Establishment of Virtual Enterprise Contracts

Gerald Quirchmayr[1,3], Zoran Milosevic[2], Roger Tagg[1], James Cole[2],
and Sachin Kulkarni[2]

[1] University of South Australia, School of Computer and Information Science
Mawson Lakes, SA-5095, Australia
{Gerald.Quirchmayr |Roger.Tagg}@unisa.edu.au
[2] Enterprise Distributed Systems Technology Centre (DSTC)
Level 7, GP South, University of Queensland,
Brisbane, Q4072, Australia
{zoran |colej |sachink}@dstc.edu.au
[3] University of Vienna, Institute for Computer Science and Business Informatics
Liebiggasse 4, A-1010 Vienna, Austria
Gerald.Quirchmayr@univie.ac.at

Abstract. In this paper we describe an approach to the modelling of virtual enterprise agreements. An important and ever-present issue in the set-up phase of such a virtual enterprise is the virtual enterprise contract. As one of the goals of virtual enterprises is to become operational quickly, the time consumed by the set-up phase must be reduced to a minimum. At the same time it has to be guaranteed that the outcome of this phase leads to a formally correct and executable agreement. Our approach aims at showing how to achieve a formal model of virtual enterprise contracts, which can later be executed and monitored in a particular automated contract management environment.

1 Introduction

Today's business world is characterised by an increasing need for flexible, dynamic, often short-term and task-specific alliances and partnerships. This is a result of growing customer pressures for improved, personalised, more versatile, and yet cost-effective services. Virtual Enterprises (VEs) are a form of enterprise structure through which these demands can be met.

In spite of substantial advantages of VEs, there are also a number of problems associated with them, such as having to set up legally binding contracts under enormous time pressure and with incomplete information. This incompleteness arises from parties not having sufficient background information about the other partners or about the environment in which the VE has to operate.

With appropriate IT systems in place, it is possible to considerably reduce the effort in setting up such alliances. IT systems can provide more information about the partners involved, e.g. their historic behaviour, their reputation, credit rating and so on. IT systems can also enable better compliance with the legal requirements when setting up contracts, helping to ensure their validity and enforceability. They can also provide support for better monitoring of the adherence of parties to the contract and more timely notifications of possible non-fulfilment of contractual obligations.

R. Cicchetti et al. (Eds.): DEXA 2002, LNCS 2453, pp. 236–248, 2002.

As amply documented in the literature [Susskind 1987], [de Korvin et al. 1995], electronic support for contract specification is difficult. One major reason is the highly complex task of transforming natural language representation of contract documents into a representation suitable for automated interpretation, reasoning and execution.

The first stage in setting up a VE is the establishment phase. This phase encompasses all the necessary processes needed for building business relationships between the constituents of a VE [Tagg and Quirchmayr 2001], ultimately leading to the formation of a legally binding contract. The contract specification can be compared to the design specification for some software system and in that sense is similar in its importance for the further stages in the VE life cycle.

In this paper we address the key issue of the VE establishment stage, the use of a contract-based approach to set up legally binding relationships in a VE structure and describe a formal approach that leads to an executable representation.

2 Virtual Enterprises

A Virtual Enterprise (VE) has been described [Davaclu et al 1999] as "a temporary consortium of autonomous, diverse and possibly geographically dispersed organizations that pool their resources to meet short-term objectives and exploit fast-changing market trends". It is implicit that a VE can disband when the objectives have been achieved. A VE is thus somewhat more mobile than a “strategic alliance”. In the view of some authors, e.g. [Riempp 1998], a VE also differs from supply chain partnerships in which there is a dominant partner.

2.1 The Virtual Enterprise Lifecycle

Tagg [Tagg 2001] proposed 3 stages in the lifecycle of a Virtual Enterprise, namely *VE Establishment, Business Development and Business Execution.* These stages are not necessarily distinct, but can be interleaved.

The *VE Establishment stage* is primarily concerned with forming the VE in the first place. This will involve the negotiation of contracts between the partners, the terms of which will include the processes to be followed and the information to be passed.

Unless membership of the VE is going to be fixed, processes will be needed for adding (and removing) members, and for guiding the way in which contracts are negotiated with new partners. These activities will be interleaved with the later lifecycle stages discussed below.

Business Development covers the activities through which a VE goes about getting business with outside markets or major customers. The processes for doing this will have been established in the Establishment stage. Activities will include cooperative market research, responding to invitations to tender, joint promotions and the negotiation of contracts with external organizations.

Business Execution refers to the time when the VE is actually fulfilling its business of supplying goods or services to customers. The routine business processes that are followed will have been agreed at the VE Establishment stage - they will involve actors working on different sides of organizational boundaries.

To complete the analogy with lifecycles in the natural world, there should also be a final stage called *Dissolving the VE*. However a more likely scenario is that the VE Establishment and Business Development stages, rather than ending when Business Execution starts, will continue in the background to monitor the activities of the VE and the state of the markets it was formed to address. When Business Development reports that the market has moved on, or when VE Establishment reports that partners are leaving the VE, then it may be time to terminate the VE.

2.2 Establishing Virtual Enterprises

Although the performance of VE contracts is an element of the Business Development and Execution stages, establishment of virtual enterprise contracts belongs primarily to the VE Establishment stage. We can best see the nature of this stage by considering how each of the processes within the VE Establishment stage might apply to an example, in this case the "Helen Club" of textile SMEs (Small to Medium Enterprises) quoted by [Ktendis and Paraskevopoulos 1999]. The list of processes is taken from [Tagg 2001].

Identifying broad areas of market opportunity
An original founding committee of representatives from the most highly motivated SMEs would create a proposal after working on a joint discussion document.

Creating and maintaining a constitution
The committee would create set of rules for the governance of the VE. These would be typically based on a precedent from a similar body. Company secretarial or legal advice might be sought. The rules for altering the constitution would be included. A process for winding up the VE should also be worked out at this stage.

Forming and recruiting an executive and committee members
This would be done following the constitution.

Identification and recruitment of partners
A target list of possible additional partners for the VE would be set up, and a recruitment process initiated. The committee of the Helen Club might give priority to companies providing complementary services, e.g. dyers and finishers to complement spinners and weavers.

Partner applications to join, and credit and reference checking of partners
The Helen Club was intended to be extensible from the beginning, so repeatable and consistent processes are needed to evaluate applicants.

Partner renewal, lapsing and severance processes
Processes are required to monitor the partners in the VE, so that unsatisfactory behaviour can be weeded out. Partners may lapse because of lack of involvement in joint development efforts, or they may proactively resign.

Proposal and approval of agreed processes and protocols between partners
The mainstay of any VE has to be the rules for developing and executing business. Such rules may cover joint marketing effort, main contractor/subcontractor relation-

ships, sales commissions, transport between factories, cross-organization consultancy, respect of commercially sensitive information etc. Well-defined workflows, together with stipulated data flows, may be necessary.

Negotiating contracts between the partners
Each partner company may be required to sign a legally binding contract agreeing to the above processes and protocols and respecting the constitution of the VE.

Such a contract between the partner and the VE would be complemented by contracts that cover the specific needs of the Business Development or Business Execution stages. In Business Development a group of Helen Club manufacturers may agree to participate in a joint marketing activity. In Business Execution, they will deal as an entity with suppliers, customers and each other; contracts will cover how this is to be done and how the outcomes of the operation will be accounted for and settled.

As with most contracts, what happens in the exceptional cases will form a major part of the rule base.

The focus of this paper is the writing and implementation of contracts that support the establishment and operation of Virtual Enterprises, and this is addressed below.

3 A Closer Look at the Process of Constructing and Maintaining a Virtual Enterprise Constitution

The establishment phase of a virtual enterprise is, like that of any other form of enterprise, characterized by setting up agreements on key aspects of the business to be operated through this form of operation and ultimately by the legally binding contracts which are considered as one of the major outputs of this process.

3.1 Structuring Contracts

The primary goal of the establishment phase of a virtual enterprise is to see it becoming operational, governed by a series of agreements and contracts. Agreements represent some mutually accepted rules of engagement between parties, that, when governed by the legalisation of a jurisdiction, lead to contracts, i.e. agreements that have legally binding weight.

The first of these agreements is usually a framework setting the general guidelines and principles for enabling parties to cooperate as a VE, regulating internal and external relationships that come within the scope of a VE. This framework must include key elements such as the common goals of the virtual enterprise, the cooperation structure, and activities to be carried out jointly, and details of how they will be carried out. They will be defined in the form of framework clauses, with each clause specifying one or more of the following: a common goal, a cooperation structure or pattern, activities to be carried out jointly, etc. As shown in figure 1, the class concept representing the generic model of VE contracts can be expressed in a standardized diagram convention such as the UML class diagram.

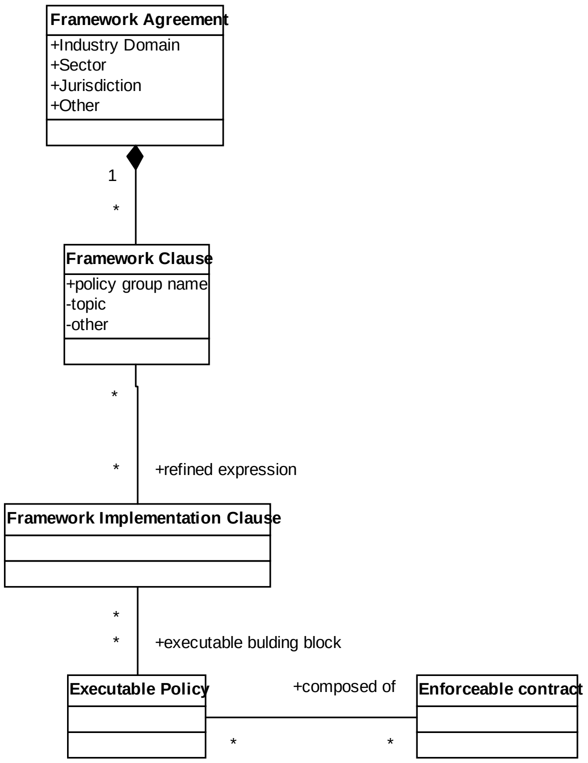

Fig. 1. Model of the Agreement Struture for VE

The structuring of agreements into framework clauses reflects standard legal and commercial practices (cf. framework agreements between suppliers and purchaser and the widespread use of international commercial terms [INCOTERMS]), whereby a clause is typically a logical self-contained unit of rules related to an identifiable topic. Examples of framework clauses in an airline virtual enterprise agreement (usually called an airline alliance) are frequent flyer program conditions, joint purchasing and training policies between partners, joint aircraft maintenance policies and so on (see figure 2). The policies expressed in framework clauses are of a general nature. They provide goals and guidelines for composing legally binding agreements, but they are not sufficiently specific to be legally enforceable.

Each of the framework clauses is further refined in terms of the framework implementation clauses. A framework implementation clause is expressed in terms of policies that are specific and unambiguous enough to form a legally enforceable contract. For example, a frequent flyer program has policies related to the calculation, the accreditation and validity of frequent flyer points.

The enforceability means that it is possible to determine whether the actions of parties to the contract are in accordance with the VE agreement stated in the contract. This determination is traditionally done by human experts and can to a certain extent be supported by automated systems, depending on the degree of ambiguity and incompleteness of the language used.

The issue of transforming clauses/expressions stated in a natural language into a formal representation has been the subject of substantial research, with many problems still remaining unsolved (for an overview see some of the classic literature of Artificial Intelligence, such as [Harmon and King 1985], [Winston 1992] and [Abelson and Sussman 1996]).

3.2 Towards a Formal Representation of Framework and Implementation Clauses

A proven approach for transforming framework implementation clauses into an executable form and for monitoring their execution is to represent them in First Order Predicate Logic (FOL) (for recent work on the representation of legal argumentation see [Kowalski 1994] and [Kowalski 1996]). For example, an airline alliance might include the following framework implementation clause in its frequent flyer program policy:

All participating airlines must have a clause in place in their frequent flyer programs that allows for the accreditation of points earned in flights that are code shared (e.g. QF1/BA7311) with other partners in addition to points earned through its own flights.

This might translate into a formal representation of the form

$$\begin{aligned}&\forall x \in \{Alliance\}\\ &R(x) = [Earn(Program(x), Miles * factor) \Leftarrow Flown(Miles, Airline(x));\\ &Earn(Program(x), Miles * factor)\\ &\Leftarrow Flown(Miles, Airline(y)) \wedge (CodeShare(Airline(x), Airline(y)),\\ &y \in \{Alliance\}]\text{, where } R(x) \text{ stands for rule applicable to x.}\end{aligned} \quad \textbf{(1)}$$

In situations where ambiguity and incompleteness prevail, terms will have to be expanded by introducing concepts from Fuzzy Logic [de Korvin et al. 1994], and ultimately by human judgment. The simple equivalence of rules, such as $[A(x) \Leftarrow (B(x) \wedge C(x)) \vee D(x)]$ being equal to $[A(x) \Leftarrow B(x) \wedge C(x); A(x) \Leftarrow D(x)]$ can be checked automatically, but more complicated identities of rules or their combinations might be difficult to prove. Therefore it is highly advisable to also require the partner airlines in our example to follow the same logical structure.

As shown, advanced logic concepts, such as permissions, prohibitions and obligations [Ciampi and Martino 1982], can be expressed in FOL with Obligation (x), being represented as $\forall x A(x)$ Permission (x) as $\exists x A(x)$ Prohibition (x) as $\neg\exists x A(x)$, where *A(x)* represents the action which a role x is obliged to, permitted or prohibited. This approach works if a resolution system similar to the one used in Prolog is provided. (For an excellent guide to Prolog see [Clocksin and Mellish]). As most of the commonly agreed rules in a VE will consist of permissions, prohibitions and obligations, this deontic extension should be sufficient.

3.3 The Role of Meta-rules

Framework clauses, which can in this context be interpreted as a sort of meta-rules, build the shell for executable rules, in our approach called framework implementation clauses. They determine the existence and the content of framework implementation clauses. They can be viewed as meta-rules, which regulate the WHAT and leave the HOW to a lower level. In cases where the aim is to integrate existing implementation clauses, they serve as constraints for checking the overall compliance. In cases where a new partner joins or where the business operation has not existed before, they can also serve as rule generators. A good example of both aims being served is the European Union data protection directive [EU Directive 95/46].

A typical framework clause for airline alliances would have to include an implementation clause stating an obligation for an existence of a rule for the reciprocal accreditation of services delivered by the partners. The implementation clause of this clause (in other words a meta-rule) can then be evaluated against the relevant rules of the frequent flyer programs of participating airlines – to check compliance between individual frequent flyer program rules and this Alliance’s rules.

Theoretically it is possible to use the VE policies as generator or even as templates for creating the rules of individual member policies (e.g. frequent flyer programs). In practice, these rules often already exist and thus, the VE policies will be used mostly as a basis for checking the compliance of individual frequent flyer programs with VE principles, i.e. making sure that none of the established constraints is compromised. This has parallels in a number of areas, such as static semantic checking of programs and the testing of correctness of implementation of software design patterns, e.g. [Neal and Linington 2001].

4 Application Example: Modelling Airline Alliance Agreements

There are already several examples of successful VEs, which give the customer the impression of being serviced by just one organisation while in reality a number of enterprises is contributing to the product or service. Some of these examples occur in the automotive, aviation and service industry, while some of the best known examples are frequent flyer programs, airline alliances and telecommunication providers. In some cases, such as the tourism industry, branding for the virtual enterprise has been very successful - leading examples are One World, Star Alliance, and the Qualiflyer Group. Even in situations where some of the partners cannot deliver for some reason (e.g. an airline does not offer a flight to a certain destination), the other members of the VE can cover the services, a highly desirable effect for the customer.

The frequent flyer programs of the larger airline networks have attempted to deliver more and more of their services online, as on-line delivery is suited to the non-tangible nature of frequent flyer points. From a legal and organisational point of view, the most interesting part is the set-up of agreements between participating enterprises, in this case airlines, hotels, and rental car companies. The goal of the VE is to make the sum of the member companies more attractive to customers by delivering value-added services. The cooperation structure is based around the collection, exchange and redemption of points. The strategic task in an airline alliance is to set up a contract harmonizing the frequent flyer programs of all participating airlines and regulating the cross-accreditation of points for services delivered by the partners.

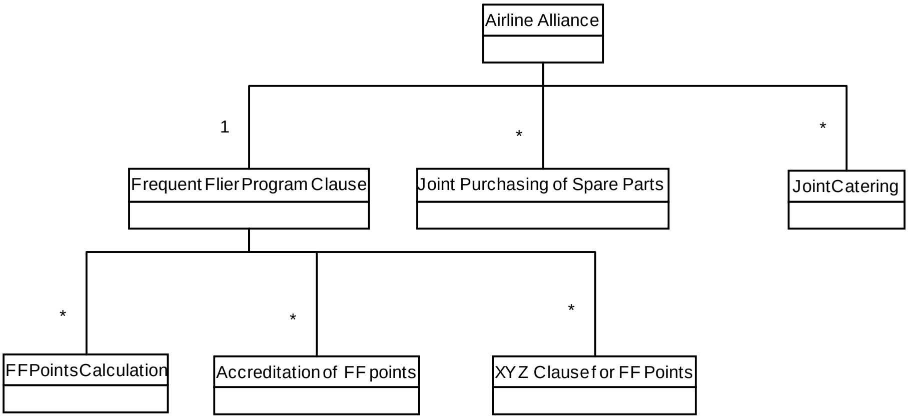
Airline Alliance
1
*
*
Frequent Flier Program Clause
Joint Purchasing of Spare Parts
JointCatering
*
*
*
FFPointsCalculation
Accreditation of FF points
XYZ Clausefor FF Points

Fig. 2. Agreement Structure for an Airline Alliance VE and its member Airlines

The actual monitoring will be the evaluation of the execution of the individual FF program rules by the participating airlines**.** For this we will apply the BCA monitoring approach as described in section 5 below**.**

By applying this model to Airline Alliances, we get the following hierarchical structure for the frequent flyer clauses:

□ The Frequent Flyer Program Clause.
□ The Frequent Flyer Program Implementation Clause, consisting of a set of rules regulating issues such as the calculation of frequent flyer points and their accreditation.
□ The actual Frequent Flyer Programs of Participating Airlines.

There will be a similar structure for other areas of cooperation such as Joint Purchasing of Spare Parts, Fuel, Catering and so on.

A realistic scenario might look as follows:

□ Airlines A, B, C, D, and E agree to form a strategic alliance called Best Airlines of the World. One of the aims of the alliance is to set up an accreditation scheme for frequent flyer programs.
□ All participating airlines should implement rules compliant to the following principles in their frequent flyer programs:
    o Calculation of frequent flyer points: one point is equal to 1 mile flown multiplied by a previously agreed factor. Miles are calculated on a basis of the official distances between the airport of departure and the airport of arrival.

  - Accreditation of points: frequent flyer points can be earned on all code share flights, but can be accredited to only one frequent flyer program.

- The frequent flyer program of airline A contains the following rules:
  - Frequent flyer points are earned on the basis of one point being equal to miles multiplied by a factor agreed by airlines participating in the alliance. Miles are calculated on a basis of the official distances between the airport of departure and the airport of arrival.
  - Accreditation of points: frequent flyer points can be earned on all flights of airline A and on all code share flights operated with a *Best Airlines of the World* (BAW) partner airline, but can be accredited to only one frequent flyer program.

For checking the compliance of an airline with the VE agreement, two levels of monitoring have to be applied. Firstly, rules in the frequent flyer program of airline X have to be compliant with the alliance agreement. Secondly, the execution of rules of the individual frequent flyer programs has to be checked by the airline. Altogether therefore, we have a three-tier hierarchy with meta-rules, rules, and the execution of rules.

Assuming that rules can be represented in FOL (for representation of legal rules in First Order Predicate Logic see classics such as [Susskind 1987]). we can then specify them in the form of:

$$A(x) \Leftarrow B(x) \wedge C(x) \wedge D(x)\ . \tag{2}$$

The meta-rules can be represented as

$$\forall x \in X \ \Big|\ R(x) = \big[A(x) \Leftarrow B(x) \wedge C(x) \wedge D(x)\big]\ . \tag{3}$$

If both sets of rules are written in disjunctive form and are not partitioned, the compliance of rules of a frequent flyer program with meta-rules can then be checked by seeing whether or not such a rule $R(x)$ exists. For the monitoring of the actual execution of a rule $R(x)$ we use the BCA monitoring approach, as indicated earlier.

In our example the meta-rules would be

$$\begin{aligned}
&\forall x \in \{BAW\} \\
&R(x) = [Earn(Program(x), Miles * \ factor) \\
&\qquad\qquad \Leftarrow Flown(Miles, Airline(x)); \\
&\qquad Earn(Program(x), Miles * \ factor) \\
&\qquad\qquad \Leftarrow Flown(Miles, Airline(y)) \wedge \\
&\qquad (CodeShare(Airline(x), Airline(y))),\ \ y \in \ BAW].
\end{aligned} \tag{4}$$

So we need to check with each frequent flyer program of the participating airlines whether they contain such set of rules, i.e. whether the frequent flyer program of airline A contains the rules

$$\begin{aligned}&Earn(Program(A), Miles * factor) \Leftarrow Flown(Miles, Airline(A));\\&Earn(Program(A), Miles * factor) \Leftarrow Flown(Miles, Airline(y)) \wedge\\&(CodeShare(Airline(A), Airline(y))),\ y \in BAW\ .\end{aligned} \quad \textbf{(5)}$$

Other interesting VE scenarios that could be modelled in this way are joint purchasing strategies, virtual competence centres and alliances between travel enterprises such as hotel chains and rental car companies.

In the following section we will discuss how the rules might be adopted in an automated contract management system.

5 Implementation of the Rule Base

The formalism introduced in the previous section can be used as a starting point for providing an increasing level of automation of those stages in the VE life-cycle that involve contract management activities. These activities cover the creation of contracts and also the management of contracts during the periods of business development and business execution stages, as discussed in 2.1.

In addition, this formalism can be built into the various components of an IT system that can support management of contracts. In this section we demonstrate this with an example of Business Contract Architecture (BCA) – which is one of the early proposals for supporting various levels of automated contract management, as described in [Milosevic and Bond 1995, Goodchild et al. 2000].

First, the BCA allows storing of *standard contract forms* and this component is called as Contract Forms Repository. These predefined forms (i.e. templates) can be used by businesses when agreeing on the specifics terms of a contract and instantiating contracts. Examples are contract forms that govern real-estate transactions, banking and insurance forms, purchase and sale of goods and services and so on. Such repositories can also contain forms for standard *contract clauses* that can be reused when deriving *new* contracts that govern specific business interactions. Availability of standard contract clauses also enables flexible changing/updating of existing contracts by simply providing references to the new contract clause from the existing contract. These changes are quite frequent in cases of long-term contracts and are known as *contract variations* and *contract escalation* clauses. These features of the BCA can thus be used as an effective mechanism for storing specific framework agreements, framework clauses and also framework implementation clauses that are valuable as a collection of pre-defined templates of use in establishing VEs. The contract forms repository can be used as facility for faster setting up and reuse of existing structures that are commonly used when structuring various VE agreements.

Second, BCA supports *digital signing* of contracts, once specific contract terms have been agreed. This can bring significant savings, in particular in cases where contracts involve multiple, geographically distributed trading partners, such as those related to international contracts, and which can involve significant time and transaction costs associated with handling the contract signing process. This is also a useful way of speeding up contract negotiation process. Once a negotiated contract is signed, it can be stored in a separate repository, termed Notary. This negotiated contract specifies a particular set of policies that parties must comply to in order to satisfy a contract. One possible format for expressing these policies can be in the form introduced in 3.2. The Notary for the Frequent Flyer example would store the full agreement structure specific to the BAW partner airlines.

Third, in cases where parties are interested in being notified about some contract non-compliance, BCA supports *Monitoring* of the business interactions governed by a contract. This can be done either by logging of business interactions and their audits at a later stage, or by a more pro-active monitoring which can be particularly applicable in cases of electronic services delivery. Essentially, this monitoring consists of evaluating policies as agreed in the contracts and stored in a Notary and comparing these with the actual execution of contracts – in particular the action of parties that are specified as part of contractual obligations.

In the Frequent Flyer example, monitoring at the Business Development stage would involve testing, against actual data, whether each participating airline had observed agreed rules when advertising their frequent flyer schemes. Likewise at the Execution stage, one could check that operations under the VE had been done properly, e.g. cross-crediting of frequent flyer points.

A special kind of monitoring particularly suitable for longer term and timed contracts *is tracking* of contracts. This allows timely reaction to some important deadlines such as contract termination, thus making it possible to re-negotiate a subsequent contract and put it in place, before or immediately after the existing contract terminates. Such tracking of contracts can be also seen as a mechanism that prevents situations in which businesses continue their interactions after the contract has expired – thus avoiding undesirable circumstances such as penalties and fines.

Finally, in cases when contract terms and conditions were breached it may be useful to provide certain *enforcement* contract terms. Although some of the enforcements can be done electronically, such as in cases of services provision and billing, the ultimate enforcements are to be executed by human decision makers.

We note that there can be several business processes identified in our business contracts model, but our architecture is essentially role-based - to enable support for many types of underlying contracting scenarios (i.e. business processes implementing them).

6 Conclusion and Further Work

In this paper we have shown how providing a formal representation of virtual enterprise contracts can support the establishment of a virtual enterprise. This formal representation serves as basis for consistency and compliance checks as well as for the development of implementation rules. Once these implementation rules have been formulated, they can be plugged into execution and monitoring mechanisms, such as those provided by the BCA facilities. We have therefore been able to point towards a way of effectively supporting the setting up of virtual enterprise contracts and consequently reducing the virtual enterprise’s time to operation.

Future plans include a detailed proposal of how the implementation of the rules in BCA would be achieved. This will include generation of rules in a standard format and the checking of sets of rules at one level of the hierarchy against those at the next higher level. There is also a need to access a range of real-world case studies, in order to collect standard contract patterns and their relationship to common B2B situations in supply chains. We will then be in a position to test the concepts in a suitable pilot environment.

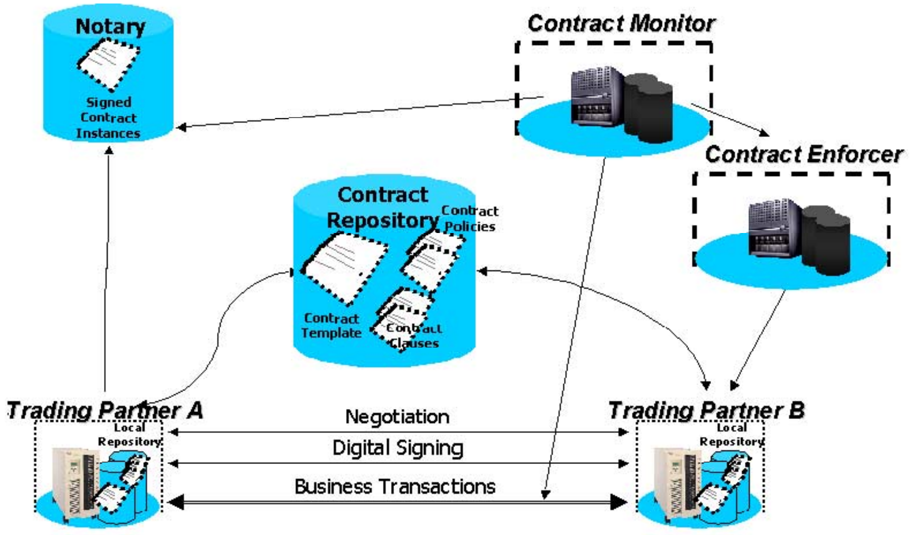

Fig. 3. BCA Architecture

Acknowledgements

The work reported in this paper has been funded in part by the Cooperative Research Centres Program through the Department of the Prime Minister and Cabinet of the Commonwealth Government of Australia.

References

[Abelson and Sussman 1996]). H. Abelson, G. and J. Sussman. Structure and Interpretation of Computer Programs, 1996.

[Ciampi and Martino 1982] C.Ciampi and A.A. Martino (Ed). Artificial Intelligence and Legal Information Systems: Deontic Logic, Computational Linguistics and Legal Information Systems, Elsevier Science Ltd; ISBN: 0444864156.

[Clocksin and Mellish] W. F. Clocksin, C. S. Mellish. Programming in Prolog, Springer, ISBN 3-540-58350-5.

[Davaclu et al 1999] H. Davaclu, M. Kifer, L.R. Pokorny, C.R. Ramakrishnan, I.V. Ramakrishnan, S. Dawson. Modeling and Analysis of Interactions in Virtual Enterprises, in Proceedings of the Workshop on Research Issues in Data Engineering - Information Technology for Virtual Enterprises (RIDE-VE'99), Sydney, Australia, March 1999.

[EU Directive 95/46] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995, on the protection of individuals with regard to the processing of personal data and on the free movement of such data.

[Goodchild et al. 2000] A.Goodchild, Z. Milosevic and C. Herring. Business Contracts for B2B, CAiSE'00 Workshop on Infrastructures for Dynamic Business-to-Business Service Outsourcing.

[Harmon and King 1985]. P. Harmon and D. King. Expert Systems: Artificial Intelligence in Business, John Wiley & Sons. ISBN 0471808245.

[INCOTERMS] International Chamber of Commerce, Incoterms 2000: ICC Official Rules for the Interpretation of Trade Terms. ICC Publishing New York 2000. ISBN 92-842-1199-9.

[Kowalski 1994] R. Kowalski, F. Toni. Argument and Reconciliation, in: International Symposium on Fifth Generation Computer Systems '94, Workshop on Legal Reasoning, Tokyo (1994).

[Kowalski 1996] R. Kowalski, F. Toni. Abstract Argumentation, In: Artificial Intelligence and Law Journal 4(3-4), Special Issue on Logical Models of Argumentation, H. Prakken and G. Sartor, eds., Kluwer Academic Publishers (1996).

[de Korvin et al. 1994] A. de Korvin, G. Quirchmayr, S. Hashemi. Dealing with Uncertainty in Insurance Contracts. Proc. of the 6th Conference on Expert Systems Applications and Artifical Intelligence, EXPERSYS-94, Houston, Texas, December 1994.

[de Korvin et al. 1995] A. de Korvin, G. Quirchmayr, S. Hashemi. Managing Uncertainty in Contract Law. Proc. of the 7th Conference on Expert Systems Applications and Artifical Intelligence, EXPERSYS-95, p. 191 ff., December 1995.

[Ktendis and Paraskevopoulos 1999] P.D. Ktendis and A.T. Paraskevopoulos. Information and Communication Technology (ICT) Infrastructure fro Small and Medium Enterprises (SMEs) Networks: Achievements and Work-in-Progress from Three European Projects, in Proceedings of the Workshop on Research Issues in Data Engineering - Information Technology for Virtual Enterprises (RIDE-VE'99), Sydney, Australia, March 1999.

[Milosevic and Bond 1995] Z. Milosevic and A. Bond. Electronic Commerce on the Internet: What is Still Missing? The 5th Annual Conference of the Internet Society, INET'95, Honolulu, Hawaii, USA, June 1995.

[Neal and Linnington 2001] S.Neal and P.F.Linington. Tool support for development using patterns. In E.Lupu and A.Wegmann, editors, Proceedings of the fifth International Enterprise Distributed Object Computng Conference, pages 237-248, Seattle, WA, USA, IEEE Computer Society.

[Riempp 1998] G. Riempp. Wide Area Workflow Management, Springer, London 1998.

[Tagg 2001] R. Tagg. Workflow in Different Styles of Virtual Enterprise, in Proceedings of Workshop on Information Technology for Virtual Enterprises (ITVE 2001), Gold Coast, Australia, January 2001.

[Tagg and Quirchmayr 2001] R. Tagg, G. Quirchmayr. Towards an Interconnection Model for Evolution of Shared Workflows in a Virtual Enterprise, in Proceedings of Third Int. Conference on Information Integration and Web-Based Applications and Services, Linz, Austria, September 2001.

[Susskind 1987]. R. E. Susskind. Expert Systems in Law: A Jurisprudential Inquiry. ASIN: 0198252730.

[Winston 1992] P. H. Winston. Artificial Intelligence. 1992. Addison-Wesley Pub Co; ISBN: 0201533774.