Towards a methodology for deriving contract-compliant business processes

Zoran Milosevic[1,2], Shazia Sadiq[1], Maria Orlowska[1]

[1] School of Information Technology and Electrical Engineering
The University of Queensland
Brisbane, Australia
{zoran, shazia, maria}@itee.uq.edu.au
[2]also with Deontik Pty Ltd, Australia
zoran@deontik.com

Abstract. This paper presents a methodology for deriving business process descriptions based on terms in business contract. The aim is to assist process modellers in structuring collaborative interactions between parties, including their internal processes, to ensure contract-compliant behaviour. The methodology requires a formal model of contracts to facilitate process derivations and to form a basis for contract analysis tools and run-time process execution.

1 Introduction

Several types of requirements are to be considered in any business process management activity, be it a process design, process execution or process monitoring. Internal requirements reflect strategic goals for improving business outcomes, e.g. an increased efficiency. External requirements reflect constraints from outside world, e.g. contractual obligations with trading partners or regulatory policies with which parties need to comply. Future requirements address likely future states of affairs, e.g. organisational commitments from new contracts. This paper addresses contract-based requirements, namely how to design contract-compliant processes between parties.

Section 2 classifies legal statements in contracts into several types of contract conditions, described in a form suitable for translation into a formal contract expression. One such formalism is briefly discussed and the translation is illustrated by means of an example. Section 3 presents our methodology for deriving contract-compliant interactions between parties, covering cross-organisational interactions, internal processes and supplementary activities. Section 4 provides concluding remarks.

2 Transforming legal statements into formal representation

A contract is a legally enforceable agreement specifying mutual promises between legal entities, e.g. Subcontractor and Outback Water (OW) in the Maintenance Service

Contract example below. Contracts are typically written using legally-centric contract statements, as illustrated through the following example.

MAINTENANCE SERVICE CONTRACT

This agreement BETWEEN Outback Water (To be known as the OW) AND OZ Pumps (To be known as the Subcontractor) governs Maintenance Services (to be known as Service) subject to the following terms and conditions:

1 Definitions and Interpretations
1.1 Price is a reference to the currency of the Australia unless otherwise stated.
1.2 MTBF is Mean Time Between Failures and MTTR is Mean Time To Repair
2 Commencement and Completion
2.1 The commencement date is scheduled as January 30, 2006.
2.2 The completion date is scheduled as January 30, 2007.
2.3 The (OW) shall notify the (Subcontractor) of possibility of extension for 1 year by 3rd quarter of the contract
3 Service and QoS Delivery
3.1 The (Subcontractor) shall make its best efforts to ensure that the following QoS conditions are met:

- not exceed the maximum asset down time on any one asset
- average above the specified MTBF and below the MTTR over a month

The maximum or minimum values are provided in schedule A of the contract.
3.2 (Subcontractor must inform (OW) within 24 hours of any event that might affect the ability to achieve the quality of service
3.3. The (Subcontractor) shall not re-assign maintenance to another party, i.e. Sub-Subcontractor
3.4. The (OW) will provide access to all asset sites based on service requirements
4 Reports and notifications
4.1 The (Subcontractor) will submit monthly reports on all preventative maintenance activities and emergency events, including full timing details.
4.2 The (OW) will provide list of assets to be maintained, with clear instructions of the maintenance cycles required
4.3 The (OW) will provide clear MTBF and MTTR targets
4.4 The (OW) will provide feed back to the subcontractor of any information received about problems with the water supply, including emergencies reported by its customers within 24 hours
4.5 After each of the 1st and 2nd quarters, the (OW) will give guidance to the subcontractor on how any shortcomings in the service might be improved.
5 Payment
5.1 The (Subcontractor) shall submit monthly invoices to (OW) for services performed during that period
5.2. The (OW) shall make full payment of (Subcontractor) invoices within 30 days of receipt
6 Termination
6.1 The (OW) can terminate the contract after three QoS violations

Our analysis of many contracts suggests that legal statements can be classified into several groups whose structure is amenable for formal representation, namely:

1. The declaration of *pre-existing external constraints* from the environment which apply to the contract or to the variables in the contract, such as policies originating from taxation law or business contracts law (e.g. clause 1.1 in the example);

2. *Definitions,* explaining meaning of contracts terms (e.g. clause 1.2 in the example);
3. A *period of validity* when the contract is in effect (e.g. clauses 2 in the example);
4. The statement of *core normative policies*, i.e. obligations, permissions, prohibitions that apply to the parties (e.g. clauses 3, 4, 5 and 6); some obligation policies represent high-level constraints, stating a goal to be achieved (e.g. clause 3.1);
5. Other type of policies used in typical business/legal jargon, which can be reduced to the core policies; we call these *compound normative policies*; examples are rights, liabilities and responsibility (there are no such policies in our example);
6. Actions that cover transfer of normative modalities between principals and agents, as in delegation statements; we call these *policy-transfer actions* (e.g. clause 3.3);
7. Events that signify policy violations occurrence or situations potentially leading to future violations (e.g. clause 3.2 and 4.4); we call them *attention events*;
8. Second-effect policies to be invoked in cases of violations of any of the above policies; we call these *reparation policies* (see [1]), e.g. clause 6.1;
9. *Force-majeure* conditions, describing circumstances which are beyond control of either parties; (there are no such policies in our example);
10. A number of *structuring constructs*, e.g. clause groups 1-6.

We show through the example how the above structures can be mapped onto Formal Contract Language (FCL) [1]. FCL statements include triggering conditions for policy activation (e.g. *AccessSiteRequest* in clause 4.1) and deontic conditions. The latter consist of deontic modality (O for obligation, P for permission and F for prohibition) and the subject's behaviour expression (e.g. $O^{H}_{OW,\ Sub}$ *ProvideAccess* is OW's obligation to ensure Subcontractor's access). Note that 'H' superscript denotes a high-level policy, while 'D' denotes an action of delegation. The contract in FCL is:

2.3: 3rdQuarterEnd,ExtensionYes ⊢ OOW, Sub ExtensionNotification

3.1: ContractStart ⊢ O HSub EnsureBestQoS

3.2: OoSProblemEvent ⊢ OSub,OW InformWithin24hrs

3.3: ContractStart ⊢ F DSub, Sub-Sub AssignMaintenance

3.4: AccessSiteRequest ⊢ O HOW, Sub ProvideAccess

4.1: ContractStart,BeginMonth ⊢ O Sub,OW SubmitMonthlyReport

4.2: ContractStart ⊢ O HOW ProvideListOfAssets

4.3: ContractStart ⊢ O HOW ProvideMTBFandMTTRTargets

4.4: ProblemOrEmergency ⊢ O OW,Sub ProvideFeedback

4.5: EndOfFirstQuarter ⊢ O OW,Sub GiveGuidance;

EndOfSecondQuarter ⊢ O OW,Sub GiveGuidance;

5.1: BeginMonth ⊢ O Sub,OW SubmitMonthlyInvoice

5.2: InvoiceReceipt ⊢ O OW,Sub FullPaymentWithin30days

6.1: ThirdQoSViolation ⊢ POW TerminateContract

The FCL can express predicates such as those included under groups 1, 2 and 3 above, but they are not described in this paper. We plan to extend FCL in future to support complex contract conditions grouped under groups 5, 9 and 10.

3 Methodology

We exploit FCL contract form in initial steps of our methodology for constructing contract-compliant business processes (Fig. 1). This methodology is developed to cover various circumstances surrounding the establishment of contracts, as well as subsequent measures for ensuring contract-consistent behaviour. For example, parties may enter contract afresh, to reflect new collaboration opportunities and without limitations imposed by their established internal processes, policies or commitments to other parties. But it may be that their existing processes and policy present conflicting conditions with the new contract. These conflicts may require renegotiation of contract terms or adaptation of the existing processes or policies to align the existing and new policy spaces. In order to detect such conflicts, the first step is to undertake static analysis of contracts, possibly involving various types of simulations. Provided the conflicts are resolved, the problem of ensuring contract-consistent behaviour is reduced to first ensuring that each party formulates its *collaborative* interactions, directly reflecting contract conditions, and then ensuring that they formulate *internal* processes to fulfil contract constraints. Both collaborative interactions and internal processes may be augmented with *supplementary* processes that track interaction progress, detect potential future policy violations and send notifications to the parties to that effect. The methodology consists of several steps as described next.

In order to derive *collaborative* interactions, or a contract framing behaviour, directly reflecting constraints in contract conditions, one can start with the identification of the primitive actions that each party is *required* to carry out, as stated in the contract, i.e. obligation modalities. Typically, one would first consider (simple, but no high-level) obligation modalities in which both subject and beneficiary are explicitly mentioned. This helps identifying messages to be sent between partners and their direction. The messages either reflect consideration aspect of contract, or have purpose of sending notifications to the other party, e.g. about progress of some activity or a warning about likely or an occurring violation. In our example, one would go trough the FCL version of contract and identify modalities of the form $O_{OW,Sub}$ or $O_{OW,Sub}$, followed by the identification of the actions that may result in messages of some form, e.g. *Inform (Within24hrs), SubmitMonthlyInvoice, SubmitMonthlyReport, ProvideFeedback*. Some messages may be a result of complex internal processes. For example, *SubmitMonthlyReport* message can be a result of multiple internal process steps within the Subcontractor organisation. High-level obligations, e.g. O^{H}_{Sub} *EnsureBestQoS*, can be refined in terms of specific QoS indicators and the corresponding objectives. Similarly, one would also identify those actions that *must not* be carried out by the parties, e.g. $F^{D}_{Sub,\ Sub\text{-}Sub}$ *AssignMaintenance;* (superscript ‘D’ denotes that this prohibition applies to a delegation action). Note that the prohibition modalities will typically result in *supplementary* processes whose purpose is to realise mechanisms to prevent the occurrence of prohibited behaviour. Additionally, one would also identify compound normative concepts that have elements of obligations, such as duties and responsibilities and similarly, identify flow of messages or notifications.

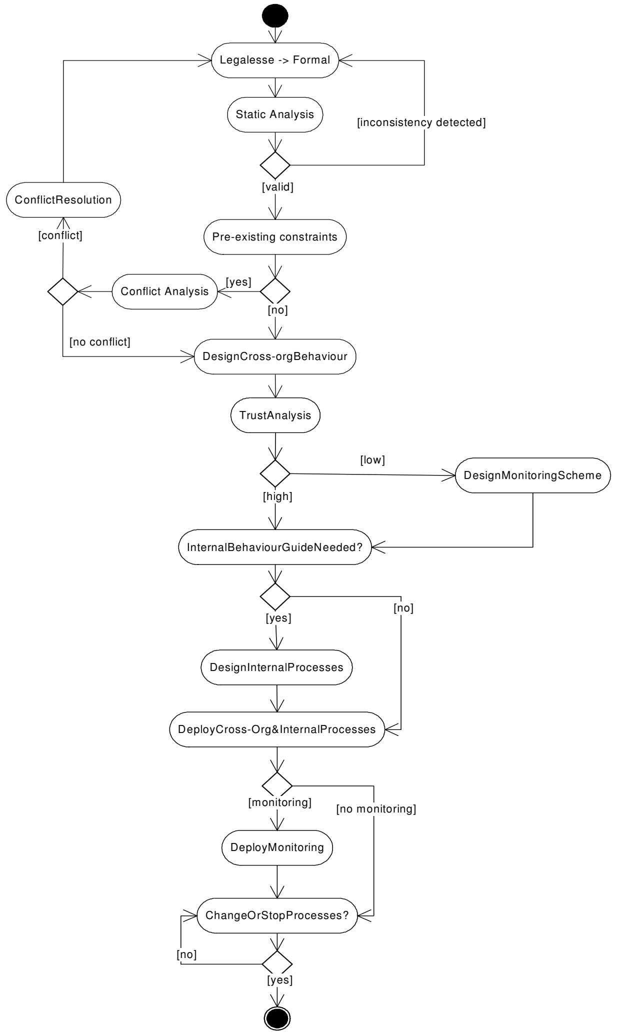

Figure 1: Methodology for deriving contract-compliant processes

Subsequent steps determine candidate *internal processes* for the parties, compliant with the contract framing behaviour. To this end, various heuristics can be applied to reflect different types of contract conditions, as proposed in [2]. Examples of such condition types are *exception conditions*, specifying actions to be done when a violation occurs and *quality oriented* conditions which imply some inspection stage. The respective heuristics are introduction of escalation branches in the internal process, and introduction of loops to check deliverable. Such heuristics will be captured over time. For some conditions it may be impossible to determine any heuristic. In our example, the required action of the Subcontractor to send monthly report imply that the Subcontractor needs to structure own processes to accomplish this, e.g. they may introduce a Work Order processes, followed by an Assemble Report activity, the completion of which generates *SubmitMonthlyReport* event.

Further step is to design *supplementary* activities for the internal business processes, e.g. tracking of the progress of internal processes and checking whether there is a likelihood of contract violations. The aim is to provide early detection of significant state of affairs that need further attention about which parties need to be notified. Lastly, *policy checking procedure*s step can be deployed in run-time to detect existing violations and activate fallback procedures; depending on the level of trust, this can be part of parties' internal processes or using trusted third party monitor.

This section focused on a methodology for creating new processes to be conformant with the contract. In many cases however, a contract will govern existing processes or augmented versions of those. The FCL formalism is applicable to both cases, and we expect that much of the methodology presented here can be reused.

4 Conclusions and Future Work

This paper presented a methodology for deriving business process descriptions based on business contract terms, to assist process modellers in constructing contract-compliant processes. The methodology provides guidelines for structuring collaborative interactions, internal processes and a number of supplementary processes. There are only few papers that dealt with links between contracts and business processes [3, 4, 5, 6]. To the best of our knowledge, the area of deriving contract-compliant processes has not been investigated so far, apart from some initial ideas presented in [2].

We hope that this paper opens new questions and stimulates further development in the area. Our own plans are to consider specific process specification languages as a target option for our derivation, in particular BPMN and BPEL, and to revise the methodology, in particular regarding the derivation of internal processes.

References

1. G. Governatori, Z. Milosevic, Dealing with contract violations: formalism and domain specific language Proc. of the 9th International Conference on Enterprise Distributed Object Computing, Enschede, The Netherlands, Sep. 2005
2. R. Tag, Z. Milosevic, S. Gibson, S. Kulkarni, Supporting Contract Execution through Recommended Workflows, DEXA04 Conference., Zaragoza, Spain, Sept. 2004.
3. W-Jan van den Heuvel, H. Weigand, Cross-Organisational Workflow Integration using Contracts, Decision Support Systems, 33(3): p. 247-265
4. P. F. Linington, Automating support for e-business contracts, International Journal of Cooperative Information Systems, 14(2&3):77-98, September 2005
5. R.M. Lee. A logic model for electronic contracting. Decision Support Systems, 4:27–44, 1988
6. A. Berry and Z. Milosevic. Extending choreography with contract constraints. Int. J. of Cooperative Inf. Syst., 14, 2005.