Linking contracts, processes and services: an event-driven approach

Z. Milosevic[1], M. Orlowska, S. Sadiq

School of Information Technology and Electrical Engineering,
The University of Queensland,
Brisbane, QLD 4072, Australia.
{zoran, maria, shazia}@itee.uq.edu.au

Abstract

This paper addresses the problem of mapping business contract conditions onto the messages and rules that represent service interactions in a collaborative business process. We describe why this mapping is not straightforward by means of an example. We then consider a message-driven process language as a target for the mapping and use this mapping solution to discuss broad range of problems related to the mapping problem.

Keywords: *Business Contracts, Collaborative Business Processes, Service Interaction*

1. Introduction

This paper addresses the need to establish closer links between behaviour defined in business contracts and in the processes governed by the contracts. Both kind of behaviour deal with the concept of service, which we take to be a 'particular abstraction of behaviour that expresses guarantees offered by a service provider'[1]. In case of contracts, services are mostly abstracted through their obligations and other policies. In case of processes, services are abstracted in terms of individual activities or as a composition of several activities forming the process. So, service is a key link between contracts and processes and is of particular importance for cross-organisational environments, increasingly requiring more *transparency* in interactions between partners, while still preserving *confidentiality* of internal structures of their processes.

The *event-driven* architectures [4] provide a suitable environment to express *process behaviour* in sufficient detail to ensure this transparency. They are also suitable for expressing various levels of specification granularity to reflect the confidentiality requirement and the level of external exposure of its processes, e.g. via services. The event paradigm is well suited for the expression of *contract conditions* because events can be used to signify occurrences that reflect key contract conditions. The use of events as a common basis for describing behaviour in contracts and processes facilitates mapping between these two modelling artefacts. This paper uses a specific event-driven process approach, called Harmonized Messaging Technology (HMT) [3] to illustrate the mappings. The HMT utilizes a Harmonized Messaging Calculus (HMC) to describe messages and rules.

One aim of this paper is to apply the HMC to the domain of business contracts, thereby providing insights into the issues and challenges involved in the mapping between the two behaviour models. A consequent aim is to discuss the broader set of issues associated with the mapping of contracts onto the corresponding processes. The mapping is not necessarily straightforward because the nature of behaviour specifications in contracts and in processes is quite different. While contracts specify 'what' aspects of a behaviour need to be satisfied and require observation mechanism to detect possible violations, processes specify 'how' aspects of behaviour. There can be many ways of implementing processes that satisfy contract conditions and parties will typically define their own processes to suit their internal needs and competitive positions. Our early ideas presented in [2] have identified some initial heuristics for deriving processes based on various styles of contract clauses which could serve as a guide in deriving 'recommended' business processes.

[1] Also with Deontik Pty Ltd, Australia (www.deontik.com)

In the next section we consider current treatment of contracts as part of enterprise systems and discuss new requirements pertinent to the cross-organisational processes. This is followed by the description of key semantic concepts of contracts and the description of HMC concepts used as a target language for expressing the contract-governed processes. We illustrate key mapping between contracts and processes by means of an example. The example is outlined in section 4, and key mappings are described in section 5 and 6, in the context of contract establishment and contract execution phases respectively. The paper finishes with conclusions and a list of future research directions.

2. Contracts and business processes

Key problems related to cross-enterprise role of contracts and their linking with business processes are:

1. At present, contracts are mostly treated as legal documents separated from their function as a governing mechanism for cross-organisational processes. The contract documents are typically stored in various databases and spreadsheets scattered around organisations. To address this problem, there is an increased push by system integrators and specialised enterprise contract management vendors, to provide more visibility into contract related data and to support more automation of contracts in their entire life-cycle, establishing better linkages between contracts and processes than is currently available today.
2. The different styles of behaviour used to express contracts and processes requires transformation between these two business concepts, which is not so straightforward and may require applying various heuristics [2].
3. To express a business process as a transformation target of a business contract, one needs a suitable process language. There are various such languages and this paper considers an event-based process language, namely HMC.
4. Since parties to the contract are autonomous entities, there may be various reasons for them failing to carry out the activities as required by the contract. To this end, it is useful to provide a support for an automated contract monitoring, which is another non-trivial problem. Note that contract monitoring is not the focus of this paper but several specialised languages such as Business Contract Language (BCL) have been proposed [2][6][7].

The focus of this paper is on the third problem above.

3. Foundation concepts

This section outlines key concepts for the specification of contracts and key aspects of the HMC.

3.1 Key components in business contracts

Key concept for describing contract is that of a policy, stating constraints for the parties to the contract. Key policy concepts are obligations, permissions and prohibitions. ***Obligations*** state behaviour which is required of the signatories to the contract, ***permissions*** express the allowed behaviour and ***prohibitions*** state what they are not allowed to do. Contracts can specify ***violation*** conditions and the policies that take effect as a result.

3.2 Key HMC concepts

The HMC provides a formal foundation for the specification of complex rules that govern (message based) interactions in collaborative business process. We outline key concepts of the HMC, needed for the subsequent mapping discussion. Further details on the safety and expressiveness of the calculus can be found in [3].

Collaboration Space (CS) is defined by a 3-tuple <Set of Message Types, Set of Participants, Set of Rules >. This concept is inspired from the concept of database space in relational database systems. The deployment environment of HMC can have several collaboration spaces, each with its own rules, message types, and participants similar to a DBMS having multiple database spaces.

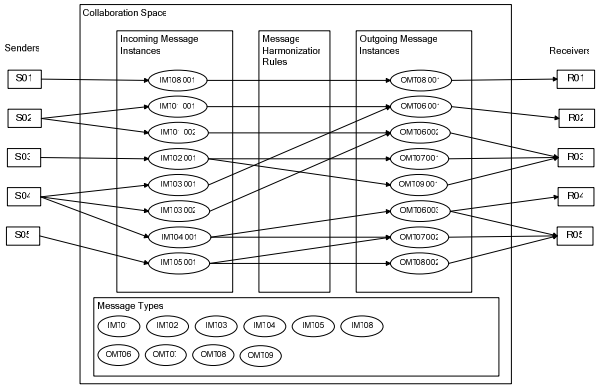

Figure 1: HMT foundation concepts

Message is the fundamental unit of data exchange for the expression of collaboration interactions. ***Message types*** provide the schema that all message instances must conform to. When required, we distinguish between two sets of message types: incoming messages or IMTs, and outgoing messages or OMTs. In figure 1, IMT01, IMT02,

OMT08, OMT09 are the message types. Typically message types are expected to be derived from standard definitions such as [15].

A ***participant*** is a primitive concept often abstracted as a role type and used to represent a component process, an organisation, or a program participating in collaboration. A participant can send and/or receive message instances

Rules constitute the most critical aspect of a collaboration space. The ability of the system to receive and generate instances of certain message types represents the system behaviour, and rules provide the means of enforcing this behaviour. For example, in figure 1 there can be a simplistic rule "upon the arrival of an instance IMT08 from S01, compose a new instance and send it to R01". A given rule type may have several rule instances simultaneously active at run-time.

A Rule is seen as a typical *condition* → *action* structure. There are three fundamental questions that need to be addressed in the overall rule evaluation:

(1) Which OMT will be initiated in response to the occurrence of certain events (events relate to the arrival of an IMT and/or timer events)

(2) What values will be assigned to the fields of the selected OMT instance(s)

(3) When the OMT instance(s) will be dispatched

Based on the above three fundamental questions, we identify three types of actions.

(1) Associate: associates an event with the suitable OMT

(2) Assign: assigns values to corresponding fields of an OMT to create OMT instances

(3) Send: sends the completed OMT instance

Consequently, there are three rule types, each dealing with one action type as elaborated in the next sections.

4. Contract and processes - an example

This section describes, by means of an example, issues associated with linking contracts and processes.

4.1 Background

We consider two contract automation phases, the *contract establishment* and *contract execution*. We distinguish between these phases for two reasons. Firstly, the message types involving interactions between partners and the corresponding rule type sets are different for the two phases, indicating a need to characterise them as two separate HMT Collaboration Spaces (CS). Secondly, and more importantly for the purpose of this paper, there is a different style of interactions between partners in these two phases. The contract establishment phase has an interaction-driven style of behaviour while the contract execution has a constraint-driven style, characterised by the obligations, prohibitions or permissions constraints. The use of an event-oriented approach for specifying both the interactions and constraints allows a uniform way of describing key behavioural aspects and more direct mapping between them.

4.2 Contract conditions

We now introduce an example initially presented in [2] to provide means for a detailed analysis of the problem. The example is centred on an Outback Water (OW) company, providing irrigation water to agriculture, industry (primarily mining and oil/gas extraction) and small towns in some central parts of Australia. The OW operates some storage lakes and both open irrigation canals and pipelines. The OW requires services of a Subcontractor to maintain various assets in the water system managed by this company and issues request for tenders to select a best Subcontractor based on criteria such as their availability, price and past performance. This contract establishment process results in a contract, specifying conditions for servicing and maintaining OW's pumps and related equipment. The contract will govern interactions between the OW and the subcontractor (S), including business processes between them. This is contract execution phase.

For the purpose of the example, we have selected several obligations, permissions and prohibitions, as listed below.

The *Subcontractor (S) obligations* are:

S1. Make best efforts to ensure that the following QoS conditions are met:

- not exceed the maximum asset down time on any one asset, in more than 1% of cases
- not exceed the call-out time limit on more than 5% of emergencies in a month
- average above the specified Mean Time Between Failures (MTBF) and below the Mean Time To Repair (MTTR) over a month.

S2. Submit monthly reports on all preventative maintenance activities and emergency events, including full timing details and description of problems and action taken.

S3. Inform the asset operator within 24 hours of any event that might affect the ability to achieve the quality of service, e.g. resignation of subcontractor engineers, recurring problem with certain asset types

S4. Submit monthly invoices of money due to the subcontractor.

The *Outback Water (OW) obligations* are:

OW1. Pay the subcontractor on monthly invoice within 30 days.

OW2. Provide list of assets to be maintained, with clear instructions of the maintenance cycles required (asset lists are in a schedule to the contract, maintenance manuals are in associated paper or on-line documents)

OW3. Provide clear MTBF and MTTR targets

OW4. Feed back to the subcontractor any information received about problems with the water supply, including emergencies reported by its customers within 24 hours

OW5. Give the subcontractor access to all the asset sites.

OW6. The contract situation is reviewed quarterly by the water utility. After each of the 1st and 2nd quarters, the OW must give guidance to the subcontractor on how any shortcomings in the service may be improved.

The *OW permissions* are:

OW7. Take on an additional subcontractor in the event that the appointed subcontractor is having difficulty in meeting the QoS targets.

OW8. As part of the contract review process, after the 3rd quarter of the contract the OW is allowed to give the subcontractor notice to quit or to be asked to continue for another year.

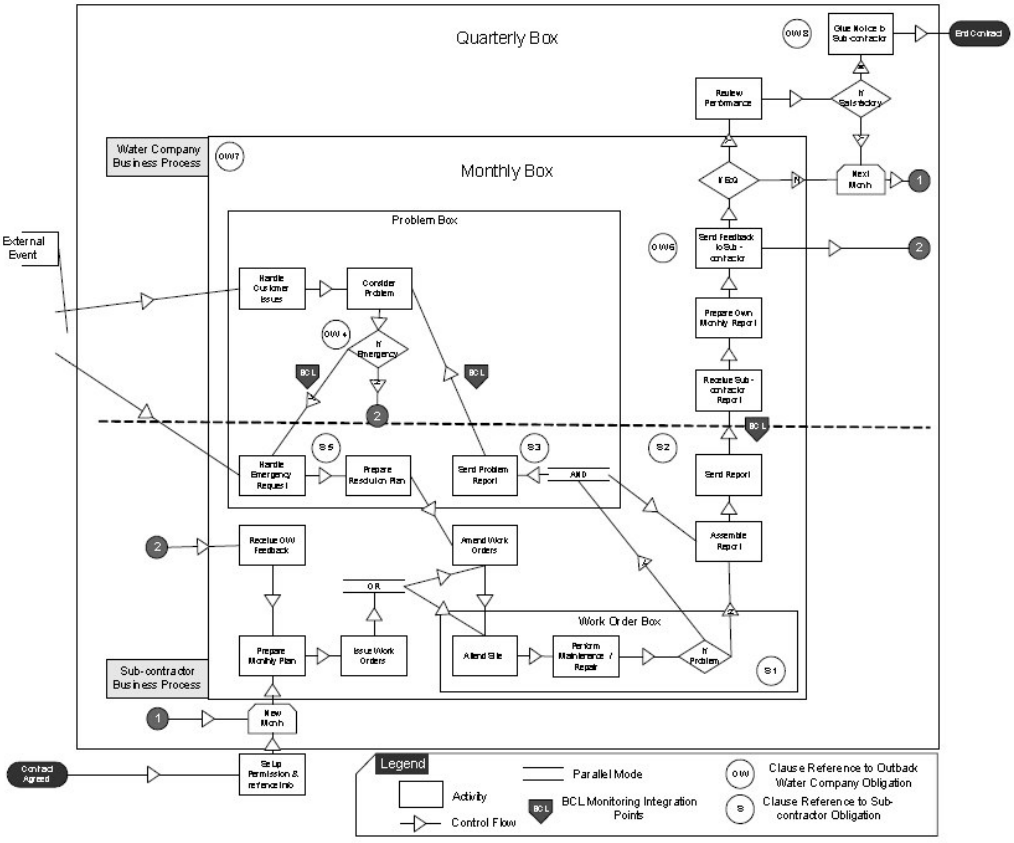

Figure 2: Collaborative Business Process for Outback Water and its Sub-contractors

The *subcontractor prohibitions* is:

S5. Not allowing to re-assign maintenance tasks to a sub-sub-contractor.

In both of these ***collaboration spaces***, namely the contract establishment and contracting execution spaces, the ***participants*** can be considered the same, namely, Outback Water (OW) company, OW Customers (OW-C) and Subcontractor (S);What is different is the specification of ***message types*** involved and ***rule types*** that drive the collaborations. There may be other contracts pertinent to these collaboration spaces, but they are not considered in this example, e.g. a contact between the OW and its customers.

4.3 Business processes description

The contract conditions presented in previous section provide a broad set of behaviour constraints needed to satisfy mutual obligations and expectations of parties. However, the contract does not prescribe the structure or form of the corresponding processes and leaves this as an internal matter for each of the parties, as an additional level of detail, beyond contract scope.

One possible way of specifying processes to meet contractual conditions for this example is shown in Figure 2. This figure depicts cross-organisational process involving the OW and the S. This process consists of both the internal processes for each of the parties and the exchange of messages across their organisational boundaries. As can be seen from the figure, this business process provides a number of activities which are not mentioned in the contract, but which are needed to satisfy contract conditions. Consider the second obligation for the subcontractor regarding the submission of monthly reports. In order to discharge this obligation (and do it on a recurrent basis), a subcontractor will need to implement certain number of activities to ensure that at the end of each month, a detailed description of required information is delivered to the OW. For example, this may involve activities such as 'prepare monthly plan', 'issue work orders' (if there are multiple workers involved for various sites), 'attend site', 'perform maintenance/repair', and if a problem was identified, 'send problem report' to the OW within 24hrs. A complete set of activities for the fulfilment of the second obligation are shown in the bottom part of the Figure 2, which refers to the subcontractor processes.

Some of the activities in the process correspond more directly to the conditions stated in the contract. For example, the 'give notice to the sub-contractor' activity directly reflects the OW's second permission, according to which they are allowed to give the subcontractor notice to quit or to be asked to continue for another year. However, this activity can only be carried out if the OW has fulfilled its dependent obligation, i.e. the obligation to review S's performance. Although not directly specified in the contract, the business process needs to include 'review performance' activity to determine how satisfactory the Subcontractor has performed

5. Mapping: Contract Establishment CS

Contract Establishment process steps are:

i. OW issues Request For Tender (RFT)
ii. Various Subcontractors send offers
iii. OW chooses the best one (the S)
iv. OW sends offer notification to the S
v. S sends Contract Agreed notification

5.1 Message Types

The message types in this collaboration space, sent by the OW and S are identified below. Each message type contains fields, which are assigned 'values' by the HMT system and/or by participant systems.

Messages Sent by the OW are:

- RFT message with several fields including: start date *value*, price field, previous track record fields, ABN field. Note that the start date value is a constant and other values will be assigned by candidate subcontractors.
- Offer Notification message type, with the acceptance value field (which can be Yes/No).

Messages sent by a subcontractor (S) are:

- Subcontractor's Offer (S-OFFER) with several fields including: price quotation value, ABN value and previous job names.
- Contract Agreement notification, with the field of the contract agreed value [which can be YES/NO]

Each dispatch and arrival of the messages is treated as an *event* that will be used as part of rule evaluation and generation of the outgoing message described below. In HMT, events also include the detection of a completion state of a message and occurrence of specific time value. The former is also referred to as the rule type 2 (see below) and will be mentioned in the example below. The later is not mentioned in this CS.

5.2 Rule types

We apply the three types of rules as introduced in 3.2, i.e. associate, assign and send rule types[3], for the processing of messages received and sent by the parties, the OW and the Subcontractor,

Example rules that apply to the business process step ii above are:

Rule 1:

RFT_Received (by S)

IF RFT.StartDate>Date1 THEN ASSOCIATE

S-OFFER

This states that if a RFT message has arrived at the S's site, and if the start date for the contract requested by the OW is after some availability date of the S (Date1), then the HMSS will generate a template S-OFFER which will then be assigned specific values as submitted by the S.

The completion of the S-OFFER document is subject of the rule type 2, namely:

Rule 2:

S-OFFER.Price

IF RFT.StartDate>Date1 THEN S-OFFER.Price = 100k

IF RFT.StartDate>Date2 THEN S-OFFER.Price = 120k, (where Date1 < Date2)

This rule states that the S can quote two prices to include in its S-OFFER message, depending on when the contract is requested to start. If earlier, the S quotes higher price (e.g. because of the requirement to satisfy its existing other commitments).

Rule 3:

Time T

SEND S-OFFER

(Where T < DeadlineForResponse)

This rule states that at the S-OFFER message will be sent by the S before the DeadlineForResponse specified by the OW company. We assume here that the exact point when this is to be sent is programmed as part of the HMMS system or is executed by a human actor in the S company. In other words, there is no policy constraint that governs this event, such as obligation policies in contracts.

The rules above implement the second BP step in the contract establishment process above. For brevity, we will not represent rules for the third step, rather we mention the subsequent rules implementing last two steps in this process. They will be related to the dispatch message, OfferNotification (sent by the OW) and arrival messages ContractAgreed (received by the OW and sent by the S upon they received OfferNotification). In fact, ContractAgreed event is of importance for contract management as well as the subsequent contract execution. This event will be the type 3 event and will be conjunction of two events representing OfferNotification(S) and send ContractAgreed message.

This scenario shows that event driven BPM languages such as HMC can support relatively simple interactions exploited as part of negotiations in the case of contract establishment process. Further research will need to consider more complicated negotiation cases to determine a broader applicability.

6. Mapping: Contract Execution CS

Once the contract has been agreed by both parties, it is their responsibility to ensure that contract conditions are satisfied. They should implement their business activities and business processes which will satisfy the policies that they accepted and that constitute contract conditions. As discussed in [2], the derivation of 'recommended' business processes to satisfy contract conditions is mostly based on applying appropriate heuristics and patterns.

For example, one way of establishing some correspondence between business contract conditions and the compliant business processes is to identify what pieces of a contract can 'drive' the execution of the process. Some examples of such 'driving' points in the contract are:

- external events that start process in the first place, i.e. the signing of a contract, or the point in time when the contract becomes 'active'.
- external events that can affect process execution, arising from the environment in which e-business operates (e.g. new business regulations).
- perhaps most importantly, the normative statements of obligations, as they state what is required to be done by the parties. In a way, there are similarities between obligations and objectives, and one can think of obligations as the objectives driving the process. By refining these objectives one can derive business processes needed to support their fulfilment, in a similar way as was proposed in [12].

We note that permissions are more difficult to be exploited as 'driving' points for processes. This is because they state optional behavioural fragments that do not have to be fulfilled.

Prohibitions can be used as pre-conditions to disallow some behaviour (this is a pessimistic approach, typical of many preventative security systems). However, real life contractual situations may require more flexibility to allow certain prohibitions to be dealt in a more optimistic way, i.e. by not explicitly disallowing behaviour that is contrary to prohibitions, as this might affect many other parts of downstream process.

Recall that there may be many variants of process implemented by the S (and OW) each of which satisfy the conditions in contract. While contract states policy conditions that are required to be satisfied and is silent on the specific form of processes, the processes themselves provide finer-grain behaviour specifications for expressing how these conditions can be fulfilled and typically companies often consider the internals of their processes as sources of their competitiveness. In our example, presented in 4.3, we have derived one such process that corresponds to several policies in the contract given in 4.2.

In what follows we will describe this process using the HMT approach. We first identify the types of HMT messages involved in communications within and across

the business processes of the contracting parties, namely the OW and the S, followed by the description of HMC rules that govern generation of such messages.

6.1 Message types

There are two broad categories of messages used to support business processes in Figure 2. One category is used for cross-organisational interactions. These are likely to be the points where contract monitoring conditions could be specified, such as using BCL expressions (shown by BCL symbols in Fig. 2). Examples are:

- Problem report (sent by the S to the OW upon detection of a problem on a site) containing the fields such as the type of the problem, asset name/location, time of detection etc.
- Report (sent by the S to the OW, containing details of preventative/maintenance work undertaken, including descriptions of the problems identified)
- Feedback message (sent by the OW to the S) containing information about the emergencies reported by the OWs' customers.
- Notice (sent by the OW to the S) about the continuation of the contract.

Another category are messages passed between the activities in internal BPs of each of the partners. They have character of control and data flow as in workflow systems. Examples are: monthly plan completion message (within the S), work order completion message (within the S) and OW's monthly report (within the OW). For brevity, we do not provide a detailed description of the format of messages as was done in 5.1

6.2 Rule types

As in section 5.2, we apply HMC rules to drive process execution, both internal activities and cross-organisational interactions. Several illustrative fragments follow next.

Rule 1:

NewMonth

ASSOCIATE MonthlyPlan

This rule is fired with the start of new month (NewMonth event, shown as filled circle 1 in Fig.2).

Rule 2:

IF Problem

THEN MonthlyPlan.Update =

ProblemText

ELSE IF Feedback MonthlyPlan.Update =

FeedbackText

ELSE

MonthlyPlan = StandardPlan

This rule will be fired when the first rule above has been executed. If, during previous maintenance period some Feedback was generated by the OW, or some Problem was detected by the S (but which does not have an emergency status and is only noted for the subsequent month maintenance) the MonthlyPlan Update field will be appropriately filled. Otherwise, a routine maintenance should proceed according to the StandardPlan (both of these data flows are shown as filled circle 2).

Rule 3:

NewMonth (AT $14*inv_{day}$ + $ThisMonth*inv_{month}$)

IssueWorkOrder

This rule states that each first day of the month, at 14:00, the new monthly WorkOrder document should be sent to appropriate maintenance staff. Note that according to the HMC semantics, each of the Send rules should also include explicit expression of the completion of previous assign rule, but we do not show this here for simplicity.

7. Related work

Service oriented architectures (SOA), such as Web Services provide an important step in achieving ***service enablement***, allowing better links between business and technology architectures. This is a step towards ***service communication*** utilising new generation of messaging technology, providing an intelligent middleware solution that can scale beyond the traditional hub-and-spoke message broker, leading to the Enterprise Service Bus (ESB) functionality [16]. Recent developments from business software vendors have identified the need for solutions which go beyond service enablement and communication capability, and provide a development environment that allows multiple services both within and across enterprise systems to be collated into value added composite applications, e.g. Enterprise Services Architecture [17] and associated Composite Application Framework [18] from SAP. A critical aspect of current enterprise architectures based on the above approaches, is the management of the rules for ***service interaction***.

As stated in the introduction, support for services and service interactions, is part of the automation of business contracts, business processes and their mappings. The last of these has been addressed in this paper, namely the mapping of business contracts into service interaction rules. With respect to business processes, there is evidence that the efforts are increasingly directed towards event-based technologies such as Rapide [4], Web Services Choreography Language [13], and often based

on messaging infrastructures HMC [3] and BPEL [10]. With respect to business contracts support, research efforts aim at providing event-based contract languages used for the expression of contract semantics, e.g. Contract Expression Language [14] and BCL [6][11]. The use of event-based approaches facilitate their alignment and mapping, which is the basic premise of this paper.

This paper is one of very few that considers positioning of business contracts and business processes. A few notable contributions in this regard are [9] that developed an approach for linking specification of contracts with workflow systems and [2], which also provides an approach for deriving processes based on contract conditions. This paper leverages and extends the initial results of that paper.

8. Conclusions and Outlook

This paper described an important aspect of systems with large scale service interactions such as in collaborative business process management. This aspect is related to providing better linkages between business contracts and processes that implement them. We have covered contract establishment and contract execution phases and used an example from a cross-organisational setting to illustrate contract automation in the two phases. Although the example was presented through a specific process language, the discussion was intended to extract general concepts regarding the positioning of processes in relation to contracts. We highlighted the difficulties in deriving processes to support the satisfaction of contract conditions and, in addition, the need to provide special processes to detect contract violations and act upon them.

We intend to continue applying the event driven process approach in the domain of contracts by using other contract scenarios. We also plan to investigate constraint-based approaches for the specification of business processes, such as those reported in [8], as a style of specification closer to that of business contracts. These may provide another target for the mapping of contracts to processes. In addition, we will investigate whether generic mapping patterns can be reused to derive process templates from standard contract templates. Lastly, we see substantial potential in exploring the possibility of the development of a toolkit that can be used to automate such a ***mapping*** as far as possible, provide critical ***validation*** functions on the resultant process specification, and where possible provide support for the ***visualisation*** of contract clauses and their relationships to enable reasoning about contract completeness and correctness. Our early ideas in this area are presented in [11].

References

[1] ITU-T Rec. X.902 | ISO/IEC 10746-2: Foundations, RM-ODP

[2] R. Tag, Z. Milosevic, S. Gibson, S. Kulkarni, Supporting Contract Execution through Recommended Workflows, DEXA04 Conference., Zaragoza, Spain, Sept. 2004.

[3] Dat C. Ma, Maria E. Orlowska, Shazia W. Sadiq *Formal Considerations of Rule-Based Messaging for Business Process Integration,* Special Issue of Cybernetics and Systems: An International Journal, Vol 37/2 (2006)..

[4] D. Luckham, *The Power of Events*, Addison-Wesley, 2002

[5] ISO\IEC IS 15414, *Open Distributed Processing-Enterprise Language*, 2002.

[6] P. Linington, Z. Milosevic, J. Cole, S. Gibson, S. Kulkarni, S. Neal, *A unified behavioural model and a contract for extended enterprise*, Data Knowledge and Engineering Journal, Elsevier Science, 51 (2004), p. 5-29.

[7] G. Governatori, Z. Milosevic, *Dealing with contract violations: formalism and domain specific language* Proc. of the 9th International Conference on Enterprise Distributed Object Computing, Sep. 2005

[8] Shazia Sadiq, Wasim Sadiq, Maria Orlowska (2005) *A Framework for Constraint Specification and Validation in Flexible Workflows.* Information Systems, Elsevier Science. Volume 30. Issue 5 pp. 349-378

[9] W-Jan van den Heuvel, H. Weigand, *Cross-Organisational Workflow Integration using Contracts,* Decision Support Systems, 33(3): p. 247-265

[10] www-106.ibm.com/developerworks/library/ws-bpel/

[11] Z. Milosevic, G. Dromey, *On Expressing and Monitoring Behaviour in Contracts*, EDOC2002 Conference, Lausanne, Switzerland.

[12] J. Cole, J. Derrick, Z. Milosevic, K. Raymond, *Author obliged to submit a paper before July 4: Policies in Enterprise Specification*, Proc. Policy01 workshop.

[13] Web Services Choreography Description Language Version 1.0, W3C Working Draft 17 December 2004

[14] http://www.crforum.org/papers/CEL-BCF-Whitepaper.pdf

[15] Universal Business Language 1.0, http://docs.oasis-open.org/ubl/cd-UBL-1.0/, 2004.

[16] Chappell, D. A. *Enterprise Service Bus* (1st ed.). Sebastopol, California: O'Reilly Media, Inc. 2004

[17] *Enterprise Services Architecture* http://www.sap.com/solutions/esa/index.epx

[18] *SAP Composite Application Framework* http://www.sap.com/solutions/netweaver/cafindex.epx