Using Multi Tier Contract Ontology to Deduce Contract Workflow Models for Enterprise Process Interoperability

Vandana Kabilan [1] Jelena Zdravkovic[2] Paul Johannesson[1]
[1]Department Of Computer and System Sciences
Royal Institute of Technology and Stockholm University,
FORUM 100, Kista, SE 164 40, Sweden
E Mail: {vandana, pajo}@dsv.su.se
[2]University Of Gävle and Royal Institute of Technology
Kungsbäcksvägen 47
Gävle, SE 80 146
E Mail : Jzc@dsv.su.se

Abstract.

Ontologies are being proposed as a medium for affecting enterprise application integration. Though it is widely accepted that ontologies can support inter-enterprise interoperability, the exact nature and extent to which ontology may be useful is uncertain. We promote the use of ontology in a two-fold way: first, as a knowledge base for fostering human-to-human shared understanding; second, as 'Interlingua' for promoting human-to-machine as well as semantics-to-execution specification. The proposed concept is described using a case scenario in the realm of legal business contracts, followed by their integration to the business domain, with an objective to model contract compliant business process models. With the case, we illustrate the use of Multi-Tier Contract Ontology (MTCO) to deduce a high level, partial business process model named the Contract Workflow Model (CWM). Such a model, from the business process perspective, may be used as a skeleton for designing internal business processes for each individual contracting party, or for mapping to existing processes.

1. Introduction

Business-to-Business (B2B) interoperability may be of different types, and defined at different levels. At the basic level, each of the business enterprises requires a shared, clear understanding of each other.

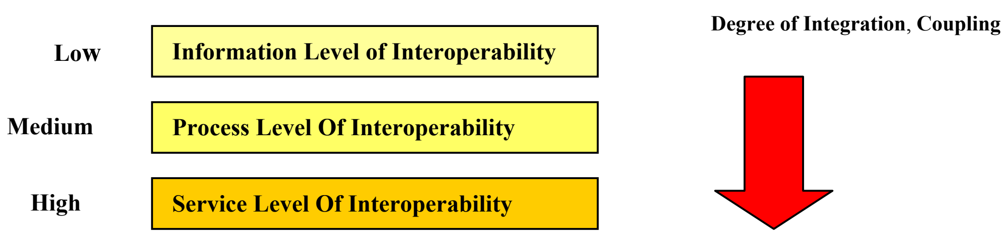

Fig. 1. Different Levels of Interoperability

We propose that enterprise interoperability may be achieved at three different levels, based on the classifications of enterprise integration patterns as proposed in [14] as seen fig 1.

At the lowest level of integration are the stand-alone enterprise applications that are integrated via simple export import of data or information. Small-scale entrepreneurs generally adopt this, or those lacking highly evolved infrastructure. At the medium level, enterprises coordinate common business process choreography, in order to get a shared understanding of their interactions. The third is the service level interoperability in which enterprises define their common interfaces for 'services' using industry standards like EDI, ebXML to carry out electronic commerce. By following this model, process interoperability need not only imply tightly coupled B2B architectures – instead, it may be anywhere ranging from simple data exchange to service interoperability.

Our current research is concentrated towards process level interoperability and the use of ontologies in that context. For that purpose, an understanding of each other's processes is needed (human-to-human knowledge transfer followed by a human-to-machine knowledge transfer). Based on discovered requirements for process interoperability the business process modelers may design or adapt the internal business processes. Ontology fulfills the above criteria sufficiently. In this work, we explore the role of ontology in supporting process interoperability, through evaluation of different domain case scenarios.

We focus the case scenarios to the domain of legal business contracts that governs the business process domain, and, in turn, the information systems (enterprise application) domain. A traditional contract management application considers the retrieval, storage, searching of contract data (information level interoperability). Electronic contracting considers agent supported (or e-market controlled), automated contract offer followed by the negotiation process and the subsequent execution monitoring by agents [5]. E contracting is a type of service level interoperability between enterprises. In contrast to that, our work explores the contract knowledge representation and the contract obligation monitoring from the perspective of process level interoperability. The proposed methodology for deducing CWM based on the MTCO may be, if required, transformed to machine-readable versions for supporting service level interoperability. This means that, starting from the contract perspective we deduce a business workflow model, which satisfies all the conditions stated within the contract.

Being derived for a business contract, the Contract Workflow Model models all business logics and interactions found in the contract. This means that the CWM comprises both "regular" courses of action and a variety of alternative, as well as exceptional conditions. A CWM, thus, deduced in a step-wised manner to the form of a high-level partial process model, provides the business process modeler with enough information to adapt or create a business process compliant to the signed contract. Using the CWM, the process modeler makes use of the MTCO in addition to his existing business domain knowledge as a reference to help him easily understand and pinpoint the *obligations* and the expected *performance* for *fulfilling* the legal *obligations.* Alternatively, the MTCO may also be mapped to a relevant enterprise or business process ontology like the Open Source Business Management Ontology [20]. In which case the CWM provides more specific details pertinent to the business process models and may be directly used to map to internal business process models. The CWM is a blend of both public and private interfaces of two or more enterprises involved in the contractual relationship. As such, the CWM is seen as an efficient methodology in ensuring process level interoperability between enterprises. Once the CWM is mapped tightly to the existing internal business process models then the third, service level interoperability may be achieved.

This paper makes use of the BPMN [12] for representing the CWM, though the same may be represented using other notations as well. Our motivation for choosing BPMN has been discussed in [21]. Thus, a CWM represented using BPMN may be visualized as a partial process model that can be mapped to an existing business process (given also in the BPMN form), or in the case when those processes do not exist, translated directly into a BPEL4WS skeleton.

The paper is structured as follows. In section 2 we present a discussion of related works, followed by a brief overview of the contract knowledge base, the MTCO in section 3. In section 4, we present the CWM methodology. In section 5, we apply the CWM methodology on a sample business contract to derive a CWM for the same contract. In section 6, we discuss the possibility of mapping the deduced CWM to existing business processes. Finally we review our results and conclude in section 7.

2. Related Research

Increasing use of ontological engineering in the realm of business process engineering has been supported by Deborah McGuinness [3] who identifies the goals for ontological development and proposes guidelines for a centralized knowledge base to capture semantics from domain, standard terms and vocabularies. D Linthicum has advocated the use of ontologies for application integration [17].

Business contracts have been viewed from different perspectives and purposes by various researchers like e-contracting [4, 7], contract performance monitoring [5], business process exception handling [10] etc.

We model *obligation states* to capture the information relevant to that obligation as the fulfillment is being executed. Relationship between *obligations, obligation states and performance events* forms the foundation for the proposed CWM methodology. Our identification of the different *obligation states* of 'active, triggered, pending, fulfilled, cancelled' is based on Yao-Hua Tan's work in [9]. The SweetDeal project [8,19] is closest to our MTCO and CWM in that they try to capture the semantics of different types of contract exceptions. They also use the MIT process ontology that has been proposed by [14] for e-contract exception handling as well as business process modeling approach.

The problem of enactment of business contracts using workflow processes has been studied in many works [11], [15], [16]. Our work differs in two aspects. First, we use ontologies as the basis for mapping the concepts from business contracts to those existing in the business process domain. Second, we deduce the contract requirements, again based on the shared domain ontologies, to the form of partial process models that then can be directly mapped to the existing business processes. Ontologies are the backbone for our work in the realm of fostering enterprise interoperability.

3. Multi Tier Contract Ontology

We support Daskalopulu et al [1] in their identification of the following roles of the contractual provisions in that contracts *define* terms. They *prescribe* certain behavior

for the parties, under set conditions for a certain period of time. Contracts specify *procedure* that needs to be followed. For example, the delivery terms inform the buyer regarding the time limit by which he has to inform the seller regarding his transporter, or any change in venue for the delivery, or delivery date etc. Also they contain *formulae* that are used to calculate various parameters, for example, the cost adjustments to price or quantity depending upon fluctuation of currency, changing duties, tax etc.
We also support [1, 5] and [9] in their analysis of contracts from various perspectives like, a contract is an organized collection of concepts, obligations, permissions, entitlements, powers etc. A contract has also been viewed as a collection of procedures or protocols that specify its operational aspects (how the business exchange is to be carried out in reality) or simply parameters (the parties, product, price, delivery quantity, delivery date).

The MTCO is a combination of the above aspects as has been presented in [2]. We have chosen UML conceptual models as optimum method of knowledge capture and representation [18]. Currently, we have identified a minimum of three different layers as presented below. A brief summary is also presented below:

- **Upper Level Core Contract Ontology** represents a general composition of a contract, which may be applicable across most of the prevalent types of contracts.
- **Specific Domain Level Contract Ontology** is a collection of various types of contract. Each of the contract type ontology represents a specific contract type like property lease rental, employment contract, and sale of goods amongst others.
- **Template Level Contract Ontology,** consists of a collection of template like definitions for established or recommended contract models like the International Chamber of Commerce's contract model for International Sale of Goods, European Union's SIMAP online procurement contract models etc.

Other extensions and layers may be possible like a fourth instance layer containing the instantiation of every contract model each time it is executed. Or a horizontal extension, by mapping to other relevant ontologies or ontology architectures like the MIT process ontology as proposed by Dellarocas [10].

3.1. Multi Tier Contract Ontology: Main Features

We have abstract high-level concepts defined at the Upper Core Level Contract ontology of the MTCO. For example, the UCLC Ontology defines the generic concepts and semantic relationships between contracts, its participants the *actors*, *the roles* they undertake within the scope of the contract, the object for which they undertake the contract. *Consideration,* the commitments they make, *obligations,* the expected business actions that will *fulfill* the obligations, *performance.*

The Sale of Goods Contract is an example for the specific *Domain Level Contract Ontology* and has the typical roles of buyer, seller, and banker. The actors would be the *business entities, or persons* who undertake the contract. The consideration is usually some *product* or object of some economic value for the buyer, for which he is prepared to pay money or some other compensation in return, which is of value to the seller. The buyer has a primary obligation, *ObligationToPay*, to pay for the goods he has ordered, whereas the seller has a primary obligation to make and deliver goods as agreed in the contract. There are other nested secondary obligations (which may be similar to the non-directed obligations as identified by Yao-Tan [6]), like the seller is obliged to pack (*ObligationToPack*) the goods suitable for transportation and delivery.

The above description of procedural knowledge and strategic knowledge is presented through the choreography of obligations and performance events in the contract, is then modeled as a set of Contract Workflow Models (CWM) (Section 4). In [16], we have presented a detailed discussion on our analysis of the *obligations* and a classification of the different *obligation states* has been presented.

4. Contract Workflow Model

Multi tier Contract Ontology provides information regarding the obligations and their expected performance activities. This expected choreography of business actions is modeled as a *Contract Workflow Model*. We define a CWM as:

A partial choreography of performance events for each of the parties concerned as deduced from the perspective of the governing business contract and is an indicative model for the contract compliant business process model.

In the following section we discuss some uses for the deduced CWM and its suitability for promoting process level interoperability.

4.1. Uses for CWM

Some of the proposed uses for a CWM deduced from a signed contract may be listed as:

Primarily, the CWM is intended as a high level conceptual model useful for the top management and decision makers in any enterprise to understand the contract compliant requirements, the various options available to them in case of any contract violation, the possible repercussions they could face in case of non –performance.

Secondly, the CWM may be for identifying the common points for mapping or communicating between the two individual business workflows. For example, identifying the corresponding performance events on each of the parties' CWM, tells the parties when to expect some activity or information from their partner and how to respond to such activities. Thus, even in the absence of a tightly coupled system like a B2B transaction, the CWM is instrumental in facilitating a process level interoperability. The Seller and Buyer can identify their mutual interdependencies and processes for interaction.

The CWM may be used to map to their internal business process models. It is impractical to expect that an enterprise shall modify their existing business process models to be completely compliant to their signed contracts for each contract and for each of their umpteen partners. Thus, we propose a more realistic approach in which the enterprises map their contract compliant CWM to their existing business process models and note the extent of deviations from the deduced CWM. As long as the execution of the business process is within the acceptable limits the enterprises need not make any changes to the existing business process models. In the current work, we have assumed that the contract has been agreed and signed only after careful negotiation and verification, and thus the deduced CWM shall not be widely dissimilar to the existing business process models.

In case there are no previously existing business process models, then the CWM is the starting point for the process modeler to elaborate as the business process model required.

Every time the contract is executed, we may deduce a CWM for each contract execution. Thereafter, we may compare the actual CWM instances with the CWM inferred in the beginning. Thus, a degree of contract performance and monitoring may also be achieved. Details and methodology for this aspect is not presented in this paper.

In the next section, we present our methodology in the form of a stepwise guideline, to deduce the CWM from a given instance of a contract and using our proposed ontology knowledge base, the MTCO.

4.2. Guidelines for deducing CWM

The following methodology for deducing the CWM is in the form of guidelines for a user, having the contract instance, the MTCO for reference as well as the internal business process working knowledge (optionally). We refer to other business process models, or other related ontologies like TOVE ontology, MIT Process Handbook Ontology [15] for providing the conceptual definitions and terms for representing the performance events. First we present an overview of the guidelines and in the next section we use the same to walkthrough a sample business contract to deduce a CWM.

Phase 1: Contract Type Identification: The first phase enables the user to identify the contract type to which the particular contract instance belongs. This is done by guiding the user to identify the principal components from the upper level core contract ontology and then moving on to match the specifics to specific domain level contract ontology.

Phase 2: Contract Instance Meta-data Extraction: Once the user identifies the specific contract type to which the contract belongs to, he identifies all the major *object* components of the contract type with respect to the contract instance given. This leads to the identification of pertinent Meta-data and information available in the contract instance. At the end of phase 2 the user should be able to draw an object diagram for his contract if so desired. It would suffice for the novice user to come up with the list of identified objects and Meta-data.

Phase 3: Obligation, Performance Events Identification: Identify the process oriented *obligations*, *performance events*, non-performance events, rights *objects* from the identified list of *objects/Meta-data.*

Phase 4: Obligation, Performance, Non-Performance Inter-relationships and Conditions: Identify the conditions, pre conditions and binding relationships between each of the identified obligations, rights, performances and non-performances. Also identify the different obligation types, the various obligation states through which each obligation may pass through and their related performance event or right as the case maybe. Finally, to arrange the identified obligation states and the performance events in a time ordered sequence list.

Phase 5: Mapping to existing Business Process Workflows: In case, a business process or business workflow is available a semantic mapping between the contractual performance events and the available business process flow is executed. In case, no such business process model exists then the CWM is taken as the high level starting point for designing the detailed business process model.

Phase 6: Logical sequencing of Contract Workflow: In this phase, the user is guided to sketch a rough activity-state flow chart to help him visualize the expected choreography of contract obligation execution. The *performance events* that respond to or change each of the obligations to the *fulfilled state* are the identified points for business process interoperability. The actual CWM may be represented in any diagrammatical representation like UML activity diagrams, or BPMN (Business Process Management Notation [12]) diagrams.

Phase 7: Deducing the final Contract Workflow Model: Now, the user, takes only the performance events in the sequence in which they occur along with the obligations from each of the two individual diagrams for the obligation-performance event (from the previous phase). We have also deduced a set of contract workflow patterns that aids the user in translating to the formal BPMN version of the CWM [21].For example, *performance events* are mapped to BPMN *tasks;* each state change of an *obligation* is mapped to *intermediate events* (either message- triggered or rule-triggered, depending upon how the obligation state change is activated); the individual obligation-performance diagram as two business processes (BPD) using *swim lanes.*

5 Sample Contract Analysis

We analyze a typical contract sample between a Seller and a Buyer (Copy is available at [22]) where the two parties agree to exchange goods for money. Accordingly, we compare with the specific domain contract type specific ontology for commercial sale of goods. Principal concepts like buyer, seller, their identification, addresses etc are readily identified.

Contract Type:	Sale of Goods Contract type.
Actor1:	ABC Computers Incorporated Ltd.
Role1:	Seller
Address for Actor 1:	Österogatan 17, Kista, Sweden
Actor 2:	Stock and Financial Movers AB
Address for Actor2:	Strandvägen 2,Stockholm, Sweden
Role 2:	Buyer.
Consideration:	Computers:
Contract date:	12^{th} Jan 2005
Primary Obligation of Seller:	Transfer and delivery of goods
Primary Obligation of Buyer:	Receive and pay for goods.
Performance:	Delivery
Performed By:	Seller
Place of delivery:	Buyer's address
Date and time:	within 30 days from the time the seller receives PO.
Delivery fulfilled when:	goods received by buyer
Performance:	Pay

Performed By:	Buyer
Place:	at the place of delivery
Date and time:	at the time of delivery
Right to inspect:	owner is buyer.
Performance:	inspect goods, notify seller.
Time period:	within 15 days after delivery receipt.
Ownee:	seller

Table 1: Meta-data extracted from contract instance

The obligation type information is useful for sorting and arranging the obligations in the order of their execution and their sequence within the business time frame. Some of the identified obligations along with the obligation Owner and Ownee, for the sample contract being analyzed are listed below in table 2. The nature of obligation is defined as an *object type property* in the MTCO (using DAML or OWL) for the concept of *Obligation.*

Obligation	**Owner**	**Ownee**	**Nature of Obligation**	**Obligation Type**
Obligation to deliver	Buyer	Seller	Legal, business	Primary
Obligation to pay	Seller	Buyer	Monetary, legal	Primary
Obligation to package the goods	Buyer	Seller	Business	Secondary

Table 2: Partial List of Obligations

Similarly a list of the different rights, payment terms, delivery conditions and other particulars may be sorted out. The MTCO gives us information regarding the associated performance activities. A list of related performance activities are then drawn up, as illustrated in Table 3:

Obligation	**Possible list of Performance Events**
Obligation to Deliver	Ship goods
	Deliver goods (basic performance)
	Load goods (extracted from internal workflow)
	Mark goods (from legal regulation which needs goods to be marked for particular buyer)
	Get raw materials (extracted from internal workflow)
	Quality assurance of goods (legal requirement that goods should conform to specification)
Obligation to pay	Pay for goods (basic performance)
	Receive goods
	Inspect Goods
	Send Acceptance

Table 3: Extract from the deduced Performance Activities

The identified obligation and their performance activities are then grouped according to the actor performing them. Then we arrange the obligations starting from the primary obligation, including nested secondary obligation, followed by reconciliatory and conditional obligations. Under each listed obligation, we now include the related tasks and their subtasks. We mark the tasks, which result in a state change of the governing obligation. Finally we order the whole set in a logical time sequence to get an ordered set of events and their associated obligation and obligation states.

Figure 2 below illustrates a free sketch from such an ordered list for the seller's and the buyer's primary obligation. For example, the seller's obligation to deliver is *inactive* when the contract is signed; it changes to the *active* state when the seller receives the order from the buyer. Now the seller performs his activities like procuring the materials from his supplier, making the goods, identifying and marking the goods etc. Note that, a secondary obligation to package is activated once the goods are marked and ready for packaging. Meanwhile, if the buyer sends and the seller receives a notification for cancellation prior to his completing the packaging, then as per the contract the seller has to accept the cancellation. The seller's obligation is *fulfillment triggered* when he sends the goods to the buyer and is waiting for the buyer to send his acceptance notification. On receipt of the acknowledgement, the seller's obligation to deliver is *fulfilled* and the obligation state returns to the *inactive* state, waiting for the next execution of the contract.

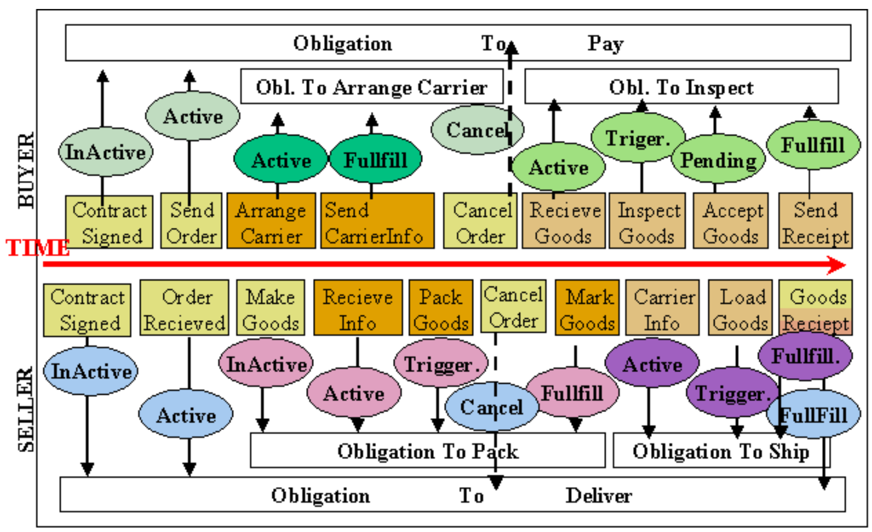

Fig 2: Obligation and Performance events ordered by Time

The last step involves the transformation of the above process description in to a formal business process model that is the CWM , using formal representation notations like BPMN. Now, we model the performance events and note the execution sequence and the interaction between the two parties as *messages* or as responding business activities on the counterpart CWM.

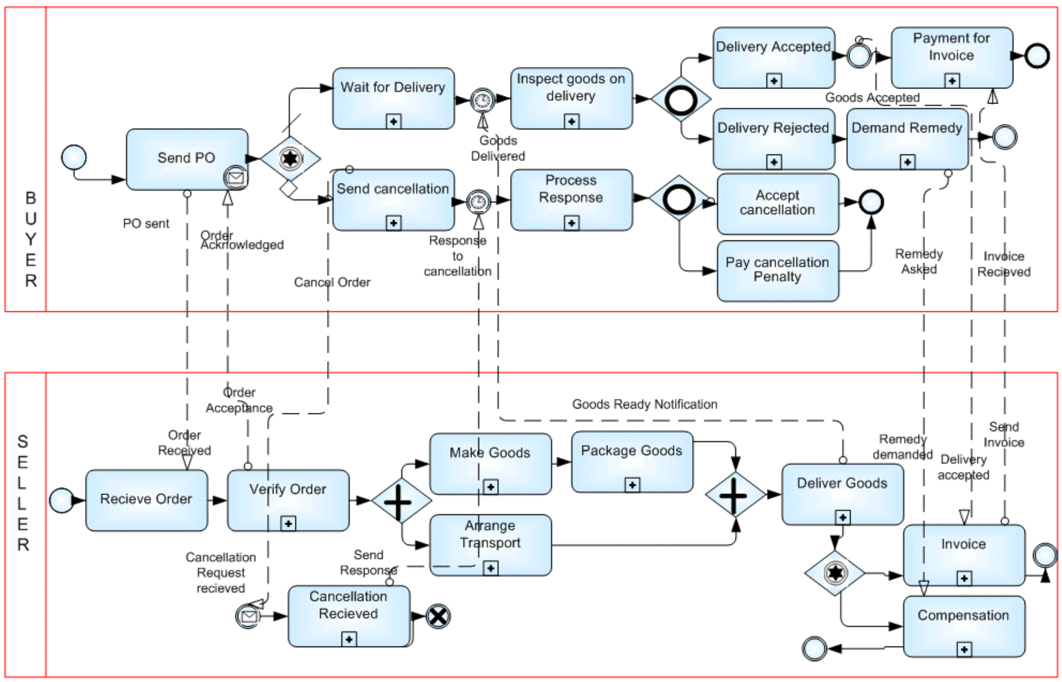

Fig 3: Overview of the CWMs for the seller and buyer

In figure 3, we show the main business activities for both the seller and the buyer. The buyer sends a purchase order and then has the option to send a cancellation within the specified time limit. On the parallel swim lane, the seller receives the order and then verifies the order and sends an acknowledgement as per the contract agreement. Note that we have modeled the order verification as a sub process based on input from the seller's internal business process models. For example, the seller may check the credit standing for the buyer, or he may need to check with his own supplier or his stock inventory or suitability for the requested delivery date etc. On the event of delivery of goods carried out by the seller, the buyer then has to perform the inspection or acceptance of the goods and so on. The sub process Cancel order received in figure 3 is used to represent the steps and options for the seller to take when he receives the cancellation request.

In this section, we have presented and illustrated our methodology for deducing a CWM starting from the contract document. We have seen based on the ontology and contract information; we can deduce a high-level business process model referred to as CWM in this paper. CWM identifies only basic and abstract level of processes, for example CWM identifies 'black box' like process definitions like 'make goods', 'package goods'. However, the CWM may be merged or mapped to the internal business workflows to get an executable business process flow, as we discuss in the following section 6.

6. Mapping the CWM to Existing Processes

Private business processes are workflow processes internal to a specific organization. When designing private business processes, consideration is paid to both business requirements and the technical context that the process should be executed in. This means that a private business is designed according to business concepts such as business activities, actors and events, but also with consideration to existing systems and services. Using BPMN, a process designer may form a process with its activities, control flow, message exchanges, which are later mapped to semantics of a single executable process specification, such as BPEL4WS.

As stated in Section 1, the final CWM may be given in the form of coordination of tasks that each of the contracting parties is obliged to. In the B2B context, thus, the final CWM is considered as a high-level partial process specification, which needs to be mapped to existing private business processes. (Figure 4).

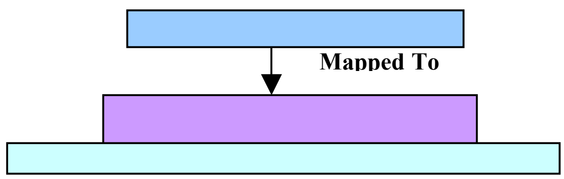

Fig 4. Mapping of the CWM with existing business processes

A CWM, given in the form of a BPMN-based model, enables the business process modeler to match the contract requirements on the process level, by mapping from the BPMN concepts as defined in the CWM to those in the existing private process. The classifications of the mapping rules are the subject of our next work.

The important conclusion is that, having the CWM in the form of a partial BPMN process model enables mapping to existing business processes. If those processes do not exist, the CWM are used as process skeletons from which complete, contract compliant process specifications are to be derived. .

7. Conclusion

As stated earlier, our objective has been to aid business-to-business process interoperability, starting from the basic level of understanding between the human counterparts of each enterprise. We have illustrated the use of ontology in our chosen domain as being successful for human knowledge transfer as well as being an Interlingua in our guidelines for deducing the CWM. The CWM in turn is instrumental in identifying the common business activities, the sequencing of message communications between the processes etc.

Our ongoing research is focused on providing semi-automated generation of the CWM and its mapping to business process flows. Another objective is focused on the integration of different domain ontologies so that the semantic network of knowledge resource can be effectively used.

8. References

[1] A Daskalopulu and M J Sergot, The Representation of Legal Contracts. AI and Society 11, 1997

[2] Kabilan, V. Johannesson, P. Semantic Representation of Contract Knowledge using Multi-Tier Contract Ontology, *proceedings of Semantic Web and Databases workshop*, (SWDB 2003)

[3] Conceptual Modeling for Distributed Ontology Environments, D L McGuinness, at www.ontology.org

[4] Yao-Hua Tan, Walter Thoen. A Logical Model of Directed Obligations and permission to Support Electronic Contracting in Electronic Commerce

[5] A Daskalopulu and T Maibaum,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

[6] Smith, H. The role of ontological engineering in B2B Net Markets, www.ontology.org June 2004)

[7] Tan, Yao-Hua. Modeling Directed Obligations and permission in Trade Contracts.31st Annual Hawaii International Conference on System Sciences, vol 5, 1998.

[8] Benjamin Grosof , T Poon ,SweetDeal:Representing Agent Contracts with exception using XML rules, Ontologies and process descriptions

[9] Griffel, M. Boger, et al. Electronic Contracting with COSMOS - How to Establish, Negotiate and Execute Electronic Contracts on the Internet. EDOC '98, 1998

[10] Yao-Hua Tan , Walter Thoen Using FLBC and Event Semantics for modeling and reasoning about Contracts.

[11] C Dellarocas and Mark Klein, A knowledge based approach for handling Exceptions in business processes. *Information Technology and Management,* Vol. 1, 3, January 2000, pages 155-169

[12] W.J Van Den Heuvel, H Weigand, Cross-Organisational Workflow Integration Using Contracts, OOPSLA 2000, 6th Business Object Workshop.

[13] Business Process Modeling Notation from the Business Process Management Initiative , specifications and white papers available online at www.bpmn.org.(accessed on March 2005)

[14] Gregor Hohpe. Enterprise Integration Patterns. Excerpts from *Enterprise Integration Patterns: Designing,Building and Deploying Messaging Solutions (*Addison Wesley). Acc. Mar 05, at. http://www.enterpriseintegrationpatterns.com/ramblings/08_integrationstyles.html

[15] C Dellarocas and Mark Klein .Designing Robust Business Processes .*Organizing Business Knowledge: The MIT Process Handbook* MIT Press, 2003, pp. 423-440

[16] M. Koetsier, P. Grefen and J. Vonk, "Contracts for Cross-Organizational Workflow Management", CrossFlow Consortium / University of Twente, TR-CTIT-18, 1999

[17] C Marshall, "Enterprise Modeling with UML: Designing Succesful Software through Business Analysis", Addison&Wesley, Reading, Massachusetts, 2000

[18] Kabilan V , Johannesson P , UML for Ontology Modelling and Interoperability, Proceedings of 1st International Workshop on Enterprise Modelling and Interoperability (EMOI 04), Riga , 2004

[19] Grosof. B, Poon T. "SweetDeal: Representing Agent Contracts With Exceptions using Semantic Web Rules, Ontologies, and Process Descriptions". IJEC journal, 2004.

[20] The Open Source Business Management Ontology. Documentation, and ontology available for download at http://www.jenzundpartner.de/Resources/RE_OSSOnt/re_ossont.htm . last accessed 25th April 2005.

[21] Kabilan V. Contract Workflow Model Patterns Using BPMN. Proceedings of the 10th International Workshop on Exploring Modeling Methods in Systems Analysis and Design (EMMSAD 05), Caise 2005, Porto.