A FEATURE-BASED TOOLKIT FOR ELECTRONIC CONTRACT NEGOTIATION AND RENEGOTIATION

Daniel Avila Vecchiato*, Maria Beatriz Felgar de Toledo*, Marcelo Fantinato** and Itana Maria de Souza Gimenes***

*University of Campinas, Institute of Computing, Av. Albert Einstein, 1251, Campinas - SP, Brazil
**University of São Paulo, School of Arts, Sciences and Humanities, R. Arlindo Béttio, 1000, São Paulo - SP, Brazil
***University of Maringá, Department of Informatics, Av. Colombo, 5.790, Maringá - PR, Brazil

ABSTRACT

E-contracts usually describe inter-organizational business processes defining e-services to be provided and consumed as well as non-functional requirements such as Quality of Service (QoS). Organizations involved in a cooperation need to provide explicit guarantees in an e-contract, which can involve renegotiation of contractual clauses, penalty application or intervention in the process execution. In this paper, feature modeling is used to represent e-services, QoS attributes and control operations to be triggered when QoS attribute levels are not met. In addition, it describes FeatureContract, a toolkit to support contract establishment based on features. An application example is exploited to show the proposed approach feasibility. The main contributions are related to improved structure and reuse in information in e-contracts.

KEYWORDS


1. INTRODUCTION

The current Business Process Management (BPM) scenario includes: i) one or more organizations that provide and/or consume electronic services (e-services); ii) negotiation and establishment of electronic contracts (e-contracts), including quality of service (QoS) attributes and levels; iii) definition, enactment, monitoring, and auditing of business process; and, iv) process analysis and optimization. E-contracts between two or more partners interested in an inter-organizational business process establish the activities to be performed and the obligations, permissions and rights related to each involved party. During contract enactment, if a party is unable to fulfill contractual clauses, a contract renegotiation may be triggered.

Fantinato et al. (2010) have developed a new approach to reduce the complexity in the wide management of e-contracts for Web services (WS-Contract). The approach, based on Software Product Line (PL) concepts, is named PL4BPM (Product Line for Business Process Management). A predominant part of the PL4BPM approach is the WS-Contract establishment process, focused on providing improved information reuse and structuring based on the use of feature modeling (Fantinato et al. 2008).

This paper discusses the e-contract life cycle from negotiation, establishment, enactment to renegotiation within the context of a feature-based BPM infrastructure. The main contributions are: i) an extension of a contract metamodel to include, respectively, the control operations to be performed in case of e-contract violation and represent it by using the WS-Agreement specification; ii) an extended infrastructure to support e-contract negotiation and renegotiation, including the extension of the FeatureContract toolkit; iii) more efficient management, and reuse of information necessary for the establishment and renegotiation of e-contracts; and, iv) an example of the approach showing the various stages of an e-contract establishment.

The feature modeling technique used in the representation of e-services and QoS attributes has shown the following advantages (Fantinato et al. 2008): flexibility for e-services specification; modularization facilities for QoS attributes specification; and, structured representation of the optional and mandatory WS-contract elements. This may also be advantageous in contract renegotiations as we will see.

The paper is organized as follows: Section 2 presents basic concepts; Section 3 presents some related work; Section 4 presents the extended Feature and WS-Contract metamodels; Section 5 presents the
extension made on the FeatureContract toolkit, within the proposed BPM infrastructure; Section 6 presents an application example using the proposed approach; and, finally, Section 7 concludes the paper.

2. BACKGROUND

In this section, firstly e-contracts are described and then feature modeling concepts are presented as this is the technique used in this approach to represent information to be included in e-contracts.

2.1 E-contracts

Organizations interested in Internet business partnerships must define details of the business process to be enacted and express them in e-contracts (Fantinato et al. 2008). An e-contract defines details about the organizations, the activities to be executed and the contractual clauses that must be met during process enactment (Grefen et al. 2001). The clauses could be of three types: obligations, rights and prohibitions. The obligation clauses include QoS of e-services within the inter-organizational process. In addition to the functional aspect of e-contracts, there is also the legal aspect that will not be considered in this paper.

The activities to be contracted are e-services that may be implemented as Web services; when e-contracts are called WS-Contracts. Web Services have spread as a promising technology for the effective automation of inter-organizational interactions (Papazoglou 2007). The major benefit of it is the wide standardization including: a language to describe service interfaces (WSDL), a service directory structure and APIs for service publication and discovery (UDDI) and a communication protocol (SOAP).

The contractual clauses may be specified in languages such as WS-Agreement, a specification offering assurance about Web Services execution (Andrieux et al. 2007). It was developed to represent and establish agreements between two parts, usually a service provider and a consumer. The specification uses XML to structure an agreement that is composed by elements such as Name, Context, Terms (Service Terms and Guarantee Terms). WS-Agreement also allows the definition of service level objectives and qualifying conditions associated with importance, penalties and rewards representing the business value to the provider.

2.2 Feature Modeling

Feature modeling captures and manages common points and variabilities in software product lines (Pohl et al. 2005). A feature model represents properties of some entity of interest. It can denote any functional or non-functional property in the requirement, architectural or other levels. Features can be mandatory, optional or alternative (Czarnecki et al. 2005). They are organized into a tree-like diagram in which a node represents a feature and each feature can be described by a set of sub-features represented as descendant nodes. A feature model describes a system family. A family member can be configured by selecting the desired features from the model within the model variability limits, which is called feature model configuration.

3. RELATED WORK

The CrossFlow project is precursor in the area of inter-organizational business process (Grefen et al. 2001). As some other earlier works, they use metamodels and templates to facilitate e-contract establishment. More recently, Angelov and Grefen (2008b) defined an e-contract metamodel with different perspectives such as function and communication perspective. They identify four e-contract lifecycle phases: i) information that describes services and possible partners; ii) pre-contracting that customizes business operational aspects; iii) contracting that establishes how the business process will be carried out; and iv) enactment that comprises the execution of the contracted e-services. Angelov and Grefen (2008a) also defined a reference architecture to contract systems development, using a component-based approach. It provides a component for each phase of e-contracting (information, pre-contracting, contracting and enactment).

Bacarin et al. (2008) put forth a negotiation protocol with some primitive actions to assign property values, to send offers, request for proposal (RFP) and votes. They identify the following phases: negotiation
announcement, leader determination, objective announcement, negotiation setup, restriction announcement, core negotiation, commit attempt, contract (re)construction.

Some works (Angelov & Grefen 2008b; Bacarin et al. 2008; Grefen et al. 2001) use e-contract template to facilitate e-contract reuse. In a general way, the renegotiation issue is still not completely addressed in a proper way by the works proposed in the literature. Some architectures and frameworks (Angelov & Grefen 2008a; Bacarin et al. 2008) allows contract update during process execution. However, none of them specifies the actions to be performed in the case of contract violation. This work addresses the issue of control operations to handle contract enactment in the case of any clause violation.

4. FEATURE AND WS-CONTRACT METAMODELS

The BPM context involves business process composed by e-services. Our approach considers e-services implemented as Web Services and hence the WS-Contracts to regulate the collaboration between the parties. The WS-Contract is composed of: parties, e-services, contractual clauses and a business process. WS-BPEL is used to define the parties and the orchestration of the e-services within an inter-organizational context (Alves et al. 2007). E-services and QoS attributes are described in WSDL and WS-Agreement, respectively.

The WS-Contract metamodel, presented in Figure 1, represents the structure that supports the creation of e-contracts. Originally, it was proposed by Fantinato et al. (2008) and was extended (in gray color) to represent control operations. The metamodel comprises: i) Web Services described in the WSDL; ii) Business process specified in the WS-BPEL; iii) QoS attributes of Web Services, described in WS-Agreement; and, iv) Control operations to be handled in case of contract violation, also described in WS-Agreement.

A useful strategy explored here is the use of contract templates. An e-contract template is defined only once, but allows instantiation of distinct and similar contracts. To facilitate the creation of templates, Fantinato et al. (2008) proposed a feature metamodel. It originally consisted of two sub-trees – e-services and

![Figure 1. WS-Contract metamodel](image-url)
QoS-attributes, but it has been extended to include a Control Operations sub-tree as presented in Figure 2. The acronyms depicted in this figure are used to identify the feature type in the feature models, as in the Figure 4 presented later. In Figure 2, each sub-tree is described as follows:

- **E-Services sub-tree**: (mandatory) contains features representing the e-services offered by an involved organization;
- **QoS-attributes sub-tree**: (optional) contains features that represent the QoS attributes, which are attached to e-services defined into the e-Services sub-tree. It includes choices of QoS attribute levels;
- **Control-operations sub-tree**: (optional) it specifies control operations to be executed when QoS attribute levels are not met. These attributes are attached to e-services defined into the e-Services sub-tree and to QoS attributes defined into the QoS-attributes sub-tree. The Control Operation and Activity are depicted latter in Figure 3, each control operation have a possibility to be controlled or not by an activity. The element Value, in dark gray color, will only exist in the fine application activity and is used, to represent the fine value.

The Control Operations sub-tree can be associated directly to an e-service or to specific QoS attributes. The former is used as a default option whereas the latter is used as a specialization option. When a QoS attribute is not met, if there are control operations settings defined for it, they are triggered; otherwise, if there are control operations settings defined for the associated e-service, these control operations are triggered instead. With this feature structure support, a unique set of control operations options, defined only once, can be reused by all the QoS attributes and levels associated to all the e-services. During feature model configuration, specific control operations options can be selected for each QoS attribute or for each e-service.

Figure 3 shows the possible Control Operations and the mapping with WS-Agreement section, as follows:

- **Renegotiation**: used for contract update in three ways: i) **Clause** (QoS attribute) that can be removed, added or updated. It can be necessary if new requirements in the inter-organization cooperation appear; ii) **Variable** (QoS level) that can be renegotiated when triggering a penalty or control operations on a process are not necessary; and, iii) **Price** that a service or a QoS level price can be renegotiated. This can be applied in services that are not having QoS attribute levels as expected;
- **Penalty Application**: used to apply a penalty to the offending party. The penalty is a fine to compensate some eventual loss. It can be selected, for example, if the QoS attribute “availability” is not fulfilled causing loss of money or clients;
- **Process**: used to directly influence the business process execution. The available operations are: i) **Rollback** that undoes operations already executed by the business process. It can be selected, for example, for atomic e-services executed in a transactional way; ii) **Suspend** that stops the business process execution until some condition is reached. It can be selected, for example, for the QoS attribute “security”, since the process can be suspended until some security level is fulfilled; and, iii) **Termination** that terminates the business process execution. It can be selected, for example, when clauses related to important e-services can not be fulfilled and the process can not proceed.

In the right side of Figure 3, the Business Value List is composed as follows (Andrieux et al. 2007):

- **Penalty**: defines an expression to be assumed when an associated objective is not met;
- **Value Unit**: (type xs:string) defines the unit for assessing penalty. It is used, in this work, to represent a currency in case of penalty application or a control operation such as renegotiation or process;
- **Value Expr**: (type xs:any) represents a value in case of penalty application or a control activity such as variable, price or clause renegotiation;
- **Assessment Interval**: defines the interval over which a penalty is assessed. It can be one of the following: i) **Time Interval** that defines the assessment interval as a duration. For example, a weekly or monthly interval for defining the assessment; and, ii) **Count** that defines the assessment interval as a service specific count, such as the number of invocations.

When the Contract Renegotiation operation is chosen, a negotiation protocol must be specified. It will be performed after a notification sent by the monitor to the collaborating parties. Other operations such as Process Terminate, Process Rollback and Process Suspend will be executed by the WS-BPEL server. The monitor and WS-BPEL server are elements of the infrastructure described in Section 5.

![Figure 3. Mapping between control operation elements regarding both metamodels](image)

**5. BPM INFRASTRUCTURE AND FEATURECONTRACT TOOLKIT**

The proposed BPM infrastructure comprises four organizations: consumer, provider, negotiator and monitor. The **Consumer Organization** includes: i) a structure called **WS-Contract Definition** responsible for negotiation and establishment of WS-Contracts based on features; ii) a structure **WS-Contract Execution** responsible for the business process execution; and, iii) a **SOC System** necessary if the consumer services are part of the business process to be executed. In the **Provider Organization**, the SOC System control the Web Services subcontracted by the consumer. The **Monitor Organization** has one structure **WS-Contract Monitoring** that follows the business process execution guided by the QoS terms contained in the WS-contract for service monitoring. The **Negotiator Organization** has one structure **WS-Contract Negotiation** that uses a set of protocols responsible to (re)negotiation of contracts between providers and consumers.

The FeatureContract toolkit, extended in this work, implements the structure WS-Contract Definition and has the following functionalities (Fantinato et al. 2008): i) Feature models creation and management; ii) Services section, business process and agreement terms creation from the feature model; iii) Contract model management through edition of its services section, business process and agreement terms; iv) Feature model configuration that represents the services and related QoS levels to be contracted; and, v) e-contract instantiation. The FeatureContract has a set of components that supports the different stages of contract establishment, some were reused from other research and development groups whereas others were developed by our research group. A description of each component is presented as follows:

- **FeaturePlugin**: supports the elaboration of feature models and also the configuration of such models. It was developed by Czarnecki research group (Antkiewicz & Czarnecki 2004), as an Eclipse plugin, and it implements cardinality-based feature modeling. Only a few adaptations were necessary in order to incorporate it as a component of the FeatureContract toolkit;
6. APPLICATION EXAMPLE

This section presents a summary of an application example to show the feasibility of the proposed approach. This example requires a contract to be established between two fictitious organizations: A Travel Agency and an Airline Company. The Travel Agency (Consumer)'s system supports tourism packages sale by composing third-party services; e-services provided by the travel agency include: customer notification, offers advertisement and travel packages that consume Airline Company e-services. The Airline Company (Provider)'s system supports flight ticket sale and booking; e-services provided by the airline company include: flight booking and purchase; seat and food selection. The stages for e-contract establishment are presented, only in the Airline Company perspective, as follows.

In the first stage (Services Feature Model Elaboration), each company must elaborate a feature model to represent: the provided e-services; the QoS levels related to each service; and the control operations related both to e-services and/or QoS levels. Figure 4(a) presents a feature model example for an Airline Company, which has two service groups: flight services group and services selection group. This first group has the following services: timetable query, flight purchase and flight booking; and the services selection group has seat selection and food selection. Only some items are expanded. The Airline Company feature model gives the consumer (Travel Agency) a structured overview of which e-services, QoS levels and control operations are offered by the provider. During the feature model configuration stage, described in the following, with the control operations associated to the feature model, the consumer can choose e-services and their QoS levels with control operations to be applied when QoS levels are not met. With the services features models elaborated, XML files can be exported by the FeaturePlugin to be used in e-contract template creation.

In the second stage (e-contract Template Creation), the XML files exported from the feature models are used by the FeatureContract to extract the WSDL and WS-Agreement sections of the template. Moreover, the Eclipse BPEL plugin is used to define the business process flow. However, BPEL and WSDL sections of the e-contract are not addressed in this paper since the extension proposed here is more closely related with the WS-Agreement section. The template is composed by all the services and related QoS levels offered to be contracted with the respective control operations in case of contract violation, as present in Figure 4(a).
The main idea is to create one template that can be instantiated many times with different feature configurations. Figure 5 shows a WS-Agreement excerpt that represents renegotiation related to a QoS Level. Not all the possible values are depicted due to lack of space. Latter they will be removed, only the level chosen in the feature model configuration will be left as described in the following subsection.

![Feature model configuration for the airline company](image)

Figure 4. (a) Feature model; and, (b) Feature model configuration for the airline company

In the third stage (Services Feature Model Configuration), the Travel Agency system selects the e-services to be contracted from the Airline Company system, QoS levels and control operations related to these services are also configured. As depicted in Figure 4(a) by the symbol *, all the features of the flight services group are mandatory, the symbols o and □ represents optional features that can be chosen by the contracting party. Figure 4(b) depicts an example of a feature model configuration. The control-operations sub-tree is associated to the flight-purchase e-service and its QoS attributes. The renegotiation of price is defined as the default control operation for the flight-purchase service, which must be triggered if any QoS attribute, for which there is no specialized control operation, is not met. Latter in the e-contract instantiation the penalty tags, related to the flight-purchase e-service, that represents renegotiation of clause and variable will be removed from the e-contract template. Only the penalty that represents price renegotiation will be left.

![WS-Agreement Template Excerpt](image)

Figure 5. WS-Agreement Template Excerpt

In the last stage (E-contract Instantiation), the feature models configurations of the Airline Company and Travel Agency systems indicate which information are relevant to a specific e-contract instance. From the feature models configurations, new XML files are generated. Using the WS-Contract Factory, of the FeatureContract toolkit, all the information related to e-services, QoS attributes and control operations not selected during feature models configuration are removed from the e-contract template. New WDSL, WS-Agreement and WS-BPEL files, containing only the information necessary to the specific agreement, form a new e-contract instance, including control operations information related to Web Services and QoS levels.
7. CONCLUSIONS AND FUTURE WORK

This paper has presented an infrastructure that supports e-contract negotiation, establishment, enactment and renegotiation. The emphasis was in the FeatureContract toolkit and the control operations in the contract metamodel. The main advantages of the approach are: i) efficient information management, organization and reuse, necessary for the negotiation and renegotiation of e-contracts through feature modeling utilization; ii) understanding of the renegotiation process through identifying all possible alternatives to dynamically adjust the e-contract; iii) information organization and presentation of e-services and QoS attributes linked with control operations using feature modeling; and, iv) extension of the WS-Contract metamodel to include the control operations elements already supported by the feature models.

As some lessons learned, the use of feature models to represent control operations to handle e-contract violations makes this process easier to understand, simple and systematic. It can bring benefits for distinct stakeholders, at different levels, as it has a high level of abstraction and structured way for representing information. However, some disadvantages or limitations of the approach can be pointed out: i) necessity of knowledge about the feature modeling; and, ii) negotiation is still made in an offline way; negotiation protocols have not yet been included to automatically perform negotiation.

Future works, besides focusing on the weaknesses cited above, include: full implementation of the WS-Contract Negotiation element with some example protocols; and integration with the WS-Contract monitoring tool which has been developed by the same research group.

ACKNOWLEDGEMENT

This work was supported by The State of São Paulo Research Foundation (FAPESP) and The National Council for Scientific and Technological Development (CNPq), Brazil.

REFERENCES