of Web services that could make the vision of service oriented computing more realistic.

We have discussed current approaches on the reconciliation of service contracts by focusing on the management of complex and diverse types of contract terms, the mapping of contract terms and the evaluation of service contract compatibility. However, as we have discussed, still there are many issues to be solved in these approaches in order to support the establishment of service contracts in multiple service providers' environments. In particular, the management of non-functional parameters and contracts should cover also service operations and data resources as well, the wrapping of contract specifications should be linked with human tasks workflows and external semantic matching services, and the evaluation of service contract compatibility should also deal with data resources and data concerns.

In our reconciliation work, semi-automatic models and techniques for monitoring and enforcement of legal and intellectual property right terms have not been addressed. As stated in previous sections, contractual terms cannot simply be monitored by the service consumer and the service provider in a fully automatic fashion. A third-party *Authority Service* is the solution proposed in most works available in the literature (Keller & Ludwig, 2002; Smith & Ramakrishnan, 2003; Cardoso et al., 2009) for dealing with multiple types of contract specifications including QoS, business, legal and intellectual right terms. In our view, for what concerns QoS and business terms, the realization of an automatic contract monitoring is feasible and this has attracted a considerable research effort. However, how to monitor and enforce legal and intellectual right terms is very much open. In our opinion, the legal and intellectual right terms of a service contract cannot be enforced in an automatic manner. The primary reason for not enforcing is the fact that technology cannot protect freedoms such as "fair use". In other words, "fair use" cannot be coded. We believe that technological enforcement encumbrances the unplanned reuse of services. We want to encourage such use. And we, along with the vision of Creative Commons (Lessig, 2004), are concerned that the ecology for creativity will be stifled by the pervasive use of technology to "manage" rights. Copyrights should be respected, no doubt. But we prefer they be respected the old fashioned way — by people acting to respect the freedoms, and limits, chosen by the author and enforced by the law. Based on these observations, we state that for legal and intellectual right terms, only the realization of a semi-automatic contract monitoring is feasible. The service customer must manually communicate related monitoring data to the authority providing details about how the authority can verify them.

ACKNOWLEDGMENT

This work is partially supported by the European Union through the FP7 projects COIN and COMPAS and by the SAS Institute srl (Grant Carlo Grandi).

REFERENCES

Aggarwal, R., Verma, K., Miller, J., & Milnor, W. (2004). Constraint driven web service composition in meteor-s. In *Proceeding of the 2004 IEEE International Conference on Services Computing* (SCC 2004), (pp. 23–30).

Anderson, A. (2004). *Web services policy language (WSPL)*. In *Proceedings of the Fifth IEEE International Workshop on Policies for Distributed Systems and Networks* (POLICY'04), (pp. 189–192).

Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., & Nakata, T. … Xu, M. (2007). *Web services agreement specification* (WS-Agreement). Grid Resource Allocation Agreement Protocol (GRAAP) WG. Retrieved from http://www.ogf.org/documents/GFD.107.pdf

Arenas, A., & Wilson, M. (2008). Contracts as trust substitutes in collaborative business. *IEEE Computer*, *41*(7), 80–83.

Borning, A., Freeman-Benson, B. N., & Wilson, M. (1992). Constraint hierarchies. *Lisp and Symbolic Computation*, *5*(3), 223–270. doi:10.1007/BF01807506

Cardoso, J., Winkler, M., & Voigt, K. (2009). A service description language for the internet of services. In *Proc. of the International Symposium on Service Science (ISSS 2009)*, March 2009.

Castano, S., Ferrara, A., & Montanelli, S. (2003). H-match: An algorithm for dynamically matching ontologies in peer-based systems. In *Proceedings of the 1st VLDB Int. Workshop on Semantic Web and Databases* (SWDB 2003).

Christensen, E., Curbera, F., Meredith, G., & Weerawarana, S. (2001). *Web services description language* (WSDL) 1.1. Retrieved from http://www.w3.org/TR/wsdl

Comerio, M., Truong, H. L., De Paoli, F., & Dustdar, S. (2009). Evaluating contract compatibility for service composition in the SeCO2 framework. In *Proceedings of International Conference on Service Oriented Computing* (ICSOC 2009).

Dan, A., Johnson, R., & Arsanjani, A. (2007). *Information as a service: Modeling and realization.* International Workshop on Systems Development in SOA Environments (SDSOA '07).

De Bruijn, J., Lausen, H., Krummenacher, R., Polleres, A., Predoiu, L., Kifer, M., & Fensel, D. (2005). *The Web service modeling language WSML* (D16. 1v0. 2). http://www.wsmo.org/TR/d16/d16.1/v0.21/20051005/

De Paoli, F., Palmonari, M., Comerio, M., & Maurino, A. (2008). A meta-model for non-functional property descriptions of Web services. In *Proceedings of the IEEE International Conference on Web Services* (ICWS 2008)

Demirkan, H., Goul, M., & Soper, D. (2005). Service level agreement negotiation: A theory-based exploratory study as a starting point for identifying negotiation support system requirements. In *Proceedings of the 38th Hawaii International Conference on System Sciences.*

Euzenat, J., & Shvaiko, P. (2007). *Ontology matching*. Springer-Verlag.

Farrell, A., Sergot, M., Salle, M., & Bartolini, C. (2005). Using the event calculus for tracking the normative state of contracts. *International Journal of Cooperative Information System*, *4*(2).

Gangadharan, G. R., D'Andrea, V., Iannella, R., & Weiss, M. (2007). *ODRL service licensing profile (ODRL-S)*. In *Proceedings of the 5th International Workshop for Technical, Economic, and Legal Aspects of Business Models for Virtual Goods.*

Giallonardo, E., & Zimeo, E. (2007). More semantics in QoS matching. In *Proceedings of International Conference on Service-Oriented Computing and Application* (SOCA'07), (pp. 163–171).

Gimpel, H., Ludwig, H., Dan, A., & Kearney, B. (2003). PANDA: Specifying policies for automated negotiations of service contracts. In *Proceedings of the First International Conference on Service Oriented Computing.*

Grosof, B., Reeves, D., & Wellman, M. (2002). Automated negotiation from declarative contract descriptions. *Computational Intelligence*, *18*(4).

Guan, Y., Ghose, A. K., & Lu, Z. (2006). Using constraint hierarchies to support QoS-guided service composition. In *Proceedings of the IEEE ICWS*, (pp. 743–752).

Kamoda, H., Yamaoka, M., Matsuda, S., Broda, K., & Sloman, M. (2005). Policy conflict analysis using free variable tableaux for access control in Web services environments. In *Proceedings of the 14th International World Wide Web Conference* (WWW).

Keller, A., & Ludwig, H. (2002). Defining and monitoring service-level agreements for dynamic ebusiness. In *Proceedings of the 16th USENIX Conference on System Administration*, (pp. 189–204).

Keller, A., & Ludwig, H. (2003). The WSLA framework: Specifying and monitoring service level agreements for Web services. *Journal of Network and Systems Management*, *11*(1). doi:10.1023/A:1022445108617

Kritikos, K., & Plexousakis, D. (2006). Semantic QoS metric matching. In *Proceedings of the European Conference on Web Services* (ECOWS'06), (pp. 265–274).

Kuno, H., Alonso, G., Casati, F., & Machiraju, V. (2003). *Web services - Concepts, architectures and applications*, 1st ed.

Lessig, L. (2004). The creative commons. *Montana Law Review*, (Winter): 1–13.

Lewis, L., & Ray, P. (1999). *Service level management definition, architecture, and research challenges*. In *Proceedings of the Global Telecommunications Conference* (GLOBECOM).

Lupu, E., & Sloman, M. (1997). *A policy based role object model*. In *Proceedings of the International Enterprise Distributed Object Computing Conference* (EDOC).

Martin, D., et al. (2004). Semantic markup for Web services. Retrieved from http://www.w3.org/Submission/OWL-S/

Michael, S. (2002). *Engineering electronic negotiations*. New York, NY: Kluwer Academic Publishers.

Mitra, P., Wiederhold, G., & Decker, S. (2001). *A scalable framework for the interoperation of information sources* (pp. 317–329). Stanford University.

Muller, N. (1999). Managing service level agreements. *International Journal of Network Management*, *9*, 155–166. doi:10.1002/(SICI)1099-1190(199905/06)9:3<155::AID-NEM317>3.0.CO;2-M

Noy, N. F., & Musen, M. A. (2003). The prompt suite: Interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, *59*(6), 983–1024. doi:10.1016/j.ijhcs.2003.08.002

OASIS. (2005). *ebXML CPP and CPA Technical Committee: Collaboration protocol profile and agreement specification version 2.1*. Retrieved from http://www.oasis-open.org/committees/ebxml-cppa/

Palmonari, M., Comerio, M., & De Paoli, F. (2009). Effective and flexible Nfp-based ranking of Web services. In *Proceedings of International Conference on Service Oriented Computing* (ICSOC 2009).

Sahai, A., Durante, A., & Machiraju, V. (2002). *Towards automated SLA management for Web services*. Technical Report HPL-2001-310 (R.1), Software Technology Laboratory, HP Laboratories, Palo Alto, USA.

Skene, J., Lamanna, D., & Emmerich, W. (2004). Precise service level agreements. In *Proceedings of 26th International Conference on Software Engineering* (ICSE).

Skene, J., Skene, A., Crampton, J., & Emmerich, W. (2007). *The monitorability of service-level agreements for application-service provision*. In *Proceedings of the 6th International Workshop on Software and Performance* (pp. 3–14).

Smith, T., & Ramakrishnan, L. (2003). Joint policy management and auditing in virtual organizations. In *Proceedings of the Fourth International Workshop on Grid Computing*.

The COMPAS Project. (2010). *Compliance-driven models, languages, and architectures for services*. Retrieved December 2010, from http://www.compas-ict.eu/

The Infochimps. (2010). Retrieved December 2010, from http://infochimps.org/

Toma, I., Roman, D., & Fensel, D. (2007). On describing and ranking services based on non-functional properties. In *Proceedings of the Third International Conference on Next Generation Web Services Practices* (NWESP '07), (pp. 61–66).

Tosic, V., Pagurek, B., Patel, K., Esfandiari, B., & Ma, W. (2003). Management applications of the Web service offerings language. In *Proceedings of the 15th Conference on Advanced Information Systems Engineering*.

Treiber, M., Truong, H. L., & Dustdar, S. (2008). SEMF - Service evolution management framework. In *Proceedings of the 34th EUROMICRO Conference on Software Engineering and Advanced Applications* (SEAA).

Truong, H. L., & Dustdar, S. (2009). *On analyzing and specifying concerns for data as a service*. In *Proc. of the 4th IEEE Asia-Pacific Services Computing Conference* (APSCC 2009), (pp. 87-94).

Truong, H. L., Gangadharan, G. R., Treiber, M., Dustdar, S., & D'Andrea, V. (2008). *On reconciliation of contractual concerns of Web services*. 2nd Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop (NFPSLA-SOC'08), co-located with The 6th IEEE European Conference on Web Services, Dublin, Ireland.

Vedamuthu, A., Orchard, D., Hirsch, F., Hondo, M., Yendluri, P., Boubez, T., & Yalcinalp, U. (2007). *Web services policy (WS-policy) framework*. Retrieved from http://www.w3.org/TR/ws-policy

Viega, J. (2009). Cloud computing and the common man. *Computer*, *42*, 106–108. doi:10.1109/MC.2009.252

Zeng, L., Benatallah, B., Ngu, A., Dumas, M., Kalagnanam, J., & Chang, H. (2004). QoS-aware middleware for web services composition. *IEEE Transactions on Software Engineering*, *30*(5), 311–327. doi:10.1109/TSE.2004.11

KEY TERMS AND DEFINITIONS

Contractual Terms: An instance built as a constraint based on non-functional properties characterizing concerns associated with services and their data. They basically describe the associated non-functional parameters of a service.

Non-Functional Parameters: Qualitative and quantitative parameters of a service covering QoS, business, legal, and intellectual rights issues related to the service usage.

Service Contract Compatibility: A process to verify whether there are no conflicting contractual terms among different contracts.

Service Contract Mapping: A process to map different service contracts described in different specifications and terminologies.

Service Contract Reconciliation: A process to reconcile the semantics and syntax of multiple service contracts specified in different languages.

Service Contract: A legal agreement between a service provider and service consumer that is enforced by the law.