Contract Workflow Model Patterns Using BPMN

Vandana Kabilan

Department Of Computer and Systems Sciences

Royal Institute of Technology and Stockholm University, FORUM 100, Kista, SE 164 40, Sweden

E Mail: vandana@dsv.su.se

Abstract.. Business Process Models are typically used to express inter or intra – enterprise business activities/processes. Contractual obligations need to be fulfilled through execution of business processes on behalf of the contracting parties . To do so, business contract terms and conditions need to be semantically integrated to existing internal business process models. Contract *obligation, performance, non-performance* and other related concepts have been expressed as conceptual models in a Multi-Tier Contract Ontology (MTCO). Based on the MTCO, business process modelers may model the contract obligation fulfillment process as Contract Workflow Models(CWM) using Business Process Modeling Notation (BPMN) diagrams. The paper discusses the ongoing research and choices made in the semantic translation from contract obligations to CWM –BPMN Diagrams. Some of the contract workflow patterns are also presented.

Keywords

Ontology-Conceptual models, Business Process Modeling, BPMN

1. Introduction

With the onset of e-commerce, new trade relationships are forged with partners across the globe. The contract is a blueprint of the commitments undertaken and the expected behavior of the partners. Contractual terms need to be aligned with their respective internal business processes. Business process modelers have to draw mostly on their experience and domain knowledge while designing workflow models for enterprises. However, interpretation of legal contractual obligations and their expected fulfillment patterns is not easy. We have proposed a common shared view of the three different domains [1], viz: legal contracts, business process and information systems (process modelers in this case). We visualize the representation of such a shared view as a reusable ontology.

We have proposed the use of Multi-Tier Contract Ontology (MTCO) [1] to model the above mentioned shared conceptual views. The MTCO models the main concepts involved in a contract like C*onsideration Actors, Roles, Obligation and PerformanceEvents.* MTCO also analyses and presents the different *obligations* and the dynamic state-changes that occur in response to activities carried out by the parties involved. Thus each execution cycle of a contract may be visualized as a *contract workflow*. In this paper we use the MTCO to deduce high-level contract compliant workflow model named Contract Workflow Model (CWM) [17]. The CWM is primarily visualized as an aid to the business process modeler in his attempt to design the internal

business process models for the business entity. As such, the CWM may be drawn as a simplistic sketch for human understanding and not requiring any specific formal modeling specifications. At the same time, the same CWM may be formalized into a machine – readable format. We chose BPMN as one of the probable formal representation notations. To aid the process modeler further, we propose some common patterns that are useful in sketching the CWM as a formal CWM-BPMN model. In this paper we motivate and present some CWM-BPMN representational patterns using BPMN [11] notation. The rest of the paper is structured as follows: In section 2 we discuss some of the related research relevant to our work. In section 3, we proceed with a description of our approach methodology, followed by a discussion on our choice of BPMN as a representation language for the CWM in section 4. In section 5, we present some CWM to BPMN transformation patterns. We discuss some open issues to be resolved in section 6, and thereafter conclude in section 7.

2. Related Research and Background

Business contracts have been viewed from different perspectives or components by various researchers [3, 4, 8, and 9] including:

- Document Centric: Contracts are handled as paper documents or in cases of e-contracting electronic annotated files (XML documents, ex. TPA (Trading Partner Agreement in ebXML)
- Data Centric: most traditional contract management applications extract the information as data to be merged in to other ERP information systems.
- Procedural: a contract defines the *choreography* of the parties' actions.
- Communicative: as a set of speech acts wherein the parties *declare, permit, prohibit, or promise* to carry out certain set of activities in exchange for some commiseration.
- Normative: contracts are governed by legislation, regulations and standards specifying pre-described course of actions.

We view the contracts from a combination of the above perspectives as discussed in section 3. Additionally, as proposed by Daskalopulu and Sergot in [8], we agree that all contracts, especially business contracts have both *explicit terms* as well as *implicit* terms included in them. For example, a standard delivery term, from the Incoterms, like EXW (Ex-Works), is included merely by its abbreviation and the parties to the contract are expected to understand its implication and the expected behavior. Thus, it is vital that all explicit, implicit knowledge from the contract is made explicit to the business process modeler. However, most business process modelers or information systems domain experts can hardly be expected to be lawyers or juridical experts. Neither can they be expected to be AI domain experts in Subjective Logic or Deontic Logic [7, 3]. Hence our aim in designing the common knowledge base, MTCO as well as the deduction of the CWM based on the MTCO and the individual contract instance, has been to represent the shared perspective of the contract, business practices and information systems domain, in a simple, re-usable and human understandable format. Thus, the MTCO is essentially a set of conceptual models, which have also been implemented in ontology languages like RDFS and DAML+OIL.

Daskalopulu and Maibaum [10] represent a contract in terms of its obligations, rights, powers, and other legal relations. They model a contract 'as a process whose state at a given time is determined by the legal relations that obtain between parties. This supports our theory of distinguishing the obligations as having a cycle of different states through their execution life cycle, as proposed by us in [2].

Karlapalam [5] has viewed the contracts as workflow Meta model. Similarly Huevel and Weigand have expressed the contract as cross-organizational workflow [6]. But, we have primarily based our CWM on Van der Aalst's workflow patterns [18, 19]. Our ontology modeling approach methodology has been based on Methontology as proposed by Gomez Perez, Fernandez et al [12, 13], who have used the same for modeling chemical ontology [14] as well as legal ontology [15].

3. Approach

We have proposed a layered architecture for capturing contract knowledge in MTCO. At present, we have three-tiered architecture which may be extended horizontally as well as vertically. We begin with a top, generic Upper Level Contract Ontology; followed by a contract type oriented Specific Contract Type Ontology and finally a predefined Template Level Contract Ontology.

We adopt an approach from the METHONTOLOGY [13], which proposes a bi-phase method for modeling ontology [12]. The approach proposes a two phase coupled with a two level of modeling, one on the conceptualizing phase and on the modeling of the domain ontology phase. The MTCO are UML conceptual models. In [16] we have discussed the suitability of UML as ontology modeling language. The other focus is on the process aspects or execution aspects, viz, the obligation and performances, as modeled by the CWM. A step-wise method for using the MTCO to deduce the CWM has been proposed [17]. As said earlier, the CWM may be a simple flow sketch or a formal high level partial contract compliant workflow model and can be formalized using BPMN. The MTCO itself can be implemented using RDFS, DAML+OIL or OWL. The CWM-BPMN model may be mapped to executable BPEL4WS language.

We have in [1] demonstrated that the UML conceptual models may be implemented in RDFS, DAML and currently we are working on OWL implementation using Protégé tool. Though the same may also be modeled using EPC (Event Process Chain Diagrams), UML activity diagrams, we chose BPMN for the reasons mentioned in section 4.

Before we present our motivation for choosing BPMN as workflow representation language, we briefly summarize our analysis of contract obligations, in order to provide a complete background to our CWM modeling methodology.

We have proposed a contract as consisting of different types of 'promises' or 'commitments' undertaken, which are called as '*Obligations*'. Obligations are *fulfilledBy* the execution of related business processes or events called as *PerfomanceEvents*. Obligations may be of the following types: Primary, Secondary, Reciprocal, and Reconciliatory (for details see [2]). We further analyze each obligation as undergoing the following states (based on Thoen and Tan's work [3]):Inactive, Active, Triggered (fulfillment triggered), Fulfilled, Terminated, and Pending (in case some compensation

has been set in motion, then the initiating obligation gets fulfilled only after the reconciliatory obligation is fulfilled). The obligations undergo a state change in response to either a message or event.

We summarize our motivation to express the contractual knowledge in terms of a *contract workflow model* as:

> To incorporate the *procedural*, the *behavioral*, *communicative*, *informational* aspects from the contract, with the existing internal business process models, functions, rules and policies.

4. Choosing BPMN

The choice of BPMN as the formalization language was based on the several features of BPMN. We discuss the advantages and disadvantages of BPMN below:

- BPMN is more expressive than other similar languages like UML activity diagrams, UML Sequence Diagrams, formal colored Petri nets. It is especially suitable for the contract –business process domain, as it gives us the flexibility to capture the domain specific semantics involved. BPMN uses two levels of information representation. On the first level, the graphical notations are simple to understand. On the second level, each BPMN construct defines a set of attributes that can be used to convey a richer specification. In case of CWM, we have used user-defined attributes to model the specific characteristics like individual *ObligationState* of each *Obligation*. In this regard, we found the expressive capacity of BPMN to be advantageous over others like UML activity diagrams.
- BPMN is graphical in nature. Thus affords easy understandability to both business process modelers, as well as domain experts including lawyers, information system experts and business management and decision makers. At the same time, the set of non-graphical attributes render BPMN as a powerful notation to map to Business Process execution languages.
- BPMN is appropriate to model abstract high-level 'black box' processes, collaboration patterns and sequence diagrams, all in one single Business Process Diagram (BPD).
- BPMN allows the different parties to have different perspectives of the entire BPMN diagram. The CWM expressed as a high-level abstraction gives all parties involved an idea of how each role player would act and their overview of the process execution. At the same time, low-level business process details may be encapsulated within the same abstract high level CWM. Each role may have a different view of the entire CWM. For example, the buyer needs to know that his counterpart, the seller would 'make goods' in response to his 'send PO' task. But the buyer does not see exactly how the seller 'makes his goods'. Whereas, the seller should be able to drill down to see his internal business activities which may include, 'get supplies from the supplier', 'order extra workers to assemble the goods', 'pay for supplies '.
- BPMN can be mapped to a number of low-level specification languages (machine executable) like BPEL4WS, RosettaNet, ebXML BPSS etc. Thus, the graphical CWM can be made operational.

- The graphical elements of BPMN can be extended to adapt for domain specific purposes. Though BPMN restricts addition of new core elements, it allows the modeler to add user-defined attributes, change format and layout specification (except a few reserved 'keywords' like layout specifications). For example: the three types of events, start, intermediate and end event defined can be defined to have different internal markers to specify additional information. Some basic types like Message, Rule, Link, Compensation, Error, Timer, have already been pre-defined. Additional markers may be specified, like for each O*bligationState* we can have a different notation. In this current document, we have used the existing markers of timer and message to denote the different obligation states. The definition of specific markers is under research.

5. CWM-BPMN Transformation Patterns

The CWM patterns are based on / application of the Workflow Patterns as proposed by Van der Aalst, Hofstede, Kiepuszewski, Barros etal [18,19]. The other theoretical background is the BPMN specification. Since, these workflow patterns have been specialized for the contract and business process domain; the basic patterns as suggested by Van der Aalst have been reified to specific cases for application in this paper.

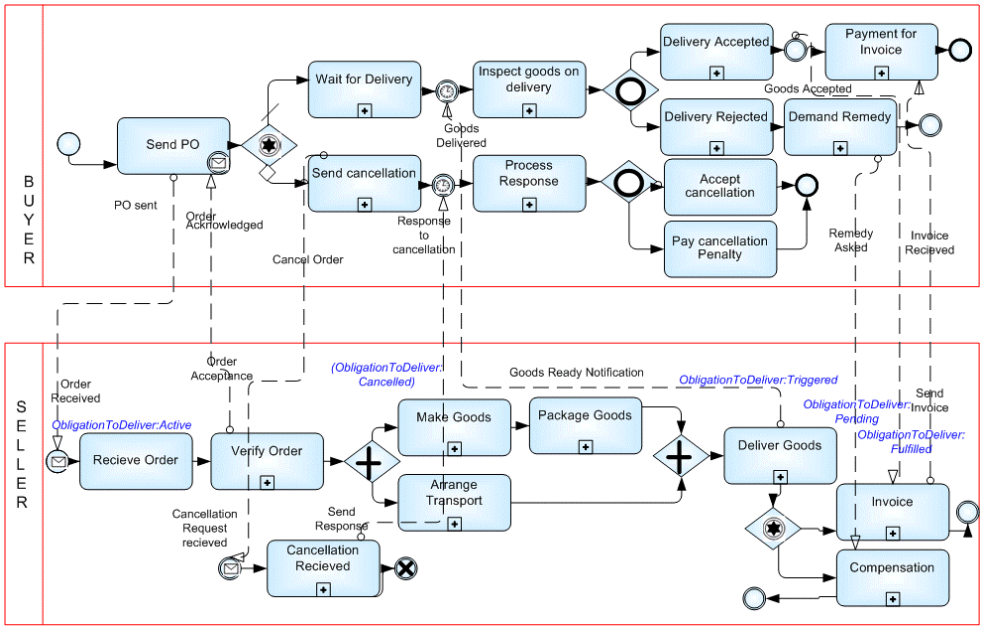

Figure 1 Sample CWM for a typical sale of goods contract scenario.

The main inputs to the process modeler are the contract knowledge base, MTCO, the actual contract instance, the set of CWM-BPMN transformation patterns and

optionally the existing internal business process model. It need not be a machine-executable business process model, but simply a source of information regarding the usual manner in which similar business activities are executed by the concerned business enterprise. The objective is to use business process or activity terminology so as to identify the *performance events* in the CWM as close as possible to the ones currently in practice within the enterprise.

We explain our CWM patterns with reference to a typical business contract scenario between a seller and buyer for a Sale-Purchase contract situation. Figure 1 [20] above presents an extract from the final CWM deduced for a typical purchase and sale scenario.

A detailed analysis of such a situation and the methodology for deducing the CWM has been discussed in [17]. A brief summary of the CWM methodology is as follows:

- Meta data from the contract instance is identified with the help of the main concepts in MTCO. The specific contract type of the contract is also identified.
- The contract type identification provides the process modeler with the common obligations for the parties involved. For example for purchase and sale of goods contract, the primary obligation of the buyer is to take delivery and pay for the goods ordered, whereas that of the seller is to deliver goods in conformance to the agreed contract.
- The MTCO also provides the information regarding the expected performance for each of the obligations. Other information like the possible liabilities, warranties and courses of action in case of dispute or non-compliance are also extracted.
- The extracted obligations and their performances are grouped by the performers responsible for executing them and thereafter arranged in a *time ordered sequence.* This choreography obtained is the CWM.

In all the following CWM-Patterns presented in section 5, we use the same example as a running case study. The process modeler may choose to formalize the CWM further or use the same for contract compliance, performance-monitoring purposes. When no previous process models exist then the CWM itself is the high-level business process model. The CWM deduced from the contract instance and the MTCO may use concepts or terms from related ontologies like the enterprise ontology, business process management ontology. In which case, the natural language interpretation of the performance events is replaced by precise business activities or process names. In case there is a pre-existing internal business process model then the task of the process modeler is to compare the deduced CWM and the pre-existing business process models to check for contract compliance. The CWM provides the boundary conditions within which the existing business process model can be allowed to vary and yet be conformant to the contract. In case the existing process model violates the CWM then the business enterprise runs the risk of contract violation.

When no business process models exist then the process modeler needs to formalize the high-level CWM in to an executable business process model. The CWM-BPMN patterns provided below are an aid in this direction.

5.1 Pattern 1: Contract -Business Process

Description: The activities of each partner are modeled as a sequence of processes or a workflow within one swim lane each in a single pool. In case, a detailed level of modeling showing all sub processes is required, then multiple swim lanes for each partner /actor/role may be used. It is thus recommended that two different levels of abstraction are maintained for clarity and easy understanding.

Examples: The seller's *Performance Events* are modeled as a sequence of business processes in a single swim lane, whose name is represented as the *role.* Similarly another Lane is attributed to the buyer, one to the carrier and so on. The entire CWM consists of this set of inter related swim lanes and is equivalent to a pool in BPD. **BPMN mapping: Swim lane, Pool**

5.2 Pattern 2: Performance Events

Description: The performance events and non-performance events from the MTCO are mapped to Activity, Process or Sub-process depending upon the level of abstraction used. Usually at the CWM, only partial information is available, so a modeler can infer only processes or sub processes. Activity may be identified at the time of low-level internal business process model description or when mapping to pre-existing process models.

Examples: In the case scenario illustrated in fig 1, examples of PerformanceEvents associated with the buyer are *Send PO, Inspect Goods*, *Send Cancellation* etc. Similarly those of the seller are V*erifyOrder, Make Goods, and Package goods*. In this example, we see that *send PO* is an identified task, hence modeled as a BPMN Activity, whereas, *Verify Order* involves internal order verification process and thus modeled as a BPMN collapsed Process. It has been used as a black box to hold the internal workings of the seller.
BPMN mapping: Activity, Process, and Sub Process

5.3 Pattern 3: Obligation State Changes

Description: The contract- business processes between the two parties have obligations that are to be fulfilled by the execution of activities. The activities themselves are represented using **activity, process or sub-process.** The obligations are modeled as *events* in BPMN. The initiation of these sequences of activities is represented by a **start event.** The start of a cycle of contract execution may be triggered by some external activity like the other party sending a purchase order. This dependence on other **external** factors for the start (triggering) of the contract obligation fulfillment is represented by **message start event.** But the counter party expects certain response activity. Thus, this dependence of external factors should be modeled as **intermediate message event**. In case there is a certain predetermined time out period, then we should model the condition as a **timer.** In case, when some internal business policy or rule triggers a process, use **rule start/intermediate event.** The process modeler may make use of the other attributes of the event types to capture specific aspects of the obligations (from the MTCO) like the

obligation state name, obligation type, the obligation owner, obligation ownee (the performer, or the role for the swim lane would be the obligation ownee)

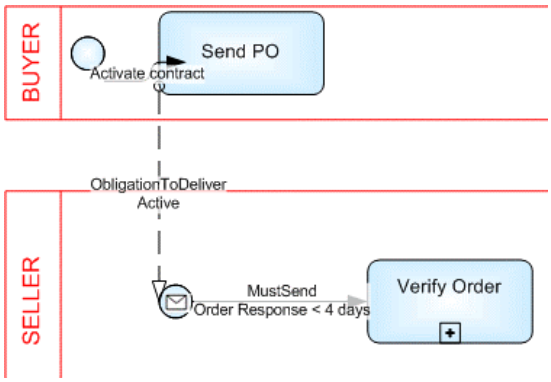

Figure 2: Closer look for modeling obligation state changes

Examples: The arrival of a purchase order from a Buyer initiates the primary obligation of the seller to *deliver goods*. Since the entire process is triggered by buyer's activity, we model the input as a Message Start. Figure 2 takes a closer look at the above scenario.
BPMN mapping: start event, intermediate event, and End event

5.4 Pattern 4: Performance Events Sequences

Description: Choreography of activities/ processes is represented as a sequence. The performance events are arranged in a time-ordered sequence. The general flow of events is depicted by BPMN sequence flows.

Examples: the seller on order receipt is obliged to send an acceptance to the buyer. Inactivity implies acceptance. But to do so, he MUST verify the order, and maybe check his production schedule, or his supplies. **BPMN mapping: sequence flow.**

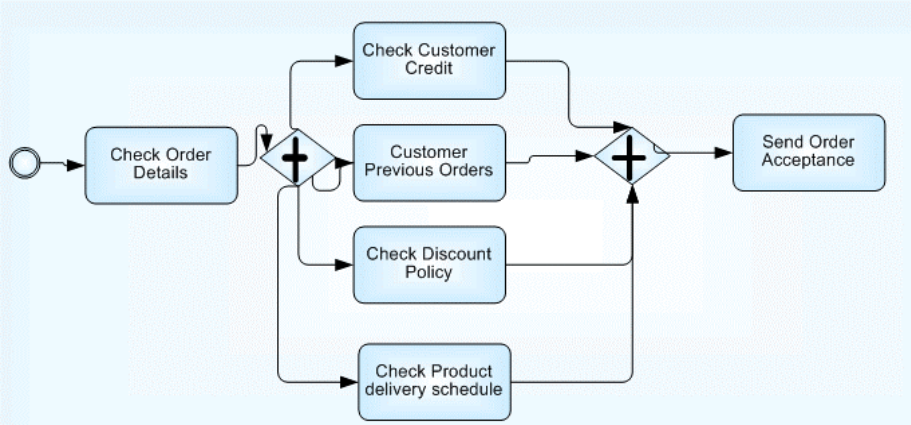

Figure 3: Illustration for performance event sequences and simultaneous processing

5.5 Pattern 5: Simultaneous Processing

Description: An AND gateway is used to describe simultaneous processing of several sub tasks or processes in parallel or to synchronize other activities.

Examples: the seller makes *goods* at the same time he also *arranges for carrier,* and procures *packaging material.* The seller would wait for all the three parallel processes to be finished before he *notifies buyer* that the goods are ready. (See fig 3 above, an expanded version of the *Verify Order* sub process shown in Figure 1)
BPMN mapping: AND Gateway

5.6 Pattern 6: Communication between the Parties (Performers)

Description: The time-ordered sequence of performance events is grouped *Performer-wise* in separate Swim lanes. The communication between the two is modeled using Message Flow connectors.
Examples: The buyer sends an order to his seller. In the buyer's swim lane this is represented as 'send order' process. From the MTCO, we deduce that the beneficiary of this performance event is the seller. Thus the communication between the two is modeled using a message flow.
BPMN mapping: message flow

5.7 Pattern 7: Exclusive Processing

Description: An XOR gateway is used to describe situations when the actor has to choose ONE from a set of available options. A conditional outflow may be modeled using the **conditionExpression** attribute of the out-going sequence flow.
BPMN mapping: Event based Gateway

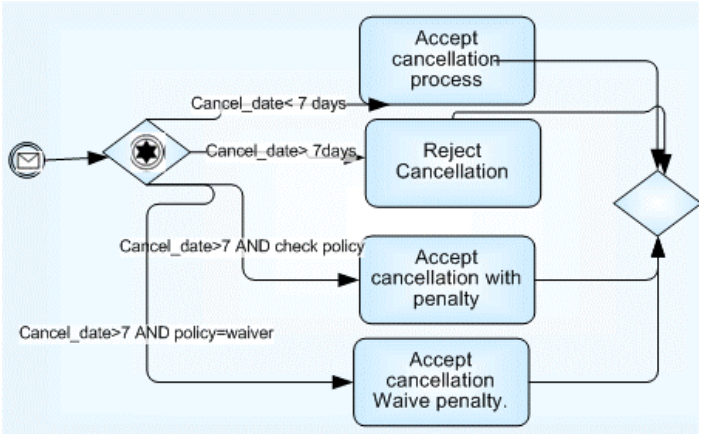

Figure 4: Illustration for Event –based decisions

Examples: The buyer is entitled to cancel his order within 7 days from the date on which the seller received his order. So in case he sends his cancellation, then the seller is faced with a decision to make.

- If the cancellation came within 7 days from the order date, then the seller has to accept the cancellation unconditionally. This is the default outgoing sequence flow, whose condition Expression is set to a string expression of 'cancel_date<=(order_date + 7)'
- The cancellation is received after 7 days; the seller may reject the cancellation. That is the buyer owes the seller entire payment for the ordered goods irrespective of whether the buyer takes the delivery or not.
- The cancellation is received after 7 days and the seller has started to make goods. But the seller may choose to accept the cancellation with a penalty fee applied.
- If the cancellation came after 7 days, but the seller has not yet begun production or for some internal reasons has not acted on the order, then he may choose to accept the cancellation, though he was entitled to reject the cancellation

As seen above, this scenario may be modeled as an event-based gateway with conditional expressions associated with the sequence flows (fig 4).

5.8 Pattern 8: Remedial Options Choice

Description: In case of deviations, one of the business partners usually has a number of remedial options (rights, permissions) available to him. The choice is complex and usually involves human decision-making. We model such options using complex decision gateways.

BPMN mapping: COMPLEX decision gateway

6. Open Issues and Future Work

In the above sections we have seen that BPMN is suitable as a process representation language for formalizing CWM in conjunction with our proposed MTCO, specifically for the domain of business contracts and business process models. However, there still remain several open issues, which are the subject of our on going research:

- Though we have analyzed and adopted several patterns for translation from conceptual models to BPMN, the list is not exhaustive or comprehensive for all possible contractual scenarios. Further work is needed to capture the semantics of different contract types, and standards. We have started with modeling Incoterms as a series of CWM-BPMN patterns. Similar work with other contract standards is required.
- CWM-BPMN model is useful to capture the semantics of the contract execution and performance fulfillment, but cannot capture all informational aspects and related concepts like *prohibitions, rights, liabilities* etc, which in turn are captured and represented by the UML conceptual models in MTCO.

7. Conclusion

The main contributions of this paper are:

- A discussion on suitability of BPMN for formalization of conceptual models into business process models.
- Some of the CWM-BPMN transformation patterns which illustrate the use of BPMN to capture the semantics of contract obligations and their execution through business processes.

In this paper, we have proposed the CWM-BPMN patterns as an aid for the process modeler to deduce the internal business process model in the absence of pre-existing business process models. In which case, the high level CWM is the starting point for the design of a contract compliant business process model. A methodology to aid the process modeler in mapping the CWM to pre-existing internal business process model is currently under research. The CWM in that case is intended for setting the boundary parameters within which the pre-existing business process model may execute and still be compliant to the contractual terms and conditions. The CWM to internal business process models and the mapping rules for the *obligation states* to appropriate private or public business activities and their corresponding state changes are the subject matter of another paper.

We conclude this paper by summarizing that our objective to model the behavioral, procedural, communicative and informational aspects of the contract has been achieved partially through the MTCO knowledge base (informational, communicative), and partially through the deduced CWM (behavioral and procedural). We further foresee, that the contract performance can be measured by comparing the CWM as intended from the contract, to the CWM of the actual execution of the business transaction.

References

1. Kabilan, V. Johannesson, P. Semantic Representation of Contract Knowledge using Multi-Tier Contract Ontology, Published in the proceedings of Semantic Web and Databases workshop, (SWDB 2003)
2. Kabilan, V. Johannesson, P. Rugaimukammu, D. Business Contract Obligation Monitoring through use of Multi-Tier Contract Ontology. Proceedings of Workshop on Regulatory Ontologies (Worm CoRe 2003), November 2003, Italy. Springer-Verlag publications (Lecture Notes in Computer Science)
3. Tan, Yao-Hua. Thoen, W. Using Event Semantics for Modeling Contracts. Proceedings of 35th Hawaii International Conference on System Sciences –2002.
4. Tan, Yao-Hua. Modeling Directed Obligations and permission in Trade Contracts.31st Annual Hawaii International Conference on System Sciences, 1998
5. Karlapalem, K. Dani, A R. and Radha Krishna; A frame Work for Modeling Electronic Contracts; *Proceedings of Entity Relationship and Conceptual Modeling ER 2001, LNCS 2224 pp 193 – 207*
6. Heuvel, W van den. Weigand, H. Cross Organizational Workflow Integration using Contracts, *Business Object Component workshop, OOPSLA* 2000.
7. Daskalopulu, A. Logic Based Tools for Legal Contract Drafting: Prospects and Problems. *Proceedings of the 1st Logic Symposium 1997. University of Cyprus Press, pp.213-222*
8. Daskalopulu, A & MJ Sergot (1997). The Representation of Legal Contracts, *AI and Society 11(Nos. ½), pp 6-17*

9. Daskalopulu, A. Evidence Based Electronic Contract Performance Monitoring. *The INFORMS Journal of Group Decision and Negotiation*. Special Issue on Formal Modeling in E-Commerce, 2002.
10. Daskalopulu A. & Maibaum T. S. E. (2001). Towards Electronic Contract Performance. *Legal Information Systems Applications*, 12th International Conference and Workshop on Database and Expert Systems Applications, IEEE C. S. Press, pp. 771
11. Business Process Management Notation version 1.0. Available online at www.bpmn.org. Last accessed on April 15th 2005.
12. M Fernandez Lopez. .Meta-Modeling for ontology development and knowledge exchange, Proceedings of the ECAI-02 Workshop on Ontologies and Semantic Interoperability Lyon, July 22, 2002.
13. Fernández-López M, Gómez-Pérez A, and Juristo N (1997) *METHONTOLOGY: From Ontological Art Towards Ontological Engineering*. Spring Symposium on Ontological Engineering of AAAI. Stanford University, California, pp 33–40
14. Fernández-López M, Gómez-Pérez A, Pazos A, Pazos J (1999) *Building a Chemical Ontology Using Methontology and the Ontology Design Environment*. IEEE Intelligent Systems & their applications 4(1):37–46
15. Corcho O, Fernández-López M, Gómez-Pérez A, López-Cima A *Building Legal Ontologies with Methontology and WEBODE. Law and the Semantic Web*. *Legal Ontologies, Methodologies, Legal Information Retrieval, and Applications*. March 2005. Springer-Verlag, LNAI 3369.
16. Kabilan V, Johannesson P. *UML for Ontology Modeling and Interoperability.* First International Workshop on Enterprise Modeling and Interoperability (INTEROP-EMOI 2004) collocated with Caise 2004, Riga. Proceedings available at CEUR workshop proceedings ISSN-1613-0073 , Vol 125
17. Kabilan V, Dzdrakovic J, Johannesson P. *Use of Multi-Tier Contract Ontology to deduce Contract Workflow Models for Enterprise Interoperability*. Proceedings of 2nd INTEROP-EMOI open workshop on Enterprise Models and Interoperability collocated with CAISE 2005, Porto.
18. W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. *Workflow Patterns,* BETA Working Paper Series, WP 47, Eindhoven University of Technology, Eindhoven, 2000.
19. W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. *Advanced Workflow Patterns.* In O. Etzion and P. Scheuermann, editors, *7th International Conference on Cooperative Information Systems (CoopIS 2000)*, volume 1901 of *Lecture Notes in Computer Science*, pages 18-29. Springer-Verlag, Berlin, 2000.
20. Figure 1. An enlarged version of the same and all other figures and examples used is available online at CWM to BPMN example scenario. Or may be requested from the author : Vandana@dsv.su.se