Lifecycle for Management of E-contracts Based on Web Service

José B. Neto, and Celso M. Hirata

***Abstract*—Electronic contracts (e-contracts) create obligations, permissions, and prohibitions for the parties in the Internet. However, the support to interpret and enforce the clauses related to contracts is limited in current implementations. Besides, e-contracts must follow the continuous change of business rules. The service provider must make available the services that enable the parties to satisfy obligations and respect permissions and prohibitions during the lifecycle of the contract. This task requires a considerable effort from the providers. We propose a lifecycle aid the management e-contracts. The lifecycle aims to allow automation of configuration of Web Service in order to facilitate the implementation work of providers. The six phases of lifecycle can be employed in Business-to-business (B2B) and Business-to-Consumer (B2C) contracts. To validate the approach, we employed the lifecycle in an implementation that uses the REST style to build, get and set the e-contract.**

***Index Terms*—E-contract, lifecycle, RESTful, Web Service.**

I. Introduction

BUSINESS in Internet has increased and has required a more complex specification of obligations, permissions, and prohibitions of parties.

The contract is a legal act of responsibility signed by two or more parts, a business that defines obligations and rights for all while having validity [1]. The elements of a contract are offer, acceptance, intention to create legal relations, and considerations. E-contract is an electronic version of the conventional contract that specifies activities and rules for business processes over the Internet.

The e-contract consists of representing organizations involved in a business process, activities, representing the electronic services, and contract terms, describing the constraints to be satisfied during the execution of the contract [2]. Contractual clauses can represent three different types of constraints: obligations that specify what the parties must do, permissions that specify what parties can do, and prohibitions, specify that the parties cannot do [3].

In this paper, we consider that e-contract is an agreement that uses Internet as a place to create and perform the obligations, permissions, and prohibitions (related to offer and acceptance) that are implemented through Web Service.

The services can be a simple acceptance through a web form or an automated process that implements all the activities related to the contract. The degree of automation depends on the employed technology.

The parties usually intend to reach the complete performance of the contract. While a contract is being performed, it is called an executions contract, and when it is completed, it is an executed contract. In some cases, there may be substantial performance but not complete performance, which allows the parties to be partially compensated.

Contracts in Internet are becoming complex. The complexity stems from the peculiarities of contractual terms that must satisfy the parties who have different requirements and restrictions. They include aspects of time, product or service functionality, quality, values, forms of payment, quantity, and others. The growing number of parameters to be considered in electronic transactions have discouraged agreements with complex implementations [4].

The parameters agreed in contracts have impact directly in the electronic services provided. For each agreement, it is necessary to adjust the services and reconfigure the Web Service. Another problem is the difficulty to configure Web Service engine once the contract has already been established. Several reasons lead to reconfiguration of the service: a cyber-attack, malfunction of system or modification of service [5].

Thus, the lifecycle is necessary to support the contract phases for creation, execution, completion, and in some cases compensation.

We propose a lifecycle of e-contracts, aiming to provide automation of configuration of Web Service in order to ease the work of the provider. The six phases of lifecycle can be employed for Business-to-Business (B2B) and Business-to-Consumer (B2C) contracts. To management the lifecycle, we use a framework that works as a Broker on a service-level agreement (SLA) architecture[6].

The difference of this paper to previous publications is the definition of contract management at the level provided by Web Service.

The rest of this paper is structured as follows. Section 2 provides some background and related work about e-contract lifecycle. In Section 3, we propose the lifecycle for e-contracts. We explain conceptual details about each phase. We pay particular attention to the phases that have activities that are subject to automation of configuration of Web Service. In Section 4, we illustrate the requirements to enforce the e-contract lifecycle, and then we use an example of emphasizing the automation of activities. Section 5 describes the analysis and conclusions of the proposal.

II. Background and Related Work

A. Web Service transactions

XML Web Service use Extensible Markup Language (XML) messages that follow the Simple Object Access Protocol (SOAP) standard. In such systems, there is often a machine-readable description of the operations offered by the service written in the Web Service Description Language (WSDL). They are W3C standard distributed in different

Manuscript received July 22, 2013; revised October 7, 2013.

J. Bernardo and C. Hirata are from Technological Institute of Aeronautics, Department of Computer Science, São José dos Campos, SP, Brazil. Contact: jneto@c4i.gmu.edu, hirata@ita.br.

layers to exchange data in SOAP envelope, WS-*/SOAP. For communication, the provider publishes the services that can be consumed by other Web Service, in WSDL through a valid Universal Description, Discovery, and Integration (UDDI). In UDDI, there are different areas for publications: the pages for business, pages to describe how to access services, and pages to provide a classification of service or business, based on the standard taxonomies. After selecting a service, the Web Service client receives WSDL file that describes how to use the service using SOAP protocol [7].

Another major class of Web Service is the REST-Compliant Web Service. Representational State Transfer (REST) is an architectural style that uses the structure of the web to transport the data. It is considered a lightweight protocol. REST is a collection of resources that are defined by a Tuple: (1) Metadata are the semantic data attached to the resource; (2) URI is a unified resource identifier for identifying the resource within the RESTful Web Service and (3) Method based on HTTP (GET, PUT, POST and DELETE) [8][9]. Services complying with the concepts of REST are often called RESTful, and the most popular language for RESTful services is WADL. REST has good adoption due to its simplicity [10]. Its main limitations are: (1) not adequate pattern of security and (2) difficulty to manipulate complex transactions.

We use the REST architectural style with CRUD (creation, retrieval, update, and deletion) methods to build, get and set the e-contract in our implementation.

B. WS-Agreement

The WS-*/SOAP has many layers to manage the whole protocol as WS-Security, WS-Policy, WS-Federation, WS-Trust and others. WS-Agreement is one layer, not standardized yet by W3C, which encapsulates the Metadata associated with an agreement. It is intended for the contract negotiation, and there are fields to record the name of the involved parties, context, terms of service, warranty terms and restrictions of the agreement that can be considered service contract. It employs many keywords as MUST, REQUIRED, SHALL, SHOULD that are used inside the clauses of the contract. An XML document represents a contractual obligation between an agreement provider and an agreement initiator [11].

Frankova et al. [12] increase five states in the traditional WS-Agreement lifecycle. The goal is the reduction of cancellations for violations in the negotiations.

C. Related Work

The literature about e-contract lifecycle is scarce. Morciniec et al. [13] suggest three phases lifecycle of e-contract: drafting, formation, and execution. In the contract drafting phase, an instance of the template is specified. The template typically has a number of free variables that are agreed for the next phase. In the contract formation phase, the parties fix and negotiate the variables (requirements and restrictions) of the contract (deadlines, order of actions). In the contract execution phase corresponds to the activities being performed under the contract. Typically this phase consists of service or goods delivery, invoicing, bill calculation, presentment and payment.

Xiao et al. [14] propose a lifecycle for e-contract divided into three phases distributed in eight processes with a focus on security, named Electronic Contract Record Center (ECRC). They present a method based in the template of electronic contract. A part of the lifecycle is automatic and other manual.

Chiu et al. [15] describe a lifecycle with three phases. The emphasis is on business rules. The rules are specified in XML descriptions for identification of obligations, permissions and prohibitions. As the basis for contract, the work uses the language Event Condition Action (ECA).

Shu et al. [16] propose a lifecycle for Service-Level Agreement (SLA) in which the client can negotiate the level of the service provider with reference a published library.

The above approaches address many aspects of the e-contract lifecycle, but there are still some gaps that need to be addressed. More specifically, they do not manage the specifications for IT environment. The lifecycle for e-contract, described in the next section, follows the requirements of Web Service's transactions and allows to build the e-contract through exchange small parameters.

In the next section, we describe the proposal of e-contract lifecycle and its phases.

III. E-CONTRACT LIFECYCLE

The lifecycle must include phases and activities that address all the needs of parties with respect to contracts using Internet as a mediator. The needs typically involve elaboration of the draft contract, offer (some form of publication of the draft), negotiation (activities to reach an acceptable set of terms by the parties), acceptance (some act that represents acceptance, such as signature), execution of the contract, and closure. The lifecycle is structured into six sequential phases from proposal to closure. For each phase, there is at least one product to finish the phase before starting the next phase. The order must be kept due to the dependencies between the activities of the phases in terms of products. The activities were defined based on the standard in transaction rules for communication between Web Services: publishing of services, search and message exchange. The main activities of the lifecycle are the same for B2B or B2C contracts although the operation phase is used for different goal. There is a product that identifies the end of phase. The phases are (1) Proposal, (2) Configuration, (3) Publication, (4) Negotiation, (5) Operation, and (6) Closure. The contract goes through a series of activities that are organized in phases. The activities have inputs and at least one output. Figure 1 illustrates the lifecycle. The dotted line from the closure phase indicates the restart of the proposal phase. The dotted line, from negotiation to proposal, represents changes or adaptations of the e-contract clauses in the negotiation phase.

In the following, we describe the phases. As indicated, each phase has activities. There are logical sequences of activities to provide a correct flow. Each user executes activity according to their roles. Roles shall be assigned to users according to some access model, such as RBAC [17]. We emphasize that the definition of the activities gives support to the work of the parties.

ISBN: 978-988-19252-3-7 (revised on 28 October 2013)
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

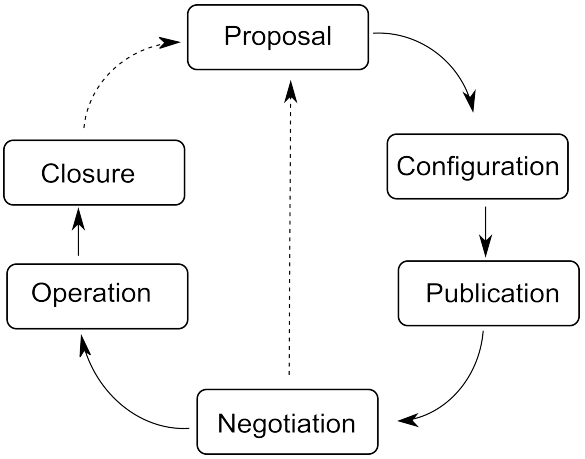

Fig. 1. Six phases of Electronic Contract's lifecycle.

A. Proposal

The reasons for proposing a contract may include a continuous intent of the provider to make available an initial set of terms to form a contract. Another reason is when a client wishes to have specific business terms. The initial set of terms in general has an established pattern that fits within transactional rules for Web Service. The set of terms with rules is the draft contract. Draft contract is a template for contracts. The draft contract is available for download from the server, named here as Broker. The Broker plays the role of manager of contracts. The Broker is used for verification and publication of services related to contracts.

This phase aims to produce a pre-contract as output.

The pre-contract is complete when it is signed. It is desirable that at least the provider agrees in all the terms of the pre-contract. Nonetheless, some of them can be left open for later definition. Restrictions can be set beforehand. The restrictions refer to what is acceptable by the provider. The terms can be negotiated later. It is desirable that they can be configured into services automatically.

As the output of this phase, the pre-contract ready to use in the configuration phase.

B. Configuration

The requirement for this phase is the pre-contract signed by the proponent part.In this phase, the Web Services related to the implementation of pre-contracts are developed and deployed. The services include transaction and supporting for web services. The transaction web services must be organized in some protocol in order to assure the desirable properties. Properties may include atomicity, consistency, isolation, and durability.

We aim to have automation of this phase. The idea is to use the pre-contract as input and generate automatically as the output, the services deployed and running. With the automation, the providers will be able to have fast implementation and publication of their services.

The reason for having the configuration before negotiation is to assure that the pre-contract can be implemented and deployed.

This anticipation allows to start the negotiation phase with a pre-contract that is implementable.

There is a set of fields, in the pre-contract, that aims the configuration.

The responsible for this phase is the Web Service proposing the pre-contract. The output of this phase is the machine ready to provide the service related to in the pre-contract.

C. Publication

As a requirement for this phase, provider must have completed the configuration services related to the pre-contract. In this phase, the process of permissions to files is carried out. The Broker is responsible for the process. The publication of the contract follows the standardized format for publishing services WS by adapting the contract to the publication format. The responsible for the publication of pre-contract is the service provider and Broker. The goal of the publication phase is a pre-contract available for download in the Broker.

D. Negotiation

With the pre-contract published on Broker, the parties can start the negotiation phase. In the negotiation, the parties make adaptations of the contract. This phase refers to exchange of messages aiming to fix the conditions. It is a decision process where two or more parties interact so that everyone win [18].

There are two possible paths from this phase. The first path corresponds the acceptance of the pre-contract without changing of the services already developed. In the first path, small changes can be accepted by the current services. The service provider must identify the need of service adaptation for the second path. If the parties reach an agreement, in the second path, the adapted pre-contract is used to configure and generate the Web Services. The negotiation phase ends with the contract signed by the parties or the parties give up the negotiation. If the deal is closed, the contract is closed and the permission to access it is read only.

The responsible roles for this phase are Web Service providers, clients involved in contract negotiations, and Broker.

As output stage negotiation phase is a contract signed by both parties, with permission to read and stored in the Broker and parts.

E. Operation

The negotiation starts with the receipt of the contract signed by the parties. The Broker is the authority that guarantee the valid the signatures. The requirement of this phase is both parties with the e-contract. In the operation phase, the services are used within protocols. The protocols are used to assure transaction properties. The services correspond mainly to orders, deliveries, and payments. In the operation phase, the parties make the effort to fulfill the obligations of the contract. The operation phase ends when all the required services were carried out (or the obligations are met) or when the contract ends.

F. Closure

This phase starts when the deadline defined in the contract is reached or all the obligations were met, and the parties

ISBN: 978-988-19252-3-7 (revised on 28 October 2013)
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE I
LIFECYCLE OF E-CONTRACT

Phase	Input	Output
Proposal	template e-contract	draft contract
Configuration	pre-contract	configuration of services
Publication	local available to publish	pre-contract URL
Negotiation	pre-contract published	e-contract signed
Operation	e-contract	orders received, services and goods delivered, payments made
Closure	deadline reached or obligations met	acceptance of the end of contract by the parties or legal dispute

do not intend to continue with the contract. It can be shortened if one of the parties wishes. In this case, there are compensations to be made.

The responsible roles of this phase are all the parties, including Broker.

This phase successfully ends with the understanding by the parties that the contract is over, and no obligation is due. The phase may also end with the initiation of a legal dispute by at least one of the parties. In the second case, all the transaction information should be kept and saved.

IV. Enforcement of E-contract lifecycle

In this section, we discuss the requirements for enforcement of e-contract lifecycle. Then we introduce layers that aim adapt the e-contract to XML language in Web Service format. After that, we formalize the structure of the e-contract in some formal definition. The goal of the formal definition of contract is to allow the automatic configuration of the contract into services present a practical example which uses the formal definition and allows illustrate the automation.

A. Definition of E-contract

This formalization aims to facilitate the understanding of the internal structure of the e-contract used during the lifecycle phases.

Definition 1 (Term): Term t is a couple (p,ω) with $p \in P$, e $\omega \in \Sigma$ where P is the set of n paragraphs e Σ is the set alphabet used in agreements. T $\subseteq P$ X Σ is the set of Terms T.

Definition 2 (Clause): Clause is a couple (t,d) with $d \in D$, e $t \in T$ where D is a set $\{\texttt{SHALL}, \texttt{MUST}, \texttt{SHOULD}, \texttt{REQUIRED}, \texttt{NOT}\}$ e T is a set of terms. C $\subseteq D$ X T is a set of Clauses C.

Definition 3 (Contract): is a tuple $\langle S,C,R \rangle$ with S set of m services, C set of n clauses and R set of n resources associated.

Definition 4 (External state): St_{out} of the contract is an element of the set $\{\texttt{proposal}, \texttt{configuration}, \texttt{publication}, \texttt{negotiation}, \texttt{operation}, \texttt{closure}\}$.

Definition 5 (Internal state): St_{in} is a couple (α,β) where β=$\{\texttt{valid}, \texttt{invalid}\}$ e α=$\{\texttt{ready}, \texttt{modifying}\}$.

Following formal specification adapted is necessary to adjust the language of the contract that is exchanged XML Web Service.

The definitions aims to facilitate the understanding of the electronic contract computer model and identify the activities and processes which formally constitute its structure.

B. Requirements for E-contract Enforcement

The framework built aims the validation of the proposal. The model used to validate has three layers: (1) document, (2) business, and (3) service [15].

The document layer is composed of various terms that form the clauses of the contract. This layer is responsible for the consolidation of the e-contract template. Each clause must identify resource and parties in the agreement.

The second layer is called business. It handles specific events and business rules referenced in the contract. Permissions and signature are set in this layer. The business layer is also responsible for the interface between the document layer and the service layer.

Service Layer allows to operate the digital contract in the Web Service's environment. This layer binds the business layer through the configuration of the Web Services. When the contract is completed, the Web Service configuration is associated with the parameters of the contract. Modify the contract setting beyond the automatic configuration of engines implies fee and penalties fixed in clauses.

The structure illustrated in Figure 2 works with three layers. The permissions are used in the Data Library to access the clauses in different phases of lifecycle. It follows native RESTful methods. The HTTP responses 200 or 204 represent successful requests. The responses 400, 401, 403, 404, and 410 mean exceptions in the request. The Template Manager saves the drafts in the database and allows Web Service access with permission. Methods idempotent (PUT, POST or DELETE) first check if the St_{in} is equal (valid, modifying). In this case, it not permitted to change the parameter associated. The module E-contract Negotiate works in the negotiation phase to support the conditions. If this phase finishes well, enables the operation phase in Web Service Manager, else the pre-contract will be cancel.

The database records the history about contracts and supports the negotiations. The Template Manager uses an empty contract saved in the database. The Web Service Manager is responsible for the connection between client and providers. The broken line is the operation phase, the client sends and receives messages directly from providers.

C. XMl syntax for e-contract

We adopt the specification the template e-contract based on B2B contract [19]. The resources exchanged via RESTful may not express the correct functionality of the agreed contract. The challenge was to design a template with resources with semantics associated. Modify small parameters during the negotiation keeps the meaning of the clauses.[20]

In the survey, aiming at the development of the template for small e-contract, the most popular e-contracts presented in the Internet were: ticket for flight, car rentals, hotel accommodation, web hosting, tickets for show, product sales,

ISBN: 978-988-19252-3-7 (revised on 28 October 2013)
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

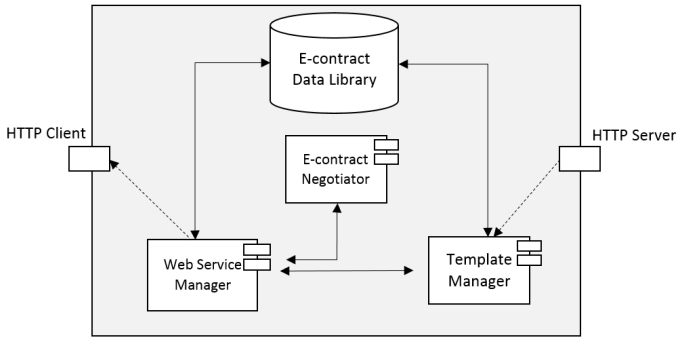

Fig. 2. Broker architecture for supporting the lifecycle

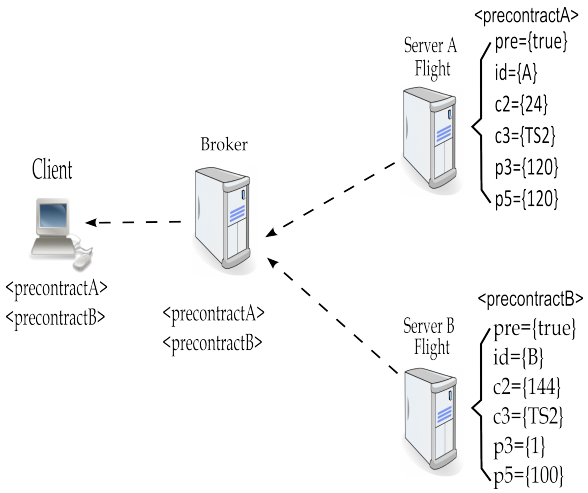

Fig. 3. Proposal, Configuration and Publication phases

and acquisition of services. This model based in those examples and the WS-Agreement structure. For this research, the more important fields to B2C contracts used are features of product, value, quantity, delivery address, payment type, validity of the contract. For B2B contracts: deadline, price, and quality of service. The e-contract follows WADL structure for RESTful server. Table II presents the parameters.

TABLE II
A BASIC E-CONTRACT

Resource	Property	URI
identify	identifier id client ip	/idOwner /idClient /ip
payment	amount idpayment type value	/amount /idPayment /type /value
configuration	configuration qos protocol timeout	/configuration /qos /protocol /timeout
product	description datepurchase warranty trackorder	/description /datePurchase /warranty /trackOrder

D. Example

To illustrate the lifecycle we describe an example of purchasing airline tickets. We use a client, simple web hosting, to demonstrate the possibilities of the e-contract management. By using the browser with REST Client application, the client can change the fields of pre-contract until completion of the e-contract.

In this example, the client chooses the best condition between two providers. The customer wants to purchase tickets for a group of friends to travel with their families. The most of the fields of e-contract can be used as parameter of the negotiation. Each field is considered a dimension. In this negotiation, we work with two dimensions: price, and deadline.

In figure 3, we illustrate the situation where the first three phases of the lifecycle were executed. Servers A and B have their pre-contracts configured and published. The Web Service providers download the template from Broker or set the pre-contract with RESTful methods. After that, they do the configuration of the engines and then send the pre-contract to Broker. The code example for server A has its pre-contract in XML stored in a Broker URI, http://192.168.1.1/serverA/contractA. Each letter represents one resource in the e-contract area. (1) *i*, identification, (2)*c*, configuration (3) *p*, payment. The description of the parameters are *p3*, maximum time for payment in days and *p5* the minimum price per unit. The configuration parameter *c2* represents the time to save a reservation in hours. The item configuration *c3=TS2* means the protocol Timestamp-based Two Phase Commit Protocol for RESTful Services. Each parameter has its own path. The path, http://192.168.1.1/serverA/contractA/p1/a, indicates that resource *p1* value '*a*'. The parameter *pre* equals to 'true' means that it is a pre-contract signed by provider. After the agreement is signed by all parties, the parameter *pre* is changed to '*false*'. In this moment, the e-contract is set to be used. For security, we use the parameter '*isolate*' assigned with '*false*' that indicates that the pre-contract cannot be modified, only with access the publication key, < resource=p1 value=a isolate="true" >. The tag that controls the access disables modify and allows read only [21].

Figure 4 shows the publication of the pre-contracts. The advantage of the server A is the deadline to finish the payment in 120 days. The server B has the best price, $100 per ticket, and keeps the reservation 6 days or 144 hours.

The conditions of negotiation are included in the pre-contract of the proposer. The other parties have no permission to read before the negotiation is concluded. The service A hides the condition: more than 50 tickets implies pay 90 per ticket. This rule represents 25% of discount.

In the negotiation phase, the client queries the Broker and reads the pre-contracts. Afterwards, the client negotiates the parameters, increasing or decreasing values from resources on the server that suits the client.

The client sends messages related to both pre-contracts to adjust the parameters. In the negotiation phase, if the client receives ok, it indicates that the change value of the parameter was accepted. If out of the conditions, the answer is negative or "bad request". Other responses can occur as results of requests from RESTful Methods.

In Figure 4, the client tries to modify the price in the server A and the maximum time of payment in the server B. As a result of its action, the client receives the positive response, 'ok' (HTTP 200) from A and negative, 'bad request' (HTTP 404), from B. Client A uses PUT method with signature (A,c2,90,55). It represents the request to set *c2* to 90 and

ISBN: 978-988-19252-3-7 (revised on 28 October 2013)
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

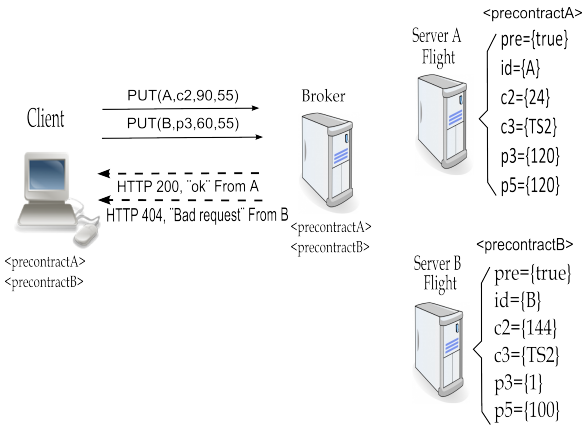

Fig. 4. Negotiation phase, client tries to modify two parameters

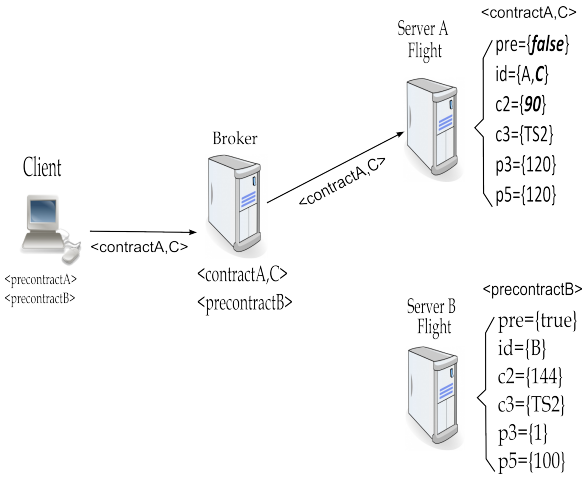

Fig. 5. Web Service are ready to Operation phase

buy 55 tickets from pre-contract A. The client decides to sign the contract with server A. The client sends the contract to Broker after signing. The Broker authenticates the signature and records the contract during the validation (operation phase). After that, it sends a copy to both parties. The parameter *pre* is set to '*false*'. The parameters in bold were modified (Fig. 5). During the operation phase, the client must connect the Server A to buy the tickets and set the date and time of the flights.

V. Conclusion and Remarks

The proposal aims at automatic configuration of the engines of the Web Service through contracts. The configuration space (set of fields) in the e-contract allows to connect the services provided by the Web Service and agreement signed.

The lifecycle serves as the basis for construction of contracts in format manipulable by Web Service and web hosting. The implementation helps the work of the client and the Web Service providers because it allows managing contracts in a dynamic way, adapting to change of business rules and helps the automatic configuration of the engine and deployment of the Web Services can be applied WS-*/SOAP, and support both types e-commerce negotiation, B2B and B2C.

As future work we intend to increase the functionality of the lifecycle. Build the composition of contracts by using pre-contracts published.

References

[1] K. V. Indukuri and P. R. Krishna, "Mining e-contract documents to classify clauses," in *Proceedings of the Third Annual ACM Bangalore Conference*, ser. COMPUTE '10. New York, NY, USA: ACM, 2010, pp. 7:1–7:5. [Online]. Available: http://doi.acm.org/10.1145/1754288.1754295

[2] A. S., *RESTful Web Services Cookbook*, 3rd ed. O'Reilly, 2010.

[3] S. Angelov, S. Till, and P. Grefen, "Dynamic and secure b2b e-contract update management," in *Proceedings of the 6th ACM conference on Electronic commerce*, ser. EC '05. New York, NY, USA: ACM, 2005, pp. 19–28. [Online]. Available: http://doi.acm.org/10.1145/1064009.1064012

[4] F. G. Marchione, M. Fantinato, M. B. F. de Toledo, and I. M. S. Gimenes, "Price definition in the establishment of electronic contracts for web services," in *Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services*, ser. iiWAS '09. New York, NY, USA: ACM, 2009, pp. 217–224. [Online]. Available: http://doi.acm.org/10.1145/1806338.1806382

[5] W. Tsai, W. Song, R. Paul, Z. Cao, and H. Huang, "Services-oriented dynamic reconfiguration framework for dependable distributed computing," in *Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International*, 2004, pp. 554–559 vol.1.

[6] Z. Shu and S. Meina, "An architecture design of life cycle based sla management," in *Advanced Communication Technology (ICACT), 2010 The 12th International Conference on*, vol. 2, 2010, pp. 1351–1355.

[7] W3C, "Service description of web services." [Online]. Available: http://www.w3.org/standards/webofservices/description

[8] X. Zhao, E. Liu, and G. Clapworthy, "A two-stage restful web service composition method based on linear logic," in *Web Services (ECOWS), 2011 Ninth IEEE European Conference on*, 2011, pp. 39–46.

[9] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Doctor of Information and Computer Science, University of California, Irvine, 2000.

[10] I. Porres and I. Rauf, "Modeling behavioral restful web service interfaces in uml," in *Proceedings of the 2011 ACM Symposium on Applied Computing*, ser. SAC '11. New York, NY, USA: ACM, 2011, pp. 1598–1605. [Online]. Available: http://doi.acm.org/10.1145/1982185.1982521

[11] G. R. A. A. P. G. WG, "Ws-agreement specification (ws-agreement)." [Online]. Available: http://www.mcs.anl.gov/ keahey/Meetings/GRAAP/WS-Agreement20Structure.pdf

[12] G. Frankova, D. Malfatti, and M. Aiello, "Semantics and extensions of ws-agreement," *Journal of Software*, vol. 1, no. 1, 2006. [Online]. Available: http://ojs.academypublisher.com/index.php/jsw/article/view/01012331

[13] [Online]. Available: http://www.w3.org/2001/03/WSWS-popa/paper30

[14] X. Yu, Y. Chai, and Y. Liu, "A secure model for electronic contract enactment, monitoring and management," in *Electronic Commerce and Security, 2009. ISECS '09. Second International Symposium on*, vol. 1, 2009, pp. 296–300.

[15] D. Chiu, S. Cheung, and S. Till, "A three-layer architecture for e-contract enforcement in an e-service environment," in *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, 2003, pp. 10 pp.–.

[16] Z. Shu and S. Meina, "An architecture design of life cycle based sla management," in *Advanced Communication Technology (ICACT), 2010 The 12th International Conference on*, vol. 2, 2010, pp. 1351–1355.

[17] W. Zhou and C. Meinel, "Team and task based rbac access control model," in *Network Operations and Management Symposium, 2007. LANOMS 2007. Latin American*, 2007, pp. 84–94.

[18] "Fgv," 2012. [Online]. Available: http://www.contratos.net.br/

[19] A. Goodchild, C. Herring, and Z. Milosevic, "Business contracts for b2b," in *Proceedings of the CAISE*00 Workshop on Infrastructure for Dynamic Business-to-Business Service Outsourcing*, 2000, pp. 5–6.

[20] S. Mayer and D. Guinard, "An extensible discovery service for smart things," in *Proceedings of the Second International Workshop on Web of Things*, ser. WoT '11. New York, NY, USA: ACM, 2011, pp. 7:1–7:6. [Online]. Available: http://doi.acm.org/10.1145/1993966.1993976

[21] C. Pautasso, "Restful web service composition with bpel for rest," *Data Knowl. Eng.*, vol. 68, no. 9, pp. 851–866, Sep. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.datak.2009.02.016

1) Date of modification: October 28th 2013

2) Brief descriptions of the changes:
Figure 2: from ”Labrary” to “Library”
Figures 4 and 5: some parameters and graphs were modified to be coherent with the text explanation.

ISBN: 978-988-19252-3-7 (revised on 28 October 2013) WCECS 2013
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)