Discretionary Enforcement of Electronic Contracts

Zoran Milosevic[1], Audun Jøsang[1], Theo Dimitrakos[2] and Mary Anne Patton[1]

[1]Distributed Systems Technology Centre *
University of Queensland, Brisbane QLD 4072, Australia
tel:+61-(0)7-3365 4310, fax:+61-(0)7-3365 4311
zoran@dstc.edu.au, ajosang@dstc.edu.au, mapatton@mac.com

[2]Central Laboratory of the Research Councils
Rutherford Appleton Laboratory, Chilton, Didcot Oxon, OX11 0QX, UK
tel:+44 (0)1235 446387, fax:+44 (0)1235 445831
t.dimitrakos@rl.ac.uk

Abstract

As in traditional commerce, parties to a contract in e-business environments are expected to operate in good faith and comply with mutually agreed terms of contract. It may be the case however that deviation from the agreed contract obligations occur either intentionally or due to force majeure. In this paper we argue that there is value in providing various levels of automated support to deal with contract non-compliance in e-marketplaces in order to reach the best overall outcome for all parties. This includes monitoring contract significant events, simple notifications to the parties about non-compliance events and a range of enforcement mechanisms. These mechanisms can be either non-discretionary (as in preventive security mechanisms) or discretionary, which rely on a number of control mechanisms that are applied when contract rules are violated. We describe a number of such control mechanisms and how they can be used to extend capabilities of a contract management architecture previously developed.

1. Introduction

There is a growing literature on research and implementation efforts on the provision of automated support for the establishment and execution of electronic contracts (cf. [16, 15, 1, 13, 9, 18, 4]). In general terms, e-contracting covers a variety of services such as brokering to identify and match prospective business partners; negotiation between partners; lodging of signed electronic contract documents; contract performance monitoring; mediation, dispute resolution and other activities aimed at facilitating correct contract-agreed behaviour. We refer to these last set of activities as contract enforcement.

Electronic contract enforcement covers various mechanisms for ensuring that actual behaviour of parties governed by a contract is compliant with their expected behaviour as stated in the contract. This includes both non-discretionary approaches (e.g. as in preventive security mechanisms) and discretionary approaches, which rely on various control mechanisms that are applied when contract rules are breached. These two approaches can be illustrated with a simple real life situation concerning certain rules when driving. There is nothing preventing me from passing a red light or exceeding the speed limit, although both actions are prohibited by law. Yet I know that passing a red light or exceeding the speed limit are both breaches of law, so the expected responsible behaviour is that I will control my actions and comply. However, what the legal system actually enforces is the penalty if it can be determined that I broke the law; in fact it does not enforce the law itself - *at least not in a preventive manner*. In addition, there are situations when passing a red light and speeding can be judged responsible behaviour, even in legal terms, for example if a passenger is so seriously ill that breaching the law in order to arrive at the hospital quickly is a question of life and death.

In e-business (as in real-life) enforcing a contract in a non-discretionary manner - however desirable it may be, in theory - can be too expensive and inefficient to implement. That is why designers (or legislative/regulatory bodies) may prefer to establish control mechanisms which react upon a

*The work reported in this paper has been funded in part by the Co-operative Research Centre for Enterprise Distributed Systems Technology (DSTC) through the Australian Federal Government's CRC Programme (Department of Industry, Science & Resources)

violation of a law or an obligation in a discretionary manner rather than prevent the violation itself. In other words the system (or policy maker) exhibits trust in the actor to behave as prescribed, and if necessary takes corrective actions upon the violation of the contract. Of course the extent to which compliance with the contract can be left to the actor's discretion, depends on the level of trust in the actor.

The legal system, which traditionally is seen as the strongest mechanisms for contract enforcement, is seen as inappropriate for many e-commerce disputes [2]. Problems include substantial legal costs, which often outweigh the value of the transactions in dispute, and the fact that the court process can be lengthy [17]. It can also be difficult to determine which law applies to e-commerce disputes, which authority has jurisdiction over a dispute, and whether or not the decision is enforceable across borders. Although legal mechanisms will remain as a means of enforcing electronic contracts, a range of alternative mechanisms that are highly suited to electronic marketplaces, may be used to enable early resolution of deviations. In general, such measures that have the purpose of avoiding escalation of disputes to the legal stage, have been called *alternative dispute resolution mechanisms* (ADR) [2].

In this paper we present a set of mechanisms for supporting discretionary contract enforcement. Our approach can be described as an ADR, and we show how it can be applied to an architecture for electronic contracts management such as the Business Contract Architecture (BCA) developed in [14, 16, 15]. We begin by introducing mechanism needed to support electronic contract management, including mechanisms to increase trust between parties in contractual relationships. Section 3 then describes how these mechanisms can be implemented in a role-based architecture supporting the full contract life-cycle. Section 4 describes how the various stages of contract enforcement are tied together, and section 5 provides an example that illustrates our approach. The conclusion discusses some future research issues.

2. Mechanisms for Electronic Contracts

Various mechanisms are needed during the contract life time. These are grouped into mechanisms supporting contract establishment, mechanisms supporting contract execution, and finally mechanisms supporting trust.

2.1. Mechanisms for Contract Establishment

2.1.1 Storing Contract Templates

This mechanism allows for storing of contract templates (i.e. standard contract forms and their building blocks) to facilitate reuse when drafting contracts. The latter is motivated by frequently adopted practices of lawyers (and other authorities engaged in drafting contracts) to compose contracts based on pre-defined contract clauses, boilerplates or other building block for contracts. To this end, it is valuable to enable electronic storage of such building blocks and their relationships, as needed. We note that these templates contain natural language description of contract fragments or full standard contract forms, but can also include data type information for certain fields. These fields are to be filled to produce a specific contract instance - and the data type information allows automated processing of various aspects of contracts.

2.1.2 Contract Validity Checking

This mechanism ensures that contracts satisfy legal validity aspects such as competence, unambiguity, consideration and legal purpose. Although the ultimate responsibility for legality of contracts is left to humans, we envision that a number of legal rules checking can be done by employing automated tools that enable checking contract policies against legal policies of an outer legislative domain, checking that the contract policies satisfy basic security mechanisms such as confidentiality, availability, accountability and fairness and so on.

2.1.3 Negotiation Mechanisms

The purpose of negotiation mechanisms is to facilitate the establishment of contractual agreements between parties. Two broad categories exist. In a simpler case, where the structure of contract template is fixed and pre-defined by some other authority who prescribes contracts (as in most standard contract forms, e.g. insurance contracts, real estate contracts, service level agreements etc.) the negotiation is referred to agreeing on the values in the contract as defined in the standard contract form. A deal between two parties is reached when a mutually agreed contract is arrived at. In a more complex case the negotiation may include negotiating the structure of the contract (e.g. addition or removal of certain clauses and other policies) and this may require a more pro-active contract validity checking. In the case of standard contract forms, this validity is effectively guaranteed by the contract drafter who issued the contract form in the first place.

2.1.4 Storing Contracts

This is to keep evidence of contracts that are agreed by parties. This mechanism represents a key feature for contract automation (in effect replacing traditional filing cabinets). 3. It provides a basis to support various automated contract management functions, such as notifications of contract expiration and other more complex monitoring features.

2.2. Mechanisms for Contract Execution

2.2.1 Contract Monitoring

Considering that actual party's behaviour to a contract can deviate from its agreed behaviour due to either their internal objectives of some other set of (possibly new) policies imposed, there is a need to provide a mechanism that would allow comparing actual vs. agreed behaviour. According to the similar problem domain from control theory, a monitoring mechanism can be put in place, performing a role similar to a sensor. Such a mechanism allows defining which contract significant events need to be observed and which contract policies need to be evaluated to determine whether there was a breach to the contract. Once such a deviation has been detected, further measures can be taken to notify parties about this breach or to provide some other interventions, including corrective measures. We refer to all such measures as contract enforcement. Contract enforcement covers both the non-discretionary (or preventive) mechanisms and discretionary mechanisms such as mediation and arbitration as described below.

2.2.2 Contract Notifications

Contract Notification mechanisms allow for notifications of parties involved in contract (or their proxies) about the existing or possibly arising contract non-compliance - detected during the monitoring activity. They can also be used to send reminders to parties to perform actions specified by contract.

2.2.3 Contract Mediation

Contract Mediation mechanisms offer the contractual parties recourse to a fair, reliable and effective process for managing situations where one or both parties deviate from the agreed contract, i.e. there is a dispute.

The dispute mediation phase begins when a non-compliance to contract is detected and a subsequent notifications sent to the non-compliant party are not acted upon to bring contract execution back on track. Next, a series of interactions may result between contracting parties possibly using third party services attempting to come to a resolution. A successfully settled dispute can be seen as a modification or addition to the original contract.

2.2.4 Contract Arbitration

In conventional (non-electronic) contracting practice, compliance with prescribed behaviour is typically evaluated individually by each party and where parties' views differ, disputes arise that require some form of resolution. Such resolution is normally undertaken by an arbitrator whose authority both parties recognise, or in extreme cases by a judge in a court of Law. Resolution comes about on the evidence presented by the parties (or their legal representatives), about their individual actions in the course of their exchange, with reference to their agreement. A human arbitrator or judge arrives at a ruling, typically with no first-hand access to the truth or falsity of the information supplied by the parties or their representatives; under the English law, the concept of "balance of probabilities" is used to qualify a ruling. An arbitration mechanism can be compared to the functionality of controller, from the domain of control theory.

An interesting arbitration mechanism which makes use of an artificial agent (or a collective of agents) is explored in [3]. Here the agent undertakes the role of an e-market controller that monitors contract execution and assesses any deviations. The e-market controller has access to a representation of the contract instances and the policy rules associated with them.

The controller also has access to information provided by advisors outside the parties directly participating in contract execution. These may be certification authorities, reputation systems, regulators or controllers of other associated markets[1]. In the architecture explored in [3] the controller's decision-making is informed by the contract instance, and each party's view of whether its own and the counter-party's behaviour comply with it. Thus, the controller forms an opinion on the basis of such evidence (and possibly additional recommendations from agents representing the parties), in a spirit similar to a (human) judge's process for arriving at his ruling.

The controller may have limited or no means of establishing with absolute certainty that an action is indeed performed as specified in the contract instance - parties may maliciously or inadvertently misinform the controller. The controller is therefore required to adopt an opinion about what actually happened, based to some extent on the parties' opinions. In analogy to conventional, non-electronic, settings, an agent's forming of opinions on the basis of information supplied by other agents is subject to trust. Ascending from the characterisation of trust provided in [5][2], *Subjective Logic* [10] can be applied as the formal basis for the evidence-based reasoning underpinning contract arbitration as described in [3]. A brief descriptions of Subjective Logic including the operators *discounting* and *consensus*

[1] A taxonomy of mediating roles that commonly appear in e-business exchanges is provided in [5] and some fundamental properties of trust relationships between these roles are analysed in [7].

[2] Trust by a party A in a party B is understood in [5] as the measurable belief of A in B behaving dependably for a specified period within a specified context. Trust affords an agent reasonable grounds to rely for a critical period on behaviour or on information communicated by another agent. Its value relates to the subjective probability that an agent will perform a particular action (which the trustor may not be able to monitor) within a context, and in a dependable manner.

can be found Section 5 and in the Appendix.

2.3. Mechanisms to Support Trust

Trust mechanisms can be employed at both the contract establishment and contract execution phases in order to address various uncertainties that may result in electronic contracting environment.

2.3.1 Reputation Systems

Reputation systems have emerged as a method for fostering trust amongst strangers in e-commerce transactions. A reputation system gathers, distributes, and aggregates feedback about participants' behaviour. These mechanisms can help people make decisions about who to trust in electronic marketplaces and may also provide an incentive for honest behaviour, due to the fact that current behaviour will be remembered and may have consequences for future business transactions[19]. Figure 1 illustrates how feedback is collected from parties in past transactions, and how reputation ratings are provided online for partners of potential future transactions.

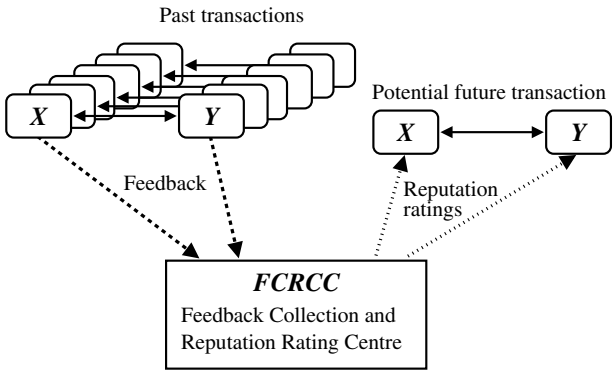


Figure 1. Collecting feedback and providing reputation ratings

Several reputation systems have been deployed in practical applications or proposed in the literature, and a reputation system that fits well into the requirements of contract management systems is the so-called Beta Reputation System [12]. In contrast to most other reputation systems which seem intuitive and ad hoc, the beta reputation system has a firm basis in the theory of statistics. The reputation rating in this system is in the range $[-1, 1]$ where -1 and 1 represents the worst and best possible ratings respectively and 0 represents neutral rating. Feedback provided by the contractual parties can be given a weight as a function of the transaction value as well as of the feedback provider's own reputation rating. The system also incorporates the concept of forgetting which gradually reduces the weight of old feedback.

2.3.2 Generic Security Mechanisms

In spite of the fact that contracting mechanisms introduced above provide a good basis for automating many mundane aspects of contract management activities, they will not be fully adopted unless there is sufficient trust in both the electronic contracting systems and other participants involved. To this end, one needs to provide a number of mechanisms which will facilitate increasing trust in these.

We consider trust as a more abstract concept from which generic security mechanisms and other aspects of dependability that underpin the validation and execution of e-commerce contracts, can be refined. For example, usual authentication security mechanisms can be applied to verify that the identity of a contracting party is as claimed and authorisation mechanisms allow giving a party access to system objects based on its identity and with respect to agreed terms and conditions. Next, confidentiality, integrity and availability are security mechanisms that underpin the competence aspect of a contract validation. Further, accountability (including non-repudiation) mechanism needs to be put in place to make sure that parties involved in contracting (and also the supporting third parties) are responsible for their actions. Finally, fairness requires an ability of contract management system to enforce contract execution, either using some automated features of by means of human involvement.

In summary, by exhibiting trust in the actions of an actor, monitoring the actors behaviour, acting upon failure in trust and adjusting the level of trust in an actor depending to reflect the deviation between actual and expected behaviour, leads to more flexible contract performance monitoring and contract enforcement architectures. This point will be elaborated in next sections.

2.3.3 Insurance

Insurance services for electronic commerce transactions participate in a transaction as third parties that assist the establishment or facilitate the increase of trust for a specific transaction by underwriting (a part of) the risk associated with the transaction on behalf of the insured actor. Entities providing insurance services that cover from financial loss caused by a fraudulent transaction on the Internet include credit card companies, who protect the card holder by underwriting (part of the) loss from stolen or misused credit cards, and merchant insurance providers, who protect the supplier against fraudulent transactions and charge backs by providing Cardholder Not Present (CNP) fraud insurance[3].

[3] See `http://www.iib.com.au/e-commerce.html` for an example of CNP insurance.

3. Role-Based Architecture for Contract Establishment and Execution

This section describes the basic components of an architecture that implements the mechanisms from Sec.2. Figure 2 depicts this role-based architecture, and indicates key information flow between the roles which in general are involved in more than one process. The architecture represents an extension of the Business Contract Architecture (BCA) described in [14, 16, 15].

3.1. Roles Supporting Contract Establishment

The following roles supporting the process of establishing a contract.

- **Negotiator** mediates the negotiation process (alternatively this can be carried out by the parties themselves). During the negotiation phase parties can exchange contract templates (offers and counter-offers). Contract templates may be submitted for validity checking.
- **Validator** ensures the creation of legally valid contract instances, assessing proposed contracts against various aspects of contract validity such as competence, clarity, legal purpose and consideration elements. See [15] for further details on contract validation.
- **Notary** is a trusted party that stores contract instances after the contract has been agreed upon, checked for validity and signed by both parties. Such contract instances can be later used as evidence of agreement in the contract monitoring and enforcement activities. Notary component can be also hosted by one or both parties involved in contract.
- **Contract Forms Repository** provides storage and access to standard contract forms or contract clauses, depending on contractual scenario. It can be used by parties to the contract who use pre-defined contract forms to produce individual contract instances or by contract drafters who are defining building blocks for contracts. There may be also a need for a specialised contract templates editor that can provide functionality of both text editing but also type definitions for the fields within the contract.

3.2. Roles Supporting Contract Execution

The following roles support discretionary contract enforcement during the performance of a contract.

- **Monitor** enables monitoring of the activities of parties, measuring their performance and recording the relevant events. It can also signal a contract non-performance to the Discretionary Enforcement Moderator (DEM, see below) if it detects such an event.

  Monitor is subscribed to contract significant events and when these occur, it evaluates the policies for these events, against the agreements that are stored in Notary. Contract Monitor can be likened to a sensor that passes the result of evaluation to other components, as needed. These other components can be a Notifier, which simply send notifications formatted in appropriate way to the parties involved or to a DEM component, to do some further more sophisticated processing such as mediation and arbitration.
- **Notifier** implements various notifications mechanisms needed to send warning messages to indicate a pending contract-significant event, including possible non-compliance event that may be detected. To simplify presentation, Notifier is not shown in the figure.
- **Enforcer** applies enforcing actions in a non-discretionary way directly to the parties to ensure that some specific behaviour conforms with the contract. Alternatively it can inform the Contract Validator which may prevent further access to the system by the non-performing parties. From a control theory point of view, this role is analogous to an actuator.
- **Discretionary Enforcement Moderator (DEM)** forms an opinion about the extent of deviation by the non-performing parties. Although we do not mandate any particular solution to describe how an arbitrator arrives at this opinion, we believe that often this well be based on second-hand evidence, and if some quantative method can be used to guide this process, we propose the use of Subjective Logic.

  Once the arbitrator forms such an opinion, it chooses a route of action which may invoke settlement leading to the success of a suitably amended transaction. Alternatively, it may endorse the enforcement of corrective measures to be executed by a preventive security mechanism realised by the Contract Enforcer role. (An overview of the Moderator's decision making procedure is modelled as a finite state machine in Figure 3.)

  The DEM forms its opinions on the basis of evidence about deviation of the non-performing parties, that is provided by the Contract Monitor, external advisors, and possibly additional recommendations from agents representing the parties, in a spirit similar to a (human) judge's process for arriving at his ruling.

  During this process the DEM component may take the following specific roles (which can be viewed as refinements of the Moderator role).

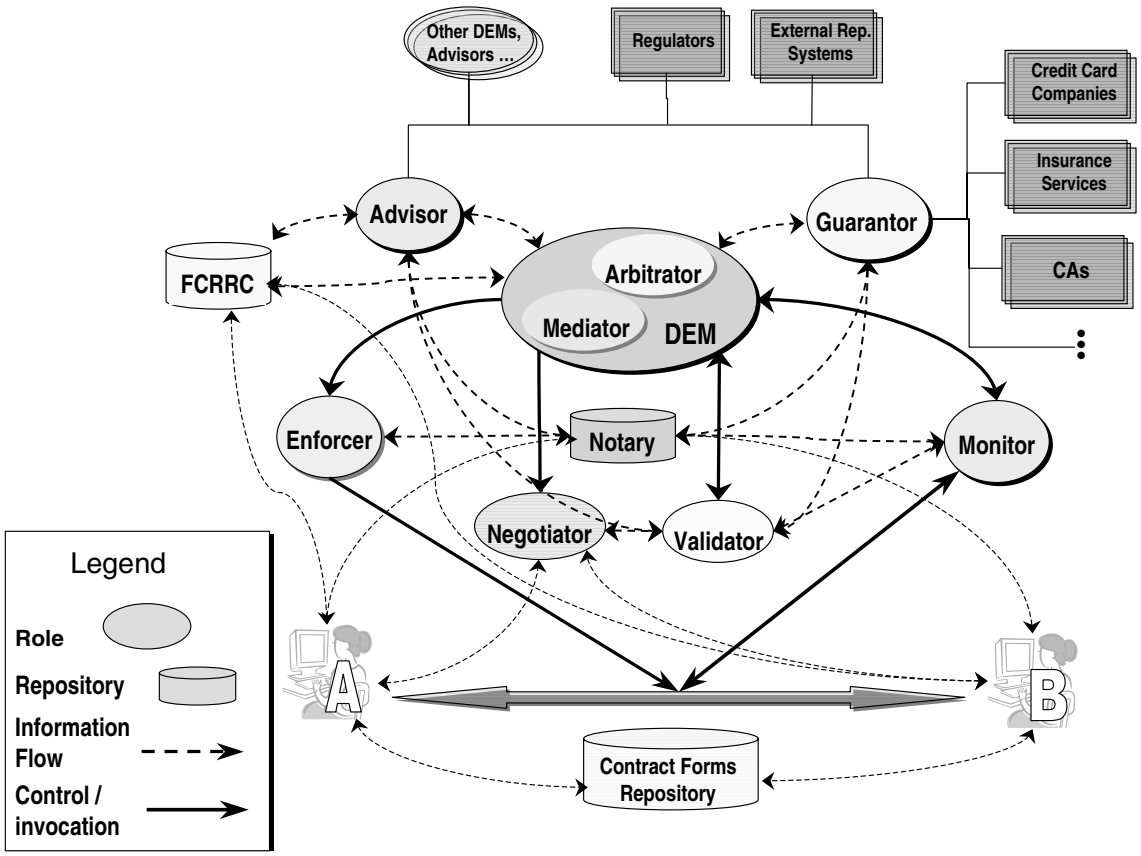


Figure 2. Roles

– **Mediator** who initiates a settlement leading to the success of an amended transaction or decides failure of mediation leading to the invocation of arbitration. If the agreed contract deals with the observed deviations by non-performing parties, then the mediator may choose to send notifications to the non-performing parties - that they should comply with the rules pertinent to such deviations. In some cases, contract amendment may be necessary, in which case the Contract Negotiator and the Contract Validator may need to be involved in order to bring about a settlement and produce an amendment to the contract.

– **Arbitrator** who takes over when a settlement as per above cannot be reached, or when a party's deviation from the expected performance is high enough to justify the deployment of corrective measures. An arbitrator may initiate the enforcement of corrective measures through the Contract Enforcer, leading to the recoverable failure of the transaction and, potentially to penalising the non-performing party. In the absence of any suitable corrective measures, the Arbitrator may signal correction failure, in which case the Contract Validator is informed so as to prevent further access to the system by the non-performing parties, if necessary, and the case is carried on outside the Contract Architecture.

3.3. Roles Supporting Trust Establishment

We distinguish three special roles that entities mediating in a trust relationship can play in relation to contract establishment and execution. These roles are guarantors, advisors and reputation repositories. In addition comes the more general role of intermediaries.

- **Guarantor** is a party taking the responsibility that the obligations of the parties she acts as a guarantor for are fulfilled at an agreed standard. Guarantors assist the establishment or facilitate the increase of trust for a specific transaction by underwriting (a part of) the risk associated with the transaction. A typical example is a credit card company.

- **Advisor** is a party that offers recommendations about the dependability of another party. Advisors include the authorities maintaining blacklists for a community. Examples include, credit scoring authorities and reputation systems.

- **Feedback Collection and Reputation Rating Centre (FCRRC)** gathers feedback about a participant's behaviour over time, enabling a reputation rating to be derived for that participant. This has the potential to be utilised at a number of stages in the contracting process. In the first instance, an Advisor may utilise reputation ratings to assess a potential e-commerce partner prior to the establishment of a contract. The Advisor may have access to a range of external reputation systems, including reputation systems shared by a trusted network of business partners. The Advisor may also have access to an internal Feedback Collection and Reputation Rating Centre, which has gathered specific information about the business's prior interactions with the business partner being assessed.

  Information gathered by the FCRRC may also be taken into account by the Arbitrator in the arbitration decision-making process should a dispute arise, allowing for a more informed decision to be made.

  While certain external reputations systems may be subject to bias and remain open to manipulation by dishonest parties [17], tight controls are inherent in the internal reputation system. A reputation system utilised by a trusted network of businesses may also provide a more trusted source of reputation ratings than external systems. Using Subjective Logic, the Advisor and the Arbitrator may take these variations into account and derive an accurate overall reputation rating for partners in e-business contracts.

- **Intermediary** is a party that intervenes between other parties in a business transaction and mediates so that they establish a business relationship with or without their knowledge. Unlike advisors, an intermediary may also participate in a contract establishment or execution providing access to the services provided by another party, or as a third party representing a service provider. Examples of intermediaries include proxies, information portals, banks who offer to their customers non-financial services such as car rental or flight bookings through an allied service provider, bookshops who offer product delivery by a third party courier as a part of their service. Intermediaries are further classified in [5] with respect to whether they reveal the existence or identity of the party the mediate for.

4. Electronic Contract Enforcement

To have a flexible approach for dealing with deviations from prescribed agreements can serve several purposes. Firstly it provides a set of procedures for handling deviations and settling disagreements. Secondly it serves as an incentive for contractual parties to comply with the agreement because they know that deviations from the agreement will provoke some sort of reaction. Finally it will make the e-marketplace more attractive as an environment for companies to conduct business because discretionary contract enforcement will better suit the needs and requirements of individual companies. The contract enforcement process can be considered as a finite state machine as illustrated in Figure 3 below.

Transition between states is governed by outcomes of previous states. Conceptually the execution of a contract will be situated on one of five possible levels at any one time, as illustrated in Figure 3. The five different levels reflect the degree with which the transaction execution is on compliance with the prescribed agreement. Level 1 reflect full compliance whereas level 5 reflect total transaction failure and the inability to apply corrective measures to the non-compliant party. The states and transitions between states are described in more detail below.

- *Level 1* reflects the fact that the transaction is executed according to the prescribed contract. The monitor observes contract-significant events and evaluates whether the corresponding behaviour pattern was compliant with the contract. Deviations can occur while the execution remains on level 1 as long as notifications to the contractual parties by the Monitor results in the execution getting back on track. After transaction completion, the contractual parties are invited to provide feedback about each others performance. The feedback is collected by the Feedback Collection Centre and is used to derive a reputation rating about each party in the system.

- *Level 2* reflects the fact that the transaction has deviated from the prescribed contract, and warnings to non-compliant parties have been ignored. The Monitor informs the DEM which in turn invokes the Mediator which is closely linked to the Negotiator. The Mediator/Negotiator attempts to establish an amended contract between the two parties. In case of settlement the contract execution returns to Level 1 and resumes with the amended contract as basis. After transaction completion the contractual parties provide feedback about each others performance.

- *Level 3* reflects the fact that the mediation failed, i.e. that the Mediator was not able to make the contrac-

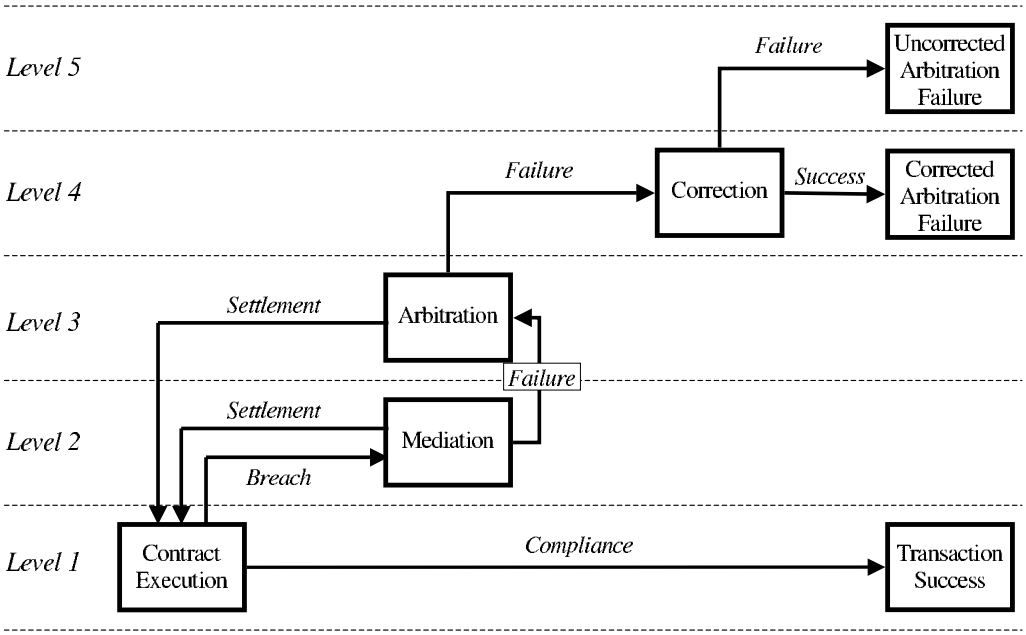


Figure 3. State diagram for enforcing contracts

tual parties agree on an amended contract. The Arbitrator collects all available evidence in order to reach the fairest decision possible. In case the decision by the Arbitrator is accepted by both parties, the contract execution returns to Level 1 and resumes with the arbitrated contract as basis. After transaction completion the contractual parties provide feedback about each others performance.

- *Level 4* reflects the fact that the arbitration decision is not accepted by one or both of the contractual parties. The DEM attempts to apply penalties to the parties it sees as non-compliant. Feedback is not collected from the contractual parties because it is assumed that the hostility between them will make the feedback highly biased and unreliable. Penalties are applied, e.g. in the form of downgraded reputation by the DEM, (i.e. not by the transaction partners.)

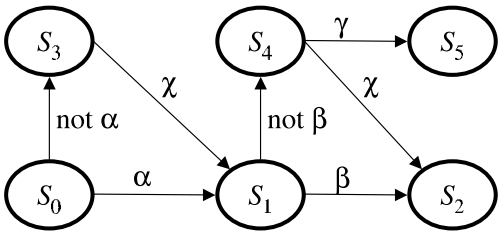
- *Level 5* reflects the fact that the DEM was not able to apply penalties to the non-compliant parties. The suffering parties have the option of initiating legal procedures. This means that the contract execution exits the electronic contractual management and that it enters the realm of the traditional legal system. This might not be practical in many situations (too expensive, slow, uncertainty about which legal domain applies etc.) but it is the last resort.

5. Contract Enforcement: Example

We describe an example with four scenarios of how a service level agreement between an e-commerce Merchant and an ISP (Internet Service Provider) can be enforced. Although the example is superficially simple, it exhibits some features that are typical of lengthier, more complex agreements in various domains. A certain future behaviour is stipulated for the parties, and provision is made in case of deviation from the agreement. The example can serve to illustrate the main points raised in this paper without loss of generality.

In the example, a service level agreement between the Merchant and the ISP expresses that the ISP shall host the Merchant's Web site at the rental charge of $100 per month. The agreed quality of service states, among other things, that unavailability of the Web site due to server down-time shall be maximum 2 hours per month. Rental discounts of $10 apply for every additional 2 hours that the down time exceeds the maximum. A down time exceding 1 day per month is considered contract breach, in which case it is likely that mediation would be needed to settle disputes.

Once the contract has been agreed upon, the contract instance is stored in the Notary, and the contract execution initialised to state S_0. During service execution the Monitor tracks the activities of the Merchant and the ISP fromo month to month, checking their performance and recording the relevant events. Figure 4 below provides a state diagram of possible contract execution scenarios.



Key:
α: service delivery
β: payment
χ: mediation/arbitration
γ: arbitration failure
S_0: contract agreement
S_1: satisfactory service
S_2: payment complete
S_3: unsatisfactory service
S_4: no payment
S_5: correction

Figure 4. State diagram for the Service Level Agreement example

In this example we use Subjective Logic [10] as the basis for reasoning about the evidence from the contract execution and for reaching enforcement decisions. Subjective Logic uses a belief metric called *opinion* to express beliefs about the truth of statements. An opinion is a tuple $\omega_x^A = (b, d, u)$, where the subscript x denotes a particular statement and the superscript A denotes the owner of the opinion about the statement. The parameters b, d and u represent *belief*, *disbelief* and *uncertainty* respectively. These parameters satisfy $b+d+u = 1$ where $b, d, u \in [0, 1]$. This is a sub-additive probability model where the probability of truth (b) and the probability of falsehood (d) not necessarily add up to 1. The advantage of sub-additivity is that it is possible to express degrees of uncertainty regarding the probability of a particular event.

The *discounting* operator denoted by $\otimes$ makes it possible to take second-hand evidence into account, i.e. advice from third parties. The *consensus* operator denoted by $\oplus$ allows the combination of advice coming from different sources. See the Appendix for a detailed description of these operators. The probability expectation value of an opinion is defined as $\mathrm{E}(\omega) = b + u/2$. The calculations in the example have been done with the java Calculate class from the DSTC SL-API repository [8].

The Beta Reputation System [12] is compatible with Subjective Logic in the sense that there exists a bijective mapping between a beta reputation and the opinion metric used in Subjective Logic. When reference is made to an agent's reputation in the scenarios below it can be assumed that it has been generated with the Beta Reputation System and mapped to an opinion.

5.1. Scenario A: Contract Compliance

Assume that the Monitor determines on the basis of first hand evidence (i.e. through its monitoring of the service execution) that the provided service was not according to agreed quality, e.g. that the Merchant's Web site has been unavailable for 4 hours during one month due to down time in the ISP's host servers. As the evidence may be partial (i.e. the unavailability could have other causes than ISP server down-time), the Monitor's confidence on the deviation from the agreed level of quality can be expressed in Subjective Logic notation. In this case we asume this to be: $\omega_{\text{not }\alpha}^{\text{Monitor}} = (0.9,\ 0.0,\ 0.1)$. In the presence of the evidence provided by the Monitor, a warning is sent to the ISP that \$10 needs to be discounted from the monthly charges. The ISP complies with the warning and discounts \$10 from the Merchant's bill. The Monitor puts the contract execution state to S_1, checks that the discounted payment is made and changes the execution state to S_2.

5.2. Scenario B: Contract Breach

Assume that the Merchant complains that its Web site has been unavailable for 2 days, and that he is convinced that this is due to ISP server down-time. The following information is communicated.

$\omega_{\text{not }\alpha}^{\text{Merchant}} = (0.9,\ 0.0,\ 0.1)$	The Merchant's opinion that the ISP server was down.
$\omega_{\text{not }\alpha}^{\text{ISP}} = (0.0,\ 0.8,\ 0.2)$	The ISP's opinion that the the ISP server was down.

The following opinions of the Mediator relate to the reliability of the ISP and the Merchant based on past experience (stored in FCRRC) and their role in the transaction.

$\omega_{\text{ISP}}^{\text{Mediator}} = (0.3,\ 0.1,\ 0.6)$	The ISP's reputation.
$\omega_{\text{Merchant}}^{\text{Mediator}} = (0.8,\ 0.0,\ 0.2)$	The Merchant's reputation.

The Mediator combines the existing evidence and forms the opinion:

$$\begin{aligned}\omega_{\text{not }\alpha}^{\text{Mediator:(ISP,Merchant)}} &= (\omega_{\text{ISP}}^{\text{Mediator}} \otimes \omega_{\text{not }\alpha}^{\text{ISP}}) \oplus (\omega_{\text{Merchant}}^{\text{Mediator}} \otimes \omega_{\text{not }\alpha}^{\text{Merchant}}) \\ &= (0.662,\ 0.081,\ 0.257). \end{aligned} \quad (1)$$

The probability expectation value is given by $\mathrm{E}(\omega_{\text{not }\alpha}^{\text{Mediator:(ISP,Merchant)}}) = 0.791$. Assuming that a minimum value of 0.7 is defined as a threshold for the Mediator to pronounce a decision, the conclusion can be drawn that a contract breach has taken place, and that the contract execution now is in state S_3.

The Merchant claims that he has lost business worth a substantial amount and seeks $1000 compensation from the ISP. The ISP argues that the unavailability probably was due to some denial of service attack for which it could not be blamed, and propose not to charge any rent for that month.

The Mediator checks the instances of the contract stored in the Notary and determines that such a deviation has not been considered in the contract. The case is then referred to the Mediator implementing a Level 2 contract enforcement process.

The Mediator assesses alternative contract amendments seeking advise based on FCRRC data and (potentially recommendations from external advisors) about the competence of the parities involved and the effectiveness of different candidate amendments. Finally, it decides to reject the Merchant's suggestion as disproportional to the deviation and delegates to the Negotiator the brokerage of a deal constituting an amendment to the existing contract. Free rental for two additional months is suggested in order to initiate the negotiation of the contract amendment. The ISP rejects this suggestion and makes a counter offer of free rental for one additional month, which is subsequently rejected by the Merchant. Finally the ISP and the Merchant agree on free Web site rental for one additional month and 50% discount for the subsequent month.This can be regarded as an additional clause to the contract and can reference the original contract.

With the Merchant and the ISP having reached a mutually acceptable amendment, which has been validated by the Validator, the Negotiator closes the amendment negotiation and informs Mediator who decides that a settlement has been reached. The contract execution is in state S_1 and back on track with an amended contract stored at the Notary.

5.3. Scenario C: Mediation Failure

Assume that the Merchant complains that a contract breach has taken place because down time has exceeded 1 day, and communicates this by means of an opinion $\omega_{\text{not }\alpha}^{\text{Merchant}} = (0.9,\ 0.0,\ 0.1)$. The ISP on the other hand claims that based on first hand evidence α has been met. The ISP communicates this information to the Monitor by means of an opinion about 'not α' expressed by $\omega_{\text{not }\alpha}^{\text{ISP}} = (0.0,\ 1.0,\ 0.0)$.

As the ISP and the Merchant have highly conflicting opinions about the state of the contract, mediation is not possible and the Arbitrator is invoked. In the presence of a potential dispute, the Arbitrator has to weigh its own evidence and consider the recommendations from its advisors, in addition to the information sent by the ISP and the Merchant in order to form a judgement. We assume that the Arbitrator has formed an opinion $\omega_{\text{not }\alpha}^{\text{Self}} = (0.3,\ 0.0,\ 0.7)$ on the basis of its own evidence and that the Advisor sends recommendation $\omega_{\text{not }\alpha}^{\text{Advisor}} = (0.2,\ 0.6,\ 0.2)$.

In order for the Arbitrator to weigh the evidence received he must have formed an opinion about the reliability of the evidence-providers for that particular purpose. Let this be expressed as:

$$\omega_{\text{Self}}^{\text{Arbitrator}} = (0.9,\ 0.0,\ 0.1) \quad \text{Arbitrator's trust in itself}^4$$

$$\omega_{\text{Advisor}}^{\text{Arbitrator}} = (0.8,\ 0.1,\ 0.1) \quad \text{Arbitrator's trust in Advisor}$$

$$\omega_{\text{ISP}}^{\text{Arbitrator}} = (0.3,\ 0.1,\ 0.6) \quad \text{Arbitrator's trust in ISP}$$

$$\omega_{\text{Merchant}}^{\text{Arbitrator}} = (0.8,\ 0.0,\ 0.2) \quad \text{Arbitrator's trust in Merchant}$$

The following formula encodes the opinion formation process of the Arbitrator, where all evidence is first weighed against trust in the corresponding evidence provider and then combined equally and fairly.

$$\begin{aligned}\omega_{\alpha}^{\text{Arbitrator:(Self,Advisor,ISP,Merchant)}} &= (\omega_{\text{Self}}^{\text{Arbitrator}} \otimes \omega_{\text{not }\alpha}^{\text{Self}}) \oplus (\omega_{\text{Advisor}}^{\text{Arbitrator}} \otimes \omega_{\text{not }\alpha}^{\text{Advisor}}) \oplus \\ &\quad (\omega_{\text{ISP}}^{\text{Arbitrator}} \otimes \omega_{\text{not }\alpha}^{\text{ISP}}) \oplus (\omega_{\text{Merchant}}^{\text{Arbitrator}} \otimes \omega_{\text{not }\alpha}^{\text{Merchant}}) \\ &= (0.551,\ 0.287,\ 0.163)\end{aligned} \tag{2}$$

The probability expectation value is given by $\mathrm{E}(\omega_{\text{not }\alpha}^{\text{Arbitrator:(Self,Advisor,ISP,Merchant)}}) = 0.633$. The probability expectation value may be used to make an informed decision that the claim 'not α' is valid and hence put the state of the contract execution to S_3. Of course, it is very difficult to decide what "the balance of probabilities" should be! In the example above, it is clear that should the Arbitrator decide in favour of 'not α' it would mean that the threshold value is less than 0.633. (Any value above 0.5 could reasonably be considered.) However, if the threshold value is greater than 0.633, then the Arbitrator needs to make another decision: either support the ISP, as the Merchant's assertion was not proved on "the balance of probabilities", or seek further evidence, if available.

As an example, we assume that the ISP server down-time is in dispute with $\mathrm{E}(\omega_{\text{not }\alpha}^{\text{Arbitrator:(Self,Advisor,ISP,Merchant)}}) = 0.633$ and that the threshold for imputing blame on the ISP is 0.7. The Arbitrator decides to consult (seek a recommendation from) the ISP's external security Auditor, which it consid-

[4] As explained in [6], an agent is aware of its degree of trust in itself. Self-assessment underlies an agent's ability to seek external advice and to delegate or offer a task to another agent, so as to improve efficiency or reduce risk.

ers to be a reliable source of information, i.e. $\omega_{\text{Auditor}}^{\text{Arbitrator}} = (0.9,\ 0.0,\ 0.1)$. The Auditor offers the opinion, with great certainty, that the ISP server was in operation without down-time for the whole month, and supports the ISP with $\omega_{\text{not }\alpha}^{\text{Auditor}} = (0.0,\ 0.9,\ 0.1)$. As a consequence the Arbitrator updates its opinion to $\omega_{\text{not }\alpha}^{\text{Arbitrator:(Self,Advisor,ISP,Merchant,Auditor)}} = (0.325,\ 0.579,\ 0.096)$ with the probability expectation value $\mathrm{E}(\omega_{\text{not }\alpha}^{\text{Arbitrator:(Self,Advisor,ISP,Merchant,Auditor)}}) = 0.373$. Hence, the Arbitrator decides that α has been fulfilled and that the contract execution is in state S_1. The Contract Arbitrator signals the decision, via the Negotiator, to the Merchant, asking him to comply by executing β. Under the pressure of the arbitration decision the Merchant finally pays, sending the execution to its final state S_2.

5.4. Scenario D: Arbitration Failure

We consider the same scenario as in scenario C above, except that this time the Merchant refuses to pay, sending the execution to state S_4. We assume that the Arbitrator has decided that the ISP provided the service with agreed quality, but that the Merchant refuses to accept the arbitration decision. The Discretionary Enforcement Moderator can do nothing more to broker an agreement between the ISP and the Merchant, sending the execution into state S_5. The Contract Enforcer will try to apply some form of penalty or correction to the party it considers to be at fault. The options can for example be to downgrade the Merchant's reputation or to revoke his authorisation to conduct business within the BCA. The ISP can of course cancel the contract with the Merchant.

6. Conclusion and Future Work

We have presented a range of options for supporting enforcement in contract management systems. The premise of this paper is that deviations from contract occur sufficiently often to deserve the investigation of possible automated solutions to facilitate dealing with such deviations. Our belief is that human will always be in the decision loop, but the sheer amount of contracts and their link to other parts of enterprise systems calls for an increased level of their automation. One of the problem that need to be addressed to achieve this is to provide certain level of determination about parties performance to the agreement and based on that to undertake appropriate corrective measures.

Our approach to this particular problem is based on crisp separation of roles that represent various enforcing options as part of an overall (role-based) contract management architecture. The roles presented in this paper are contract monitor, notifier, discretionary enforcement moderator (with its mediator and arbitrator role options) and non-discretionary contract enforcer. The paper also shows a way of providing quantitative assessment of parties behaviour based on second-hand evidence and by adopting an approach based on Subjective Logic. We used the previously developed Business Contract Architecture (BCA) as an example architecture to describe this value added functionality.

The BCA roles described are specified in a platform independent manner so that the BCA can be implemented using any specific technology available (e.g. J2EE, .Net, CORBA etc). We use Web Services standards to implement BCA components such as Notary, Monitor and the DEM components.

There are several related research topics that we plan to investigate in future. One problem is determining legal validity of contracts after it has been drafted, in particular the legal purpose of the contract, namely its compliance with the rules of an outer legal system. Another problem is ensuring the completeness of contracts to minimise possible situations where contracts are 'silent', i.e. do not cover all possible conditions of interest. Finally, we are planning to investigate feasibility of automated negotiation for contracts, which also need to rely on sound validation mechanisms.

Appendix: Subjective Logic Operators

The Subjective Logic operators used in the example of Section 5 are the *discounting* operator which supports the incorporation of second-hand evidence, and the *consensus* operator which allows the combination of evidence from different sources. The interpretations of, and the algebraic expressions for these operators are given below. A more complete description of these and other operators of Subjective Logic can be found in [10]. See [8] for online demonstrations and resources.

- *Discounting.* Assume two agents A and B where A has an opinion about B (i.e. that "B *is knowledgeable and will tell the truth*") denoted by ω_B^A. In addition B has an opinion about a proposition x, denoted by ω_x^B. Agent A can then form an opinion about x by discounting B's opinion about x with A's opinion about B, denoted by $\omega_x^{A:B}$. By using the symbol '$\otimes$' to designate this operator, we define $\omega_x^{A:B} = \omega_B^A \otimes \omega_x^B$.

- *Consensus.* The consensus of two possibly conflicting opinions is an opinion that reflects both opinions in a fair and equal way. Let ω_x^A and ω_x^B be A's and B's opinions about the same proposition x. The opinion $\omega_x^{A,B}$is then called the consensus between ω_x^A and ω_x^B, denoting an imaginary agent $[A, B]$'s opinion about x, as if she represented both A and B. By using the symbol '$\oplus$' to designate this operator, we define $\omega_x^{A,B} = \omega_x^A \oplus \omega_x^B$.

	Discounting: $\omega_x^{A:B} = \omega_B^A \otimes \omega_x^B$	**Consensus:*** $\omega_x^{A,B} = \omega_x^A \oplus \omega_x^B$
Belief:	$b_x^{A:B} = b_B^A b_x^B$	$b_x^{A,B} = \frac{b_x^A u_x^B + b_x^B u_x^A}{u_x^A + u_x^B - u_x^A u_x^B}$
Disbelief:	$d_x^{A:B} = b_B^A d_x^B$	$d_x^{A,B} = \frac{d_x^A u_x^B + d_x^B u_x^A}{u_x^A + u_x^B - u_x^A u_x^B}$
Uncertainty	$u_x^{A:B} = d_B^A + u_B^A + b_B^A d_x^B$	$u_x^{A,B} = \frac{u_x^A u_x^B}{u_x^A + u_x^B - u_x^A u_x^B}$

Remark: *Limits can be computed [11] for $u_x^A = u_x^B = 0$.

Table 1. Overview of the Subjective Logic operators used in this paper

The complete definition of these as well as other operators can be found in [10]. In addition to the parameters *belief*, *disbelief* and *uncertainty*, the operators can also contain a parameter called *relative atomicity*. In the example of Section 5 the relative atomicity is not relevant and has therefore been omitted. When using the java Calculate class from the SL-API for the numerical calculations the relative atomicity was set to the default value of 0.5. See [8] for online Subjective Logic demonstrations and SL-API resources.

References

[1] T. Allen and R. Widdison. Can Computers Make Contracts? *Harward Journal of Law and Technology*, 9(1), 1996.

[2] A. Carblanc. Privacy protection and redress in the online environment: Fostering effective alternative dispute resolution. In *Proceedings of the 22nd International Conference on Privacy and Personal Data Protection*, Venice, September 2000.

[3] A. Daskalopulu, T. Dimitrakos, and T. Maibaum. Evidence-Based Electronic Contract Performance Monitoring. *INFORMS Journal of Group Decision and Negotiation, Special Issue: Formal Modeling of Electronic Commerce*, Spring 2002.

[4] C. Dellarocas, M. Klein, and J. Rodrigues-Aguilar. An exception-handling architecture for open electronic marketplaces of contract net software agents. In *Proceedings of the 2nd ACM Conference on Electronic Commerce*, pages 225–232, Minneapolis, 2000.

[5] T. Dimitrakos. System Models, e-Risk and e-Trust. In *Towards the E-Society: E-Business, E-Commerce, and E-Government*. Kluwer Ac. Pub, 2001. (1st IFIP Conf. on e-business, e-commerce, e-government).

[6] T. Dimitrakos. Towards a Formal Model of Trust in e-Commerce. In *Proceedings of the Novel E-COmmerce Applications of Agents Workshop, AI2001*, pages 13–22, Ottawa, Canada, 2001. NRC-44883.

[7] T. Dimitrakos and J. Bicarregui. A Framework for Managing Trust in e-Services. In 4^{th} *International Conf. on Electronic Commerce Research*, pages 360–381. ISBN 0-9716253-0-1, 2001. ATSMA, IFIP, INFORMS vol. 2.

[8] DSTC. Subjective Logic demo and Subjective Logic API. http://security.dstc.edu.au/spectrum/.

[9] J. Ibbotson and M. Sachs. Electronic Trading Partner Agreement for E-Commerce. Technical report, IBM Corporation, 1999.

[10] A. Jøsang. A Logic for Uncertain Probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–311, June 2001.

[11] A. Jøsang. Subjective Evidential Reasoning. In *Proceedings of the International Conference on Information Processing and Management of Uncertainty (IPMU2002)*, Annecy, France, July 2002.

[12] A. Jøsang and R. Ismail. The Beta Reputation System. In *Proceedings of the 15th Bled Electronic Commerce Conference*, Bled, Slovenia, June 2002.

[13] M. Merz, F. Griffel, T. Tu, S. Müller-Wilken, H. Weinreich, M. Boger, and W. Lamersdorf. Supporting electronic commerce transactions with contracting services. *International Journal on Cooperative Information Systems*, 7(4), 1998.

[14] Z. Milosevic. *Enterprise Aspects of Open Distributed Systems*. PhD thesis, Computer Science Dept. The University of Queensland, October 1995.

[15] Z. Milosevic, D. Arnold, and L. O'Connor. Inter-enterprise contract architecture for open distributed systems: Security requirements. In *Proceedings of WET ICE'96 Workshop on Enterprise Security*, Standford, June 1996.

[16] Z. Milosevic and A. Bond. Electronic Commerce on the Internet: What is Still Missing? In *Proc.of the 5th COnf. of the Internet Society*, Honolulu, June 1995.

[17] M. Patton and A. Jøsang. Technologies for Trust in E-Commerce. In *Proceedings of the IFIP working conference on E-Commerce*, Salzburg, Austria, June 2001.

[18] D. Reeves et al. Towards a declarative language for negotiating executable contracts. In *Proceedings of AAAI-99 Workshop on Artificial Intelligence in Electronic Commerce*, 1999.

[19] P. Resnick et al. Reputation systems. *Communications of the ACM*, 43(12):45–48, December 2000.