Service Orientation: Licensing Perspectives

G.R.Gangadharan
Telematica Instituut, Enschede, The Netherlands
gr@telin.nl

Vincenzo D'Andrea
University of Trento, Trento, Italy
dandrea@disi.unitn.it

Abstract. Services proliferate in myriad domains, with their seamless potentiality, raising new issues such as the need to articulate rights and obligations associated with services. The current research on services insights primarily on the aspects of technology and sparsely focuses on business and intellectual values associated with services. Service licensing is a promising way to manage the normative aspects of the relationship between service consumers and service providers. Conceptualizing service licenses and making them in machine interpretable form would promote broader usage of services. In this paper, we analyze and formalize service licensing clauses and unambiguously describe a service license in machine interpretable form.

Keywords: Web Services, Service Oriented Computing, Intellectual Rights, Licenses, Rights Expression Languages

1. Introduction

Service Oriented Computing (SOC) allows the software-as-a-service concept to expand to include the delivery of complex business process and transactions as a service, allowing applications to be constructed on the fly and services to be reused everywhere (Alonso et al., 2004). Services are autonomous, platform independent, business functions that are published and described using standard description and publication languages. They are remotely invocable over different networks using standard protocols. A service is represented by an interface part defining the functionality visible to the external world and an implementation part realizing the interface. A service can be accessed whenever required. However, a service remains idle until a request for invocation arrives.

In a dynamic market environment, the usage of services is governed by agreements that specify the terms and conditions of using and provisioning the services. A license is an agreement between parties which includes all transactions between the licensor and the licensee, in which the licensor agrees to grant the licensee the right to use and access the asset under predefined terms and conditions (Classen, 1996).

The concept of software licensing has emerged when the production and sale of individual software came into the market. While licensing was already present in the software world, the move to mass market software has introduced shrink wrap licenses, the terms of which can only be read and accepted by the consumer after using the product. With the advent of the Internet-based marketing and distribution strategies, click wrap licensing (similar to shrink wrap licensing) continues as one of the universal practices. The transformation from software as a product to software-as-a-service is the reflection of the transition of the distribution of software. As SOC extends the concept of software-as-a-service to include the delivery of complex business processes as a service, there arises a requirement for developing service licensing strategies.

Similar to software licensing, service licensing is extremely important for distribution of services. Software serves as a stand-alone application. In contrast, the rationale behind services is making network accessible operations available anywhere and anytime. A consequence is that while software is separately installed and executed using the computing resources internal to an organization, a service is often executed using external resources. In addition, while software is designed with particular use in mind, services are designed to facilitate potential reuse. The design of services supports loose coupling, wherein a service acquires knowledge of another services, still remaining independent. Software is designed to incorporate a set of specific functions and usually is not allowed to be integrated with other software. Further, software could be restricted by the organizational boundaries and could not communicate with other software crossing the boundaries. The fundamental to service orientation is to design services to encourage composition. Thus, the distinguishing characteristics and nature of services prevent services directly to adopt the licensing models of software.

One could question if licenses are a legal artefact applicable to services. First of all we notice that the service implementation is by means of software, so that part of a service would be "license-able". The service interface is also license-able, being the result of intellectual work aimed at describing the service functionalities

with a formal language. In general, we submit that a service can be licensed, being the result of intellectual work that can be accessed and used (Gangadharan, 2008). In this paper, we conceptualize service licenses and express in a machine interpretable way. The paper is organized as follows: We present a detailed analysis of service licensing clauses depicting the anatomy of a service license in Section 2. Section 3 presents a formal representation of licensing clauses in order to unambiguously describe a service license. In Section 4, we present a language for denoting license clauses in machine interpretable form. Section 5 discusses the related works in this field, showing the distinct contribution of this paper, followed by conclusions in Section 6.

2. Service Orientation of Software

Service provider exposes the business functionality in the form of service. Service provider, in other words, the owner of web services, decides the functions to be exposed, negotiation, and pricing strategies. A service is advertised in a public registry through ***Publish*** operation. Service requester can be either a consumer consuming services directly or a provider acting as an aggregator of services. Ultimately, service requester is the user for the published services. A service requester communicates with service broker through ***Find*** operation to select the most appropriate service to satisfy specific requirements. Further, the Service requester interacts with the concerned service provider through ***Bind*** operation and uses the service. Service broker is a registry where the descriptions of the services are stored. Based on the information in the registry about a service, service requester contacts the corresponding service provider and thus consumes the service.

A service is represented by an interface part defining the functionality visible to the external world as a means to access the functionality and an implementation part realizing the interface. The interface part is the description of the service having all the specifications for invoking the service. The service description contains the operations (like a method signature in a programming language), protocol, data formats, specifying how a service interface is implemented by the service provider. The implementation part is the realization of the interface. A service implementation could provide the functionality directly or could combine other services to provide the same functionality. A service can be implemented in any language.

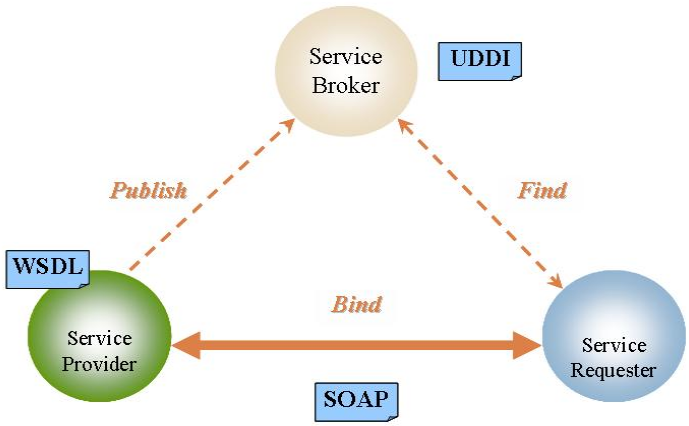

Figure 1. Service Oriented Computing (Instances with Web Service)

The application of SOC model (see Figure 1) to web resources is manifested by web services to provide a loosely coupled model for distributed processing. Web services are the enabling technology, standardized to construct and integrate applications and organizational interfaces as services, using the Internet as the communication medium and open Internet-based standards (Weerawarana et al., 2005). The Web Services Definition Language (WSDL) is an XML based interface definition language, describing services as a collection of messages (abstract descriptions of the data being exchanged) and port types (abstract collections of operations), separated from their concrete network deployment or data format bindings. Universal Description, Discovery, and Integration (UDDI) enables publishing and accessing WSDL specifications in directories. Simple Object Access Protocol (SOAP) is a platform and language independent protocol, providing a way of communication between applications.

In engineering service oriented applications, the concept of composition plays a central role. A service oriented application will in general rely on using several other services, or their operations. Service composition (Dustdar & Schreiner, 2005) is related to the implementation of a service whose internal logic involves the

invocation of operations offered by other services. Thus, service composition can be perceived as a way of developing a new service, for execution of which a composite service depends on the service(s) being composed. The goal of SOC is also to allow dynamic, runtime, identification, and binding of composed services. That is, given an operation a service is found which can perform the wanted operation. Beside the functional description of operations, considering non-functional parameters such as response time, cost is also significant in service composition.

Though the concept of arbitrarily mixing and matching the services from different service providers seems interesting, the basic clauses of service licenses associated with services would enforce certain terms and conditions on composition. Questions of ownership and distribution could impede composition, thereby impacting the reuse of services.

3. Conceptualization of Service Licensing

In traditional software and components, a license is a legal document comprising the permissions and prohibitions imposed on software. A software producer can make decision on how her software could be consumed and what rights she could provide the software for consumers to access. Similarly, a consumer can also have a freedom of knowing how she can use the software. The expression of these rights together with additional clauses, are traditionally seen as a license. Under a software license, the licensee is permitted to use the licensed software in compliance with the specific terms of the license. As a starting point to define the goals of a service license, we will consider the objectives of a software license (Robertson, 1990).

- 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 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 will also specify information on service delivery, acceptance, and payment. Some of the key concepts of a service license are elucidated as follows (Although, we have undertaken an endeavor to represent a "standard form" of a service license, we do not claim that the given anatomy of a service license is complete. It is almost impossible to generalize all the terms of a license.).

3.1. Subject

The Subject of a license relates to the definition of the service that is being licensed. Similar to the context of traditional software and components, all rights remain vested in the licensor of the service. The licensees reserve all kinds of rights for any services or applications created by them based on or utilizing the given licensed service.

As an example, we examine the Subject clause of Google AJAX Search API, which is given as follows (http://code.google.com/apis/ajaxsearch/terms.html).

> *[1.1] The API consists of JavaScript and associated service protocols that allow You to display results from Google searches, including text and URL results from Google Web Search and results from searches on other Google services ("Google Search Results") on your website, subject to the limitations and conditions described below. The API is limited to allowing You to host and display Google Search Results on your site, and does not provide You with the ability to access other underlying Google Services or data. Subject to the limitations and conditions described below, You may use the API to enable end users of your site to designate Google Search Results for you to host and display in conjunction with other information that they upload or post to your site. The API may be used only for services that are accessible to your end users without charge.*

Similar to a software license, a service license need not live up to any particular standard of performance or function. The licensor can provide a right to use the service "as is". However, this is obviously not good for the licensor, unless the price and other terms reflect this. The best position for the person licensing the service is that the service must meet clearly the defined specifications of access criteria and performance limits.

3.2. Scope of Rights

The Scope of Rights are the rights the licensor authorizes the licensee to exercise in a service. Most software licenses (World Intellectual Property Organization, 2004) restrict the scope by offering the rights to use object

code and limited rights to make archival copy. Free/Open Source Software (FOSS) (Rosen, 2004) licenses allow free access to the source code of software. In contrast, the nature of services induces broader license grants to support reusability of services with other applications. Most services may even require the rights of modifications to provide value added services. As the scope of rights is the provision stipulating the rights the licensor grants to the licensee, the following scenarios arise in SOC that determine the flexibility of rights.

3.2.1 Scope of Rights Associating Service Interface

As service interfaces together with bindings are publicly available, several services could be created with the same interface with or without modifications. These services can vary in their performances. Following are common scenarios of reproducing interface with modifications.

The interface of a service could be modified by changing the name of some operations such as for translation (the expression of a service in a language other than that of the original version). Assume that the interface of a service S_A provides spell checking of a word, *Spell (word)*. The interface of S_A could be translated by another service S_B to provide a spell checking operation in Italian language, say *Ortografia(parole)*. In this case, S_B simply translates the interface of S_A and results in the Italian version of S_A.

The interface of a service could be reproduced by data translation. For example, consider a service S_P that provides the temperature of a city in Celsius scale. Another service S_Q could provide the temperature in Fahrenheit scale by modifying the interface of S_P. Though S_Q seems as a new service, it simply translates the data of S_P.

The interface of a service could be modified by making changes in service parameters or by some pre-processing and/or post-processing of the given service. For example, the interface of S_A providing spell checking of a word, *Spell(word)*, could be modified by S_B to provide spell checking for a set of words, say, *Spellwords(w1,w2,w3)*.

A service interface and service implementation could be licensed differently. The interface could even be licensed with a clause denying modification, though the source code of the interface is openly available. Thus, irrespective of service implementation, interface rights become highly significant in SOC.

3.2.2 Scope of Rights Associating Service Implementation

In line with software, a service provider could distribute (possibly with a fee) a service software to other providers. Another provider will then be able to offer a similar service to the original one, possibly with different data or using a different business model.

A service could allow its implementation to be used as an executable by other services. Consider that a service S_A allows its implementation as an executable for another service S_B. However, S_A could restrict S_B not to modify any operations of S_A. By distributing a service software in the form of executable, a service provider can restrict other services not to modify original service operations.

In contrast to this model, a service even could allow other services to modify its implementation. To modify a service implementation, the service should be a Free/Open Service (D'Andrea & Gangadharan, 2006). If a service interface and implementation are allowed to be copied, new independent services could be created by replicating source codes of implementation and interface. Theoretically, there will be no differences (not considering network delays!) in performances of these services.

Service composition is referred as a combination of independently developed services into a complex service. Let S_B be a service providing a spell checking operation *Spell(sentence)* for sentences, that could compose internally operations for spelling of words with a parser. S_B could be designed in such a way that *Spell(word)* of S_B directly invokes the operation of S_A, executing on the host of S_A. In the absence of S_A, S_B fails to perform. Though the underlying assumption of SOC is composition, a service can deny other services to compose with itself.

3.2.3 Extended Scope of Rights

Copyright law refers to attribution as the requirement to acknowledge or credit the author of a work which is used or appears in another work. A service could expect attribution for its use by other services. The requirement of attribution can be specified explicitly in a service license.

A service can require another service to follow its same licensing terms and conditions. The requirement of a service to follow the same licensing terms is similar to Copyleft of GPL (Stallman, 1999) or Sharealike of Creative Commons (Lessig, 2004). From the perspective of service providers and developers, this right of services could be seen as a restriction imposed to the new service, that allows the value addition solely with the same conditions that the original has. However, from the perspective of a service consumer, this could be viewed as an ultimate guide for using any value added services inheriting from a particular similar termed service.

A service provider can allow a service to be used for commercial use or non-commercial use. We follow the best practice guidelines proposed by Creative Commons to clarify the meaning of non-commercial use of a service.

Following are (partial) Scope of Rights of Amazon web service license (www.amazon.com/AWS-License-home-page-Money/b?ie=UTF8&node=3440661).

> **4.1. Permitted Uses Generally.**
>
> 4.1.1. You may write a software application or Web site("Application") that interfaces with the Services. You acknowledge that we may change, deprecate or republish APIs for any Service or feature of a Service from time to time, and that it is your responsibility to ensure that calls you make to any Service are compatible with then-current APIs for the Service.
>
> 4.1.2. You may make network calls or requests to the Services at any time that the Services are available, provided that, unless otherwise set forth in an applicable Authorized Use Policy set forth herein for any Service ("AUP"), you (or if you build and release an Application, each installed copy of your Application) may not exceed the maximum file size or maximum calls per second limit (if any) set forth for any particular Service in its AUP (or, in the event the AUP for a Service does not indicate a maximum file size, greater than 40K).
>
> **4.2. Restricted Uses Generally.**
>
> 4.2.1. You may not interfere or attempt to interfere in any manner with the functionality or proper working of the Services.
>
> 4.2.2. You may not compile or use the Amazon Properties or any other information obtained through the Services for the purpose of direct marketing, spamming, unsolicited contacting of sellers or customers, or other impermissible advertising, marketing or other activities, including, without limitation, any activities that violate anti-spamming laws and regulations.
>
> 4.2.4. Subject to the terms and conditions of this Agreement, you may generally publicize your use of the Services; however, you may not issue any press release with respect to the Services or this Agreement without our prior written consent.

The clauses regarding the Scope of Rights of a service license describe the right to use, modify, compose, or derive the service. To the best of our knowledge, the analysis of license clauses for the APIs of Amazon, Google, and Yahoo! reveals the surprising fact that they do not focus on the Scope of Rights specific to a service. In other words, the Scope of Rights clauses of these licenses are similar to a software license and silent about the clauses of economic and moral rights associated with services as described.

3.3. Financial Terms

Software licenses generally stipulate an up-front or one-time license fees. This assumption of payment method is not suitable for services because multi-faceted use and provision of a service exists. The financial terms of a service license depends on the perspectives of service providers and service consumers. A service consumer assesses how much she can afford to pay for the service and how will the service be consumed. From the perspective of service providers, the estimation of cost of producing a service and amount of return on investment plays the critical role in determining the price.

Consider that a *Stockquote* service is provided by a service provider. The value of this service lies in the on-time delivery of stock prices to consumers. This type of service is based on critical data that change continuously and requires a considerable investment in provision of the service. Certain consumers may expect a *Stockquote* service to deliver stock quotes with a delay, but with lowering cost for their applications. In this case, a provider can offer the service based on varying levels of functionalities and non-functional properties, as demanded by consumers. Thus, the price of a service becomes directly proportional to the features offered. In both cases, service consumers define in what way the service is going to be used by them. A service consumer may invoke the service unlimited times. In another case, consumers may wish to pay based on the number of times they use a given service. We can generalize the pricing models of services as follows (Gunther et al., 2006).

- ***Transaction based model*** allows service providers to charge for each use, as the license defines the term 'use'. The use of services can be continuously recorded and monitored by service management systems. This model of pricing is quite similar to charging true utilities like electricity and water.
- ***Subscription based model*** allows service consumers to purchase services for a fixed term, during which time they automatically receive full support from service providers including any upgrades or feature enhancements. Service consumers typically pay periodically.

Following (partial) clauses represent the Financial Terms of Google Apps Premier Edition Agreement (http://www.google.com/a/help/intl/en/admins/premier _terms.html).

> [6.1] ... End User Accounts that are ordered and for which Customer pays Fees prior to the General Availability Date will be considered to be paid up through the one year anniversary of the General Availability Date. Thereafter, each End User Account will be subject to the standard yearly fees, as determined by Google and set forth in the Quote.
> [6.3] Customer shall pay Google the fees in the amount and on the terms specified in the Quote, free and clear of, and without any reduction for, any and all taxes. Customer shall pay any taxes, including sales, use, personal property, value-added, excise, customs fees, import duties or stamp duties or other taxes and duties imposed by governmental agencies of whatever kind and imposed with respect to all transactions under the Agreement, including penalties and interest, but specifically excluding taxes based upon Google's net income.

It would be appropriate if a service supports all pricing models. In this case, we propose that a service consumer should select a pricing model appropriate for her use and the provision of a pricing model for services should not be decided by service providers.

3.4. Warranties, Indemnities, and Limitation of Liabilities

In commercial law (Goode, 2006), a warranty is referred as a promise that something sold is as factually stated or legally implied by the seller. The licensor warrants the rendering of services in a professional and workman like manner. Also, licenses specify that the operations of the licensed service are not beyond errors. Software licenses have limited and short-term warranties (Chavez et al., 1998). In contrast, reuse being the cornerstone of the services paradigm, a service provider should offer warranties and support provisions for attracting and retaining consumers. Warranties are inexorably linked to the motivations of licensees who access and use a given service in their applications. As services can involve inter-organizational collaboration, more balanced and impartial warranties are fundamental to create a trustworthy environment.

Consider a *Stockquote* service that offers stock quotes having delay of five minutes and with a mean response time (time delay for accessing this service) of 2 milliseconds. A service license can (optionally) specify these quality of services (as in the case of a SLA) with other licensing clauses. The violations of these terms (for example, if response time is more than 2 milliseconds) will lead to termination of the license and even apply penalties for the licensor subject to licensing clauses.

Indemnity is a legal exemption from incurred penalties or liabilities. An indemnification clause (Classen, 1996) protects a licensee against a third party's claim on the infringement of the third party's rights by the licensee's use of the licensed service. In context with a software component license (Chavez et al., 1998), the licensor can indemnify the licensee for intellectual rights infringement by the licensed service, but only to the extent those infringement claims arise from the licensee's authorized use of the licensed service. However, the licensee is required to bear the cost of defending infringement claims to the extent those claims arise from the combination of the licensed service with licensee's own application/service, or from modifications to the licensed service by the licensee, or from licensee's misuse of the licensed service.

Limitation of liability (Chavez et al., 1998) limits the liability of the licensor and the licensee under the license. Under this clause, both parties disclaim liability for unforeseeable damages (network errors or hosting server problems) or indirect damages. Often, limitation of liability clauses includes a ceiling for monetary liability.

Following are the partial representation of Indemnity, Disclaimer of Warranties, and Limitation of liability clauses of Google AJAX Search API license (http://code.google.com/apis/ajaxsearch/terms.html).

> [5] YOU EXPRESSLY UNDERSTAND AND AGREE THAT:
> [b.] GOOGLE MAKES NO WARRANTY THAT (i) THE SERVICE WILL MEET YOUR REQUIREMENTS, (ii) THE SERVICE WILL BE UNINTERRUPTED, TIMELY, SECURE, OR ERROR-FREE, (iii) THE RESULTS THAT MAY BE OBTAINED FROM THE USE OF THE SERVICE WILL BE ACCURATE OR RELIABLE, ...
> [6] YOU EXPRESSLY UNDERSTAND AND AGREE THAT GOOGLE SHALL NOT BE LIABLE TO YOU FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES, ...
> [8.7] Nothing in the Terms of Use should be construed to confer any rights to third party beneficiaries.

Although the issues related to warranties, indemnities, and limitation of liabilities can be legally complex, these clauses identify the bearer for financial risk of service failures and the bearer for the risk that a third party will bring a legal action claiming that the service violates his/her intellectual rights.

3.5. Evolution

Service evolution results from modifications by a service provider in functional and/or non-functional specifications of a service, and results in new releases or new versions (Canfora, 2004; Gangadharan et al., 2007). A new version of a service may behave functionally different than its previous version. Even if a service functionality remains unaltered, any changes in physical infrastructures could cause unexpected behaviours in non-functional properties. The release of new versions of traditional software and components does not affect functioning of systems in which they are integrated, as these are independent from new versions. However, services are invoked rather than being integrated. Thus, service consumers or even service aggregators cannot control the unexpected behaviour due to evolution of services.

Changes to a service interface (service operations and parameters) and/or service implementation could directly impact the normal use of a service at the consumer's side. The new version of the service could also have a different license from the previous one. Furthermore, a service changing its service license could be referred as an evolved service.

The clause of evolution of a Google AJAX Search API is given as follows (http://code.google.com/apis/ajaxsearch/terms.html).

> [1.2] Google reserves the right to release subsequent versions of the API and to require You to obtain and use the most recent version. Google may modify the Terms of Use at any time with or without notice, and You can review the most current version of the Terms of Use online at any time at http://code.google.com/apis/ajaxsearch/ terms.html, or such future URL as Google may designate.

The licensee wants ideally to receive broader rights to new releases and enhancements of services. The licensor wants to limit the commitments to the licensee for the sake of vitality of the business. Thus, by evolution clauses, service licenses address the rights to future versions/releases of the service for a licensee.

4. Formalization of a Service License

For drafting machine interpretable licenses, clauses of a service license should be unambiguous. We formalize service licensing clauses to avoid ambiguity in describing service licenses as follows:

Subject : The subject of a license relates to the definition of the service being licensed. This includes an unique identification code for the service, a name for the service, location of the service and other additional relevant information.

Scope of Rights: The scope of rights of a service license reflects what could be done with the service. We represent the set of operations (listed in the service interface) of a service *S* by *O(S)* and a single operation by *o* where $o \in O(S)$. We denote a service license by *L(S)*. We refer to a service as independent service that will execute in a different context or that is owned and/or maintained by a different organization. With the symbol $\S$, we denote the dependence relation of a service operation (or a set of operations) in left side on another operation(s) in right side.

Adaptation: Adaptation signifies the making of a new independent service from an existing service interface without modifying the implementation. A service S is reproduced as another independent service S^* by adaptation if $O(S^*) \neq O(S)$ and S and S^* are independent in execution.

Composition: Composition is the federation of a service with other remote services. A service S is said to be composite if its operations depend on operations belonging to n other services, represented as, $O(S) \S \{o_f : o_f \in O(S_i), i=[1,..,n]\}$.

Derivation: Derivation of a service, inspired by FOSS, is a novel aspect of creating a new service from existing service, modifying service interface and service implementation. A service S^* is said to be derived from another service S if $O(S^*) \supseteq O(S)$ on satisfying the following two conditions. (i) To exist S^*, S should be a Free/Open Service and (ii) S and S^* should be independent in execution.

Attribution: Attribution is ascribing a service to the entity responsible for its creation. If a service S^* uses another service S, then attribution to S is $A(S^*) \supset A(S)$.

Sharealike: A service *S* could expect another service S^* being adapted or composed or derived to reflect the same terms and conditions of S. In other words, $L(S) = L(S^*)$ where S^* uses *S*.

Non-Commercial Use*:* A service can be used either for non-commercial purposes or for commercial purposes. By including the clause of non-commercial use, a service denies its use for commercial purposes.

Financial Terms : Service consumers make payments either as royalties or lump sum for using services. Generally royalties are based on per unit sales. In case of services, royalties can be viewed as the amount for per-use of a service (not considering a possible discount for volume sales). For a payment of p per use, a service consumer has to pay $R = n * p$ where n being the number of times the service has been used. In this case, p can be renewed annually or over the life of license. Lump sum payments are alternative method to royalties. Sometimes lump sum payments are also used in addition to royalties. In case of services, a lump sum payment can be paid by a service consumer before using the service (prepay) or at a later stage (postpay). By paying lump sum amount, a licensee obtains rights to use the given service over a period t (irrespective of number of times that services being invoked). There can be other possible models defining payment mechanisms for services.

Warranties, Indemnities, and Limitation of Liabilities : Warranties describe functional and non-functional properties of services, provided as a way of attracting and retaining consumers. In a service license, the representation of warranties is optional. Warranties are generally similar to the notions given in WSLA (Ludwig et al., 2003) and in SLAng (Skene et al., 2004). A service license also specifies indemnification clauses (Chavez et al., 1998), a way of defence by the licensor for the licensee if a third party sues the licensee, alleging that the licensee's use of the licensed software infringes or violates the third party's intellectual rights. Limitation of liability clauses restricts the liability of each of parties under the license agreement.

Evolution : Evolution clauses define access rights to an evolved service. Modifications in functional and/or non-functional properties are represented by new releases or new versions. A service *S* is substitutable by another service S^* if $O(S) \equiv O(S^*)$ holds. A service *S* is generic over by another service S^* if $O(S) \supseteq O(S^*)$ holds.

5. ODRL-S: A Language for Expressing a Service License

As some legal doctrines are inherently flexible and vague, the translation of legal concepts into a machine interpretable language is highly complex. We have made an endeavour to represent service licensing clauses in a machine interpretable way (in a best possible manner). We have developed ODRL-S, a language that extends the Open Digital Rights Language (ODRL) to implement the clauses of service licensing. ODRL-S is nominated by the ODRL consortium for approval as a standard (http://www.odrl.net/ Profiles/Services/).

ODRL-S is designed as a complementary language to unambiguously describe service licensing clauses in machine interpretable form. The salient features of ODRL-S are as follows. (i) ODRL-S is simple yet powerful and fully extensible language. (ii) ODRL-S can specify licenses at service level and service operation level. (iii) ODRL-S can be used with any of existing service description standards and languages.

Following is a scenario (see Figure 2) that illustrates the use of services with licenses specified in ODRL-S. Diagrammatically, services are represented by boxes and their interfaces are represented by lines with black circles. The interface descriptions are in boxes with side bars and a top label "<<desc>>". Data is represented with a curved square shape.

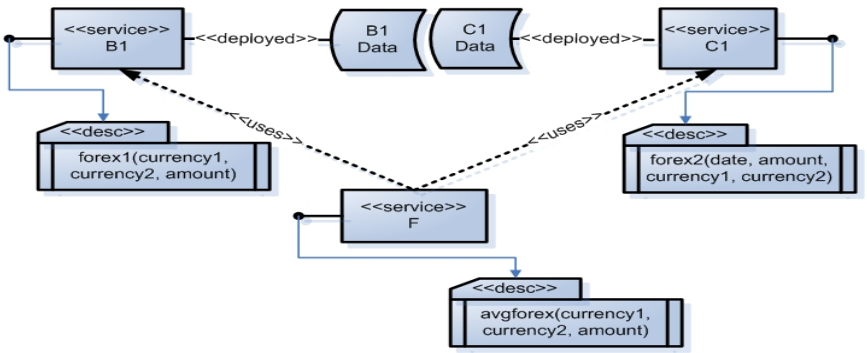

Figure 2. Service Composition Illustrated with Service Licenses

Consider a bank ***B*** that facilitates buying and selling of foreign currencies. ***B*** offers a free service, say ***B1***, for the consumers to quickly and easily calculate foreign exchange (forex) conversion based on daily rates offered by ***B***. Assume that ***B1*** provides the daily forex rates service for free and allows to be composed. This can be represented by a service license for ***B1*** as follows:

```
<o-ex:offer>
...
   <o-ex:permission>
      <sl:composition>
   </o-ex:permission>
   <o-ex:requirement>
      <o-dd:attribution/>
   </o-ex:requirement>
...
</o-ex:offer>
```

We also use another service ***C1*** similar to ***B1*** in functionality but offered by a bank ***C***. Assume that the service ***C1*** allows derivation and requires attribution. Then, the license of ***C1*** will be as follows:

```
<o-ex:offer>
...
   <o-ex:permission>
      <sl:derivation>
   </o-ex:permission>
   <o-ex:requirement>
      <o-dd:attribution/>
   </o-ex:requirement>
...
</o-ex:offer>
```

In our scenario, ***B*** also keeps a historical data records on forex, accessible via another service, say ***B2***. However, the composite service is restricted to access the recorded data of ***B2***. Assume that a new financial service, say ***F***, composes ***B1*** and ***C1*** in order to give average forex rates. Here, ***F*** is allowed to use ***B1***'s current daily forex rates to provide the average rate at the time of request. In this case, if F wants to access the recorded data of ***B2***, there could be some charges. Even here also, the ownership of data remains with ***B2*** and ***F*** could access the data. The license of the composite service ***F*** should be compatible with licenses of ***B1*** and ***C1***. When a service composes with other services, there is a possibility of having several compatible service licenses for a composite service. ***F*** can have the following license (one of the licenses in the compatible set of licenses), compatible with the licenses of ***B1*** and ***C1***.

```
<o-ex:offer>
...
   <o-ex:permission>
      <sl:derivation>
   </o-ex:permission>
   <o-ex:requirement>
      <o-dd:attribution/>
   </o-ex:requirement>
...
</o-ex:offer>
```

6. Related Work and Discussion

To perform any business, some form of understanding is required between a producer (seller) and a consumer (buyer). This mutual understanding about business transactions is established by several approaches across various application domains. In the field of services today, service level agreements (SLA) and policies are the common approaches for specifying this mutual understanding. Although a SLA is rather different from a license, they both regulate the activities of services. A basic difference is the fact that a SLA involves negotiations between a consumer and a provider, while a license is a unilateral statement, specified by the provider to one or more

consumers, without involving negotiations. Policies and licenses are similar in that they govern what a service does, but are not the same. Policies provide the means for specifying and modulating the behaviour of a feature to align its capabilities and constraints with the requirements of its users (Kamoda et al., 2005) whereas service licenses reflect the rights of providers to control how the service is distributed.

Though there are examples of service licenses in practical use (by Amazon, Google, Yahoo!), to the best of our knowledge, there appears to be no conceptualization of service licensing in general and the license terms given by these industries are not in machine interpretable form. Furthermore, these license terms simply represent the software dimension of services and do not focus service aspects. The business and legal contractual information are not described at a detailed level by the services research community, either in industry or academia. Though the design of service licenses could be an initiative of the software industry, there is no active involvement in this topic by industry. One of the primary causes for this could be fear still faced by industries over the lack of standardization of technologies surrounding SOC (Papazoglou et al., 2006).

Of the current service description standards and languages, WSLA (Ludwig et al., 2003) and WSOL (Tosic et al., 2003) are optimized for describing SLAs of web services. SLAng covers broader range of SLAs beyond web services. WS-Policy (Vedamuthu et al., 2007) and WSPL (Anderson, 2004) describe policy specifications for web services. Business contracts are detailed in ebXML CPP/CPA (OASIS, 2005). However, none of them describe distribution aspects and ownership clauses of licensing. To the best of our knowledge, the business and legal contractual information are not focused in detailed level by the services research community.

7. Concluding Remarks

As the nature of services differs significantly from traditional software and components, services prevent the direct adoption of software and component licenses. We have conceptualized the clauses of a service license, by detailed analysis of several service usage terms. Following the conceptualization, we have proposed a formalization of licensing clauses for unambiguous definition of a service license. Furthermore, we have proposed a language for expressing service licensing clauses that services can interpret automatically. We are working towards providing a foundation for an interoperable mechanism for selection and composition of services, yet legally enforceable.

Acknowledgement

This work was partially supported by funding from the Information Society Technologies Programme under the 7th Research Framework Programme of the European Union, within the research project COMPAS (Compliance-driven Models, Languages, and Architectures for Services). Call: FP7-ICT-2007-1 Grant agreement no.: 215175

References

1. Alonso, G., Casati, F., Kuno, H., & Machiraju, V. (2004) Web Services Concepts, Architectures, and Applications. Springer Verlag.
2. Anderson , A. (2004). Web Services Policy Language (WSPL). In Proc. of the IEEE Intl. Workshop on Policies for Distributed Systems and Networks (p. 189- 192).
3. Canfora, G. (2004). Software Evolution in the Era of Software Services. In Proc. of the 7th Intl. Workshop on Principles of Software Evolution (p. 9-18).
4. Chavez, A., Tornabene, C., & Wiederhold, G. (1998). Software Component Licensing: A Primer. IEEE Software, 15 (5), 47-53.
5. Classen, W. (1996). Fundamentals of Software Licensing. IDEA: The Journal of Law and Technology, 37 (1).
6. D'Andrea, V., & Gangadharan, G. R. (2006). Licensing Services: An "Open" Perspective. In Open Source Systems (IFIP Working Group 2.13 Foundation Conference on Open Source Software), Vol. 203, Springer Verlag (p. 143-154).
7. Dustdar. S. & Schreiner. W. (2005) A Survey on Web Services Composition. International Journal of Web and Grid Services, 1(1), 1–30.
8. Gangadharan. G.R. (2008) Service Licensing. Ph.D. Thesis, University of Trento, Trento, Italy.
9. Gangadharan, G. R., Frankova, G., & D'Andrea, V. (2007). Service License Life Cycle. In Proc. of the Intl. Symp. on Collaborative Technologies and Systems (CTS'07).
10. Goode, R. (2006). Commercial Law. Penguin Books Limited.
11. Gunther, O., Tamm, G., & Leymann, F. (2006). Pricing Web Services. In Dagstuhl Seminar Proceedings 06291.

12. 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 Proc. of the 14th Intl. World Wide Web Conference (WWW).
13. Lessig, L. (2004). The Creative Commons. Montana Law Review, 1-13.
14. Ludwig, H., Keller, A., Dan, A., King, R., & Franck, R. (2003). Web Service Level Agreement (WSLA) Language Specification. IBM Coporation.
15. OASIS ebXML CPP and CPA Technical Committee. (2005). Collaboration Protocol Profile and Agreement Specification Version 2.1. http://www.oasis-open.org/committees/ebxml-cppa/.
16. Papazoglou, M., Traverso, P., Dustdar, S., & Leymann, F. (2006). Service-oriented Computing: State of the Art and Research Challenges. IEEE Computer, 40 (11).
17. Robertson, R. (1990). Legal Protection of Computer Software. Longman Practitioner Series.
18. Rosen, L. (2004). Open Source Licensing: Software Freedom and Intellectual Property Law . Prentice Hall.
19. Skene, J., Lamanna, D., & Emmerich, W. (2004). Precise Service Level Agreements. In Proc. of 26th Intl. Conf. on Software Engineering (ICSE).
20. Stallman, R. (1999). Open Sources: Voices from the Open Source Revolution. In M. S. Chris DiBona Sam Ockman (Ed.), (chap. The GNU Operating System and the Free Software Movement). O'Reilly.
21. Tosic, V., Ma, W., Pagurek, B., & Esfandiari, B. (2003). On the Dynamic Manipulation of Classes of Service for XML Web Services. In Proceedings of the HP open view university association (hp-ovua) workshop.
22. Vedamuthu, A., Orchard, D., Hirsch, F., Hondo, M., Yendluri, P., Boubez, T., et al. (2007). Web Services Policy Framework. (http://www.w3.org/TR/ws-policy)
23. Weerawarana, S., Curbera, F., Leymann, F., Storey, T., & Ferguson, D. (2005). Web Services Platform Architecture : SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More. Prentice Hall PTR.
24. World Intellectual Property Organization. (2004). Successful Technology Licensing. WIPO Publishers, Geneva, Switzerland.