

# Supporting Business Contracts in Open Distributed Systems

Zoran Milosevic, Andrew Berry, Andy Bond, Kerry Raymond

{zoran,berry,bond,kerry@dstc.edu.au}

CRC for Distributed Systems Technology<sup>1</sup>

Level 7, Gehrman Laboratories, The University of Queensland, 4072, Australia

## Abstract

Open distributed systems will increasingly be used to support business transactions within and across organisations. In order to achieve this goal, fundamental concepts of business practices should be incorporated into a supporting architecture. Support for *business contracts* can be regarded as an essential ingredient of such an architecture in terms of facilitating more efficient inter-organisational business interactions. This is augmented by the prevailing feeling in business today that global interdependencies are becoming more critical for gaining and maintaining the competitive edge. In this paper we make an initial attempt to identify important contract concepts from economic and legal standpoints and use them to derive a business contract framework. We further discuss how these concepts can be incorporated into an architecture for open distributed processing (ODP) [1].

## 1 Introduction

Today, large organisations are frequently facing the challenge of service deployment in the presence of massive organisational change. In some, the speed of service deployment has effectively become the limiting factor in the rate of change of organisational structures and policies, at both intra- and inter-organisational levels. This is unacceptable in present commercial environments in which success depends on a rapid response to changing circumstances. With the ultimate objective of advancing the enterprise goals of their owners and users, open distributed systems can play an important role in facilitating organisational transformations and thus can contribute to the long-term success of organisations. An important step in this direction is the provision of a support for effective and efficient *inter-organisational* business interactions in distributed and networked environments. The essential part of this is an architectural support for *business contract* transactions. It is our view that a structured approach to implementing business contracts in an open distributed

system (ODS) architecture will contribute to the rapid deployment of services that will meet critical requirements of businesses. Furthermore, the strategic importance of Information Technology (IT) for competitive advantage makes a compelling business case for the development of an ODS architecture to support business contracts.

In the context of an ODS, support for business contracts is important to address business requirements of:

- users (both service providers and consumers) who will use ODS products (e.g. OMG's OMA [2], TINA[3], ANSA[4]) for the benefit of their businesses
- stakeholders in an ODS who will provide the underlying infrastructure

Business contracts are an essential part of business dealings and legal practices. A business contract is an agreement between two or more parties to perform or refrain from performing certain business activities. Business contracts are not directly represented in current IT technology. If open distributed processing is to successfully span multiple organisations, then business contracts must be supported: a requirement much broader than a standardised protocol which supports electronic data interchange (EDI)<sup>2</sup>, since it addresses certain semantic aspects of business contracts, e.g. electronic support for business contracts should include contract validity checking and contract enforcement procedures (these are not directly supported in EDI).

Although ODS communities recognise the significance of business concepts, e.g. by positioning them in the RM-ODP enterprise viewpoint [1], little work has yet been done within relevant groups to progress these issues. The recent establishment of a Business Object Management Special Interest Group (BOMSIG) within the OMG consortium is to the best of our knowledge among the first attempts which aim to ensure an increased usability and usefulness of an ODS in terms of supporting business services. BOMSIG's aim is to expand the awareness, understanding and use of object orientation as a tool to help enterprises meet their business needs [5]. BOMSIG is currently in the process of developing a Business Objects

---

1. 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.

2. For example, the international EDI standard set under the auspices of the United Nations, EDIFACT.

Reference Model. The fundamental component of this model is the business object. It encapsulates the storage, metadata, concurrency and business rules associated with a thing, process or event in a business [6].

The focus of this paper is developing a framework which can be used to facilitate inter-organisational business transactions. More specifically, we develop a business contract architectural framework (BCAF), applicable to both real-world scenarios and the IT systems that reflect them. This framework is based on current business practices, but also includes emerging business procedures dictated by new technologies. The BCAF embodies concepts and principles used in procedures associated with creating, executing and terminating business contracts.

In relation to the topic of IT-based inter-organisational business transactions, in particular business contracts, our work has some similarity with the research reported in [7], in which an attempt was made to extend EDI towards electronic contracting in international trade. This has been done by applying AI concepts to make the use of knowledge bases for representing and applying the complex rules and regulations governing the interaction of contractual parties. However, rather than focusing on the ways contract operations can be represented within an IT system as is done in [7], we are endeavouring to identify necessary components (and their relationships) of an ODS architecture so that electronic business contracts can be supported.

In developing the BCAF, we were guided by the principle of conformance to RM-ODP standards, so that this framework can be applied to any ODS architecture, e.g. OMG's OMA (more specifically the OMG's Business Objects Reference Model<sup>1</sup>).

Section 2 highlights the economic and legal role of contracts in human society and identifies the fundamental concepts of contracts. Based on the observations of how contracts operate in the business world, a set of basic concepts of a business contract architecture is developed (Section 3). In section 4 we describe how each concept of such an architecture can be realised in the context of an ODS, using an example to illustrate the applicability of our approach. Section 5 outlines our plans for further refinement of the concepts described in this paper towards specification techniques and infrastructure development for a typical ODS environment.

## 2 Contracts in Business

Contracts involve a relationship between two or more parties, representing an agreement that govern their inter-

---

1. In terms of the BOMSIG Reference Model, this framework will represent a part of the business application architecture [6].

actions. Contracts reflect economic and legal concerns. From an *economic* standpoint, a contract arises as a result of efficiency-seeking behaviour in a world of limited information [8]. That is, by creating an on-going relationship, individual interactions can be processed more efficiently. From a *legal* standpoint, a contract alleviates mistrust in a world of uncertainty, e.g. by constraining the unpredictable activities of the other autonomous parties. In the world of *business*, a contract is a major concept of business law<sup>2</sup>. It is a mutually binding agreement between two or more parties to perform or not perform certain acts.

The following subsections introduce the fundamental aspects of business contracts.

### 2.1 Contract Domain

To prevent any ambiguity, contracts are bound to a particular contract domain. The rules and policies with which members of a particular business contract domain comply normally emanate from statutory or administrative law.

In the real world, there are many autonomous countries, sometimes composed of autonomous states or provinces. Each of these has its own legislation which might place restrictions on the contracts that can be made within their *boundaries* which define the *contract domain*. Typically, the legal system of these countries will only enforce the contracts deemed valid by their legislation (through courts).

A contract domain is not necessarily restricted to country or other regional authorities. For example, many professional bodies are responsible for maintaining a register of licensed practitioners in their profession, and these professional bodies can regulate and enforce appropriate behaviour of those licensed practitioners. Similarly, in the world of commerce, certain rules and policies are applied to govern specific fields of commerce, e.g. stock market trading rules.

Interactions between parties belonging to different contract domains are normally governed by a set of policies which apply across domains, and thus define a higher-level domain. An example of a set of rules and policies which define such a domain is the United States Uniform Commercial Code, i.e. a uniform statute adopted in whole or in part by each state legislature in the US to govern specified fields of commerce.

### 2.2 Contract Negotiation

Contract *negotiation* is an important business activity because it is inherently competitive. Each party is seeking

---

2. Some other concepts of business law are: *agency*, where one party (the agent) is authorised and consents to act on behalf of another party (the principal), *negotiable instruments*, *tort*, *property* etc. [9].

to maximise the value it obtains from the contract while minimising its contribution to that contract, i.e. the greatest benefit for the least cost. Negotiation is a process of offer and counter-offer between the parties (perhaps via a broker); each successive offer either gives more or takes less than the previous offer. Ideally, this process iterates towards a compromise position, acceptable to all parties. Once a contract is successfully negotiated, it can be submitted by any party (or parties) to a legal expert or authority for validation.

## 2.3 Contract Validity

In the real world, the legal system determines the *validity* (and hence enforceability) of business contracts by identifying the mandatory elements. Typically these include [9]:

- *An agreement*: an offer seriously and clearly made by one party to another party, who must accept it seriously and clearly and without reservation. In addition, both parties make the agreement voluntarily, without restraint or influence, acting of their own free will.
- *Consideration*: This is something of value that each party gets or gives. Each party thus establishes obligation to each other. With few exceptions, it must be shown that both parties intended to bargain and have actually exchanged something for a contract to be enforceable by a court. Consideration can take the form of money, act performed or withheld, services rendered, other property or individual rights. In general, it need not be tangible or possess an economic value.
- *Competence* (or capacity): The ability to incur liability or to gain legal rights (e.g. does an individual have the authority to represent their organisation in contract establishment?).
- *A legal purpose*: A contract cannot be enforced unless the actions agreed upon are legal in the jurisdiction where the contract is made; in other words, a contract's purpose or object must comply with law. One party cannot bring suit against the other for breach of the contract if the act required by the agreement is illegal.

A court will enforce a contract if it meets the four requirements [9]. In general, enforcement of contracts will only occur if the contract is breached by a party to that contract.

## 2.4 Contract Monitoring

Although a contract is intended to regulate the activities of the parties involved, reality can often fall short of expectations. Having committed to a contract, a business wishes to ensure that they receive the value they expected from the contract. For this reason, the business might *mon-*

*itor* the activities that are governed by the contract to ensure that the other parties comply with the agreed terms. Equally, the business might monitor its own activities to ensure that other parties have no cause for complaint.

For example, a business might establish procedures to scrutinise all invoices to check that they receive the discounts agreed in their contract. Some methods of contract monitoring might be carried out irregularly, e.g. customer satisfaction surveys.

## 2.5 Contract Enforcement

As a consequence of contract monitoring, an organisation might decide that the contract has not been honoured by another party and might wish to take corrective action. This process of *contract enforcement* can take many forms (possibly more than one):

- to require the other party to conform in the future to the contract (e.g. to ensure that goods are delivered on-time)
- to require the other party to correct the previous problems (e.g. to replace damaged merchandise)
- to demand compensation for past problems (e.g. to pay interest on outstanding balances)
- to terminate the contract

Contract enforcement might occur through direct discussion between the contract participants. If this does not produce a satisfactory resolution, the dispute can escalate through various levels of mediation or arbitration, ultimately leading to a court case.

The steps associated with contract establishment and execution are illustrated in Fig. 1.

## 3 Contract Framework

Observing how contracts operate in the business world establishes the basis for a contract Framework, as illustrated in Fig. 1. The intention is to describe the concepts and principles of contracts without reference to the particular scenario (i.e. business or IT) to which the framework might be applied.

The following subsections describe the concepts of our contract framework.

### 3.1 Contract Domain

A contract domain defines the scope or boundary in which:

- a contract is deemed to be valid
- a contract dispute can be arbitrated
- correct contract behaviour can be enforced

The parties themselves might or might not be within the contract domain.

Therefore, a contract domain must define:

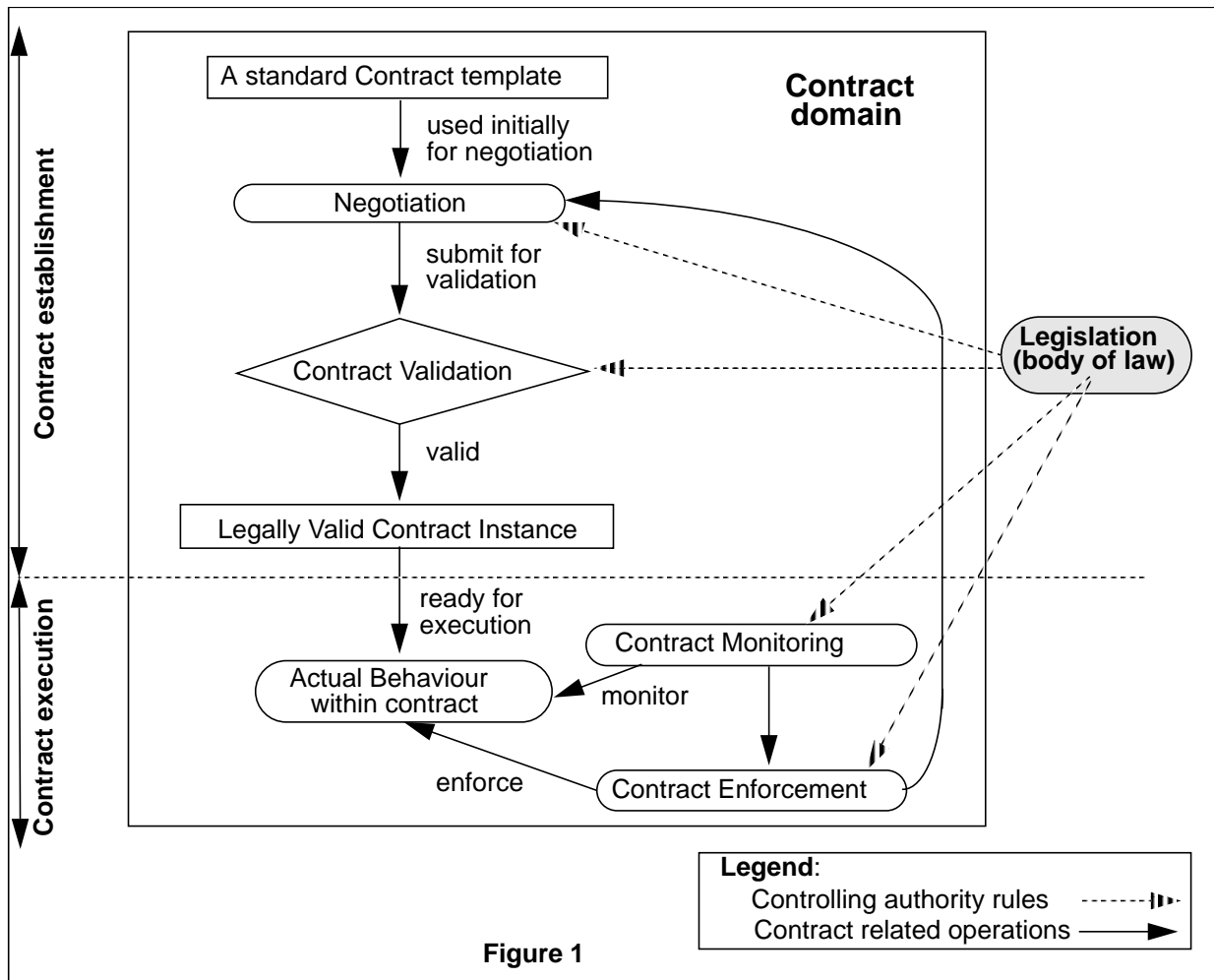


Figure 1

- a set of contract validity rules
- a procedure to arbitrate contract disputes
- a procedure to enforce contract behaviour

### 3.2 Contract Template

The contract template defines a particular class of contracts. Typically, a contract template requires further refinement to specialise the template into an actual contract.

Generally, a contract template specifies:

- the roles of the parties
- the period of the contract (the times at which the contract is in force)
- the nature of consideration (what is given or received), e.g. actions or items
- the obligations associated with each role, expressed in terms of criteria over the considerations (how much, how many, how often), e.g. quality, quantity, cost, time
- the domain of the contract (which determines the rules under which the validity, correctness, and enforcement of the contract will operate)

### 3.3 Contract Negotiation

*Contract negotiation* is a multistep process in which parties with conflicting interests come to a mutual assent regarding the terms and conditions of the contract; negotiation precedes contract establishment. Contract negotiation involves interaction between the parties, either directly or via a third party. Negotiation is the refinement of a contract template into a (mutually agreeable) contract.

### 3.4 Contract Validation

*Contract validation* is the process of ensuring that a contract satisfies the contract validity rules of the nominated contract domain. The contract validity rules can require that the parties must be members of the domain (i.e. the domain administration has some control over the activities of parties).

### 3.5 Contract Monitoring

*Contract monitoring* is the process of observing the activities of the parties for the purpose of ensuring that those activities correspond to the contract. Contract monitoring can be performed by:

- the parties themselves
- third-party agents acting on behalf of individual parties
- trusted third-party agents acting on behalf of all participants in the contract

Contract monitoring can be a continual process or it might occur only from time to time.

### 3.6 Contract Enforcement

Contract enforcement is the process of ensuring actual behaviour conforms to the contract (pro-active enforcement), or by ensuring corrective actions to minimise the deviation from the contract (reactive enforcement). Corrective actions might be performed by the parties of the contract, or by some external object at the direction of the domain jurisdiction.

The effectiveness of contract enforcement might be limited if some of the parties are outside the contract domain (which is why domain membership might be required for a valid contract). If a party outside the domain does not comply with the enforcement, then the ultimate sanction of the domain is to exclude that party from the domain or from contracts within the domain in future.

## 4 Business Contracts in Open Distributed Systems

This section demonstrates how each concept of our contract architecture is realised in an ODS and investigates the relationship between business contract components and ODS contract components. We note the limitations of such correspondences that prevent a complete ODS implementation of a business contract, resulting in some aspects continuing to be handled within the real world; examples include subjective judgements such as “intent”, “negligence” and the ultimate authority of society’s legal system.

In an ODS, a party to a contract is represented by a set of *objects* that implements the behaviour of that party (both within and beyond the contract). Further, in order to ensure that the components of our contract architecture represent an (optional) extension to an ODS architecture, we will represent contract templates as a special kind of *binding* template. In general, a binding is defined as a relationship<sup>1</sup> between a set of objects that defines the interac-

1. A *relationship* is an association between any non-empty set of entities that satisfies a predicate [10].

tion that can occur amongst the objects during the relationship. It is expected that an implementation of a typical ODS architecture will normally support a number of (low level) generic interaction types, such as RPC, multicast, and stream interactions. However, it is also possible that a binding supports high-level interactions, such as those described in business contracts, as will be illustrated in 4.2.

In an ODS, bindings and interactions are described using the type concept[10]. In order to capture and manipulate the definitions of types, an ODS typically has an extensive type management system. A type management system is a repository for type information and operations to query and manipulate types. The purpose of the type management in an ODS is to facilitate common understanding and interoperability in an open distributed environment[14]. A type management system includes a type hierarchy defining the subtype or substitutability relationship between types. This can be particularly useful in selecting contract templates and supporting contract negotiation, as will be shown in the following sections.

To illustrate the applicability of the approach, we introduce the example of an electronic stock exchange loosely based on the Australian Stock Exchange. The stock exchange is populated by stockbrokers, and the trading of shares is carried out electronically across all state capital of Australia. Electronic trading includes procedures for offer, offer acceptance and payment of shares.

### 4.1 Business Contract Domain

In terms of an ODS architecture, a *business contract domain* can be defined as a scope or a boundary within which the activities of objects are regulated by the *jurisdiction* of that domain. The activities of the jurisdiction encompass the interpretation, enforcement and administration of a body of business contract law.

The jurisdiction administers the rules and policies associated with a particular legislative authority (e.g. a state or a country) and rules which govern interactions with other domains. These rules and policies can be stored in an appropriate repository which should be accessed by the jurisdiction. The activities of legislative authorities are outside the scope of this architecture.

In the stock exchange example, the contract domain consists of all registered brokers of the exchange. The stock exchange itself is the jurisdiction, administering and applying the policies of the domain, which are defined by the Australian Securities Commission.

The jurisdiction of a domain could be implemented as a single object or a set of cooperating objects. A contract domain can be realised in a variety of ways, e.g.

- a scenario in which the domain's jurisdiction completely and solely controls a number of nodes and the networks between them
- a situation in which parties and contracts within the domain are distributed as objects and bindings throughout an ODS where the authority of the jurisdiction over its members is retained.

The electronic stock exchange is realised using the former approach, being implemented over a single, centrally controlled network operated by the stock exchange, providing access for all registered brokers.

Due to the similarity of many aspects of management and business contract issues in distributed systems, the paradigm of domains developed for management purposes in [12,13] can be adapted to further refine business contract domain operations and relationships between domains. This similarity will be investigated in future work. In particular, it is worth noting that there may be different relations between domains, i.e. hierarchical or peer-to-peer relationships.

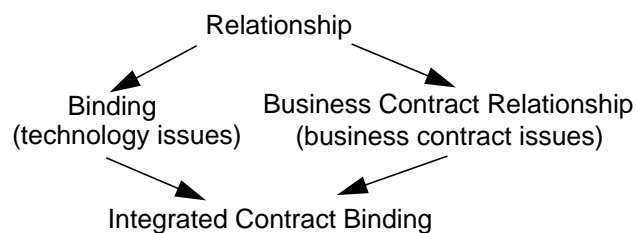
## 4.2 Contract Templates

We believe that the *binding* concept, such as one defined in [10], can be a suitable means to model business contracts since:

- it can be used to specify the particular behaviour related to business contract interactions between the objects
- it can be used to specify non-functional aspects of interactions, e.g. Quality of Service (QoS).
- it normally represents a part of an ODS type system.

At the lowest level, some general contract *terms* (e.g. the date of the contract agreement, duration of the contract, parties to the contracts, etc.) can represent a part of a type management system. Further, different contract *types* can also be incorporated in the type management system; in particular those which can be standardised (this can significantly reduce the overhead of contract establishment).

The position of a business contract type as a part of an ODS type management systems is depicted in Fig.2. In order to separate the business issues from the technical details, we identify two elements that contribute to the so-called *integrated contract binding type*. The basic business contract terms (e.g payment scheme, contract length, obligations/liability etc.) can be incorporated in what we call a *business contract relationship type*, which would include business contract specific details (Fig. 2). This type is sufficient to describe business contracts. The other element is the *binding type* which describes interactions and any QoS issues relevant to the infrastructure (e.g time constraints on interactions, required bandwidth). Therefore, the resulting Integrated Contract Binding Type would include both business specific details and technology related issues.



**Figure 2**

In the electronic stock exchange, the usual binding template is the share purchase template, which defines the obligations associated with buyers and sellers in a share trading transaction. Referring to section 3.2, the template defines:

- the roles of buyer and seller (brokers);
- that the contract is immediately effective;
- that money is being offered by the buyer, and that shares are being offered by the seller as consideration;
- other obligations, including the requirement that exchange of shares and money take place within 5 days.

The domain of the contract is, by implication, the Australian Stock Exchange. Note that this contract template is a relatively simple one, since there is no ongoing relationship associated with the contract, and the items being traded (shares and currency) are well defined and have minimal complexity. We anticipate more complex contracts for other inter-organisational interaction.

## 4.3 Contract Negotiation

Contract negotiation can be realised as the refinement of contract (binding) templates by the selection of subtypes and actualisation of parameters. This reflects the fact that negotiation often involves one party which submits a contract template offer, which the other party should either accept immediately or make a counter-offer. The subtype hierarchy can be used to determine the substitutability of contract templates. Substitutability can be used as the basis for acceptance of counter-offers.

In order to satisfy the need for agreement in contract validation (and thus enforcement), we prescribe a contract negotiation binding type. This binding type describes interactions associated with contract templates i.e., offer, counter-offer, acceptance and rejection. In the case of a successful negotiation, the binding type specifies that these interactions are recorded by a notary for use as evidence of agreement in the contract validation and enforcement activities.

In the electronic stock exchange, negotiation is achieved by having the system match offers to buy with offers to sell. An exact price match must be achieved for the successful establishment of a contract to purchase.

Offers are placed on a public “blackboard”, with open access for all brokers—the negotiating environment is hence competitive, rather than strictly two-party. In this environment, the contract template is fixed, with negotiation based purely on parameters (i.e. price and number/type of shares).

#### 4.4 Contract Validation

Contract validation is an optional step performed individually or collectively by parties to the contract. As described in 2.3, there are four elements of a legally valid contract. The realisation of these elements in an ODS occurs as follows:

- *An agreement.* The existence of an agreement is proven by the record of negotiation kept by a notary as described in 4.3. Clarity is implied by a common understanding of the types associated with the contract. This common understanding is provided by a type management system. Descriptions of these types are introduced to the type manager by the jurisdiction. In the electronic stock exchange, agreement and understanding is achieved through system-controlled negotiation using a fixed template.
- *Consideration.* The existence of consideration is ensured by requiring that contract templates include a description of what is exchanged by the parties to the contract. The jurisdiction prescribes what exchanges are considered acceptable consideration through definition of types in the type management system. In the context of an ODS the most common forms of consideration are information and currency. In our example, consideration is in the form of currency and stocks.
- *Competence.* Competence is assessed through certification authorities that can assess the competence of parties to fulfil particular roles, e.g. a credit rating agency can assess the ability of a company to meet financial obligations of a contract. Typically parties demonstrate their competence to such certification authorities in advance to streamline the checking of their competence when establishing contracts. We do not constrain the establishment or operation of these authorities. In the general case this checking involves some form of authentication of parties. We assume that such a security procedure is established and concentrate on contract specific issues.  
In the stock exchange example, competence of the brokers is implicitly guaranteed by their registration. However, we can envisage an extended system that identifies the buyer for whom stocks are being traded, allowing checks for such things as buyer bankruptcy.
- *Legal purpose.* Legal purpose can only be determined through interpretation of the laws of the contract domain stored in say a business law repository. In an

ODS, the legal purpose constrains the actions that may be performed by parties to the contract. Such constraints can be implemented by requiring that all contract templates satisfy constraints specified as types of a type management system. Formulation of the types is a responsibility of the jurisdiction. However, the interpretation of law requires either human intervention or an AI approach such as that suggested in [7].

In the current electronic stock exchange, the legal purpose of the share trading contract cannot be fully determined at the time of trading. For example, certain industries have minimum Australian ownership constraints, and purchaser information is not currently available on the system. Manual, retrospective action is used to rectify any violations of the law. A more complete system that identified the purchaser of shares could avoid such problems automatically.

It is worth noting that contract validation does not need to be used at each contract establishment procedure. For example, once the parties to the contract have established a contractual relationship, they both deal with legally valid contract types/terms which may be exploited in subsequent contract negotiations/renegotiations.

To summarise, contract validation can be realised by using rules and policies of the underlying business contract domain (possibly incorporated in type management) and through implementation of appropriate supporting services to facilitate validity checking.

#### 4.5 Contract Monitoring

After the contract terms are checked for validity and the contract is accepted by all parties (i.e. objects) the interactions between the objects (e.g. service provision) can proceed. In the course of this, either party may check whether the contractual obligations have been met during the contract realisation, i.e. the *consideration* element of the contract is being fulfilled. More specifically, this requires recording of the actions and measurement of the performance of parties, ensuring that they comply with the contract specification.

Contract monitoring can be done by parties themselves (and possibly verified later by contract enforcement activity if needed; similar to maintaining diaries which can be regarded as some sort of legal documents)

Alternatively, a trusted third-party Contract Monitor (CM) can be used by one or more parties to the contract. The parties employing the CM specify the actions required upon detection of contract non-performance, e.g. it can notify an appropriate contract enforcement component.

Hence, the CM has the following roles:

- monitoring activities of parties
- recording actions and measuring performance
- dealing with non-performance of parties

Contract monitoring can be implemented either within the objects representing the parties or by the introduction of auxiliary objects into the binding to monitor the interactions between the representative objects.

In the stock exchange example, monitoring is carried out by the parties themselves, with any violations reported to the stock exchange. Note that there is no strong requirement to monitor QoS, since there is no scope for variation in the quality offered by brokers in a share trading contract.

#### 4.6 Contract Enforcement

Contract *enforcement* should ensure that the actual behaviour conforms to the contract. This can be done in either of the following ways:

- pro-actively through constraints implemented within the binding between the representative objects. This can potentially be implemented through dynamic type checking of interactions between objects.
- reactively, via auxiliary objects, which might or might not be part of the binding, taking corrective action to minimise the deviation from the contract
- post-contract, by constraining future activities within the contract domain for parties that have violated contracts

It is important to note that the pro-active approach to contract enforcement is not widely used in business contracts. We envisage, however that a system of electronic commerce can effectively support pro-active enforcement, since a business law is normally less ambiguous than common law and direct interpretation is possible. Additionally, reactive and post-contract enforcement will usually require arbitration and possibly human intervention in determining the appropriate (corrective or punitive) actions.

In the stock exchange example, enforcement is carried out reactively, with financial penalties for late or non-settlement of the contract. This is carried out automatically once a violation is detected. In addition, post-contract deregistration of brokers can take place when regular violations occur, although this is carried out manually.

#### 5 Future Work

We are currently in a process of further refining the concepts described in this paper towards specification techniques and infrastructure development for an example of an ODS environment being developed within the DSTC. Once this contract framework is tested on our ODS architecture, we plan to relate such a refined business contract architecture to other ODS architecture models, e.g. OMG's Business Objects Reference Model.

While understanding of contracts was a necessary starting point to develop a business contract architecture for an ODS, it is interesting to investigate possible influence in the other direction, i.e. the changes in structures and behaviour of organisations as a consequence of contracts electronically negotiated, monitored and enforced. This represents a different research area, i.e. exploring the expansion of business scope provided by ODS-implemented contracts and the role of IT in changing the nature of business relationships in ODS-based business networks.

#### References

- [1] ISO/IEC DIS 10746-3.1 Draft Recommendation X.903: *Basic Reference Model of Open Distributed Processing - Part 3: Prescriptive Model*, April 1994.
- [2] Object Management Group, Object Management Architecture Guide, second edition, Sept. 1992. Revision 2.0, OMG TC Document 92.11.1.
- [3] Barr, W., Boyd T., and Inoue Y., 'The TINA Initiative', IEEE Communications Magazine. March 1993.
- [4] Architecture Projects Management Ltd. (APM)., 'ANSAware 4.1 Application Programmer's Manual, February 1993.
- [5] Object Management Group, Business Object Management Special Interest Group, Description of a Business Object, Version 2.0, Oct. 1994.
- [6] OMG BomSig - Business Object Reference Model, Cory Casanove, Dec. 1994.
- [7] Lee, R.M, Dewitz, S.D., *Facilitating International Contracting: AI Extensions to EDI*, International Information Systems, January 1992.
- [8] Oster S., *Modern Competitive Analysis*, Oxford University Press, 1990.
- [9] Reinecke et al., *Introduction to Business - a Contemporary View*, Allyn and Bacon, 1989.
- [10] Berry, A. and Raymond, K., *The AI✓ Architecture Model*, in Proceedings of ICODP'95, Feb. 1995, Brisbane, Australia.
- [11] ISO/IEC DIS 10746-2 Draft Recommendation X.902: *Basic Reference Model of Open Distributed Processing - Part 2: Descriptive Model*, 1994
- [12] Sloman M., and Twidle K., *Domains: A Framework for Structuring Management Policy*, in Network and Distributed Systems Management, Ed. Morris Sloman, Addison-Wesley, 1994.
- [13] Moffett J.D., *Specification of Management Policies and Discretionary Access Control*, in Network and Distributed Systems Management, Ed. Morris Sloman, Addison-Wesley, 1994.
- [14] Brookes, W., Arnold, D., Berry, A., Bond A., Indulska, J., Raymond, K., *A Type Model Supporting Interoperability in Open Distributed Systems*, accepted for the TINA95, Feb. 95, Melbourne, Australia.