A Formal Analysis of Auditing Principles for Electronic Trade Procedures

Roger W. H. Bons
Frank Dignum
Ronald M. Lee
Yao-Hua Tan

ABSTRACT: One of the major issues involved in establishing new trading relationships is the lack of an a priori trust relationship between the parties. Without a solution of this trust problem it is virtually impossible to establish new trade relationships. This is an old and well-known problem in international trade. One way to create the necessary trust is by using trade procedures that involve exchanges of documents between the trading partners to verify that the counter party has fulfilled his part of the agreement. Up till now these documents were paper documents. In electronic commerce these paper documents are replaced by information exchanges. One crucial question, however, is whether these documents provide the same trustworthiness as the paper documents. For example, uniqueness of ownership documents is simple to guarantee in the case of a signed paper document. However, uniqueness of electronic messages is much harder to achieve. Electronic commerce can only become a success in international trade if the trustworthiness of electronic versions of trade procedures can be proven beyond reasonable doubt. We present the formal specification of the design principles for trustworthy trade procedures, building on the work on audit daemons [17, 18] and their application in [3] on inter-organisational trade procedures. It appears that the basic principles of internal auditing within a company can be adapted for this auditing method. This auditing method has already been implemented in INTERPROCS, which is a Prolog-based tool for representation and analysis of procedures. In the formal specification given in this paper we give a more abstract specification of the auditing method, which gives a more structured semantics for the key concepts of the auditing method used in INTERPROCS. Our formal specification is based on a combination of deontic, dynamic and illocutionary modal logics.

KEYWORDS AND PHRASES: Trade procedures, Open-EDI, Open electronic commerce, Automated auditing, Formal specification, Deontic logic.

Introduction: Open Electronic Commerce.

Modern businesses are confronted with an increasing need to deliver customer-specific products and services at a competitive level of costs. This poses high requirements on the company's ability to organise its contacts with both suppliers and customers in a flexible, efficient and effective way. Electronic commerce tools have shown to be essential to achieve this goal. For example, many success stories are known in which Electronic Data Interchange (EDI) has been employed to reduce the transaction costs between companies. Also, the Internet has become a serious source for information, both in a business-to-consumer and a business-to-business environment. Furthermore, electronic commerce techniques offer the possibility to conduct business at a global level. New business opportunities are easily found, regardless of the geographical location of the companies involved, possibly between partners that have never done business with each other before. This is referred to as 'open electronic commerce': electronic commerce between partners who do not have an established business relationship and who may do business with only a minimal level of mutual trust [4].

In this paper the design of trustworthy trade procedures for open electronic commerce is discussed, dealing with the issue of the lack of trust between (electronic) trading partners. Most of the successful applications of EDI and electronic commerce have so far been realised between partners that have a high level of mutual trust and have a long-lasting relationship involving many transactions. In such a setting the only information that has to be exchanged among the parties is for the synchronisation of each other's operational activities. Whenever a dispute arises, parties strive towards a compromise as they do not want to jeopardise their long-standing relationship. However, when parties do not have such a relationship they want to exchange additional information about each other's behaviour. In other words, when they do not (fully) trust each other, they want to rely on a trade procedure that governs their activities at a minimal level of risk.

The outline of this paper is as follows. First, the need for such procedures in the context of open electronic commerce is discussed in more detail and it is argued why a formal theory is needed. Then a set of design principles for these procedures, based on [3] is presented. Third, a (deontic) logic, which provides a formal representation for these design principles is sketched. The paper is concluded and some areas for future research are given.

Trustworthy Trade Procedures.

Trustworthy trade procedures have been a necessity since people started trading. In the early days of trade the exchange of goods took place following a so-called 'handshake' protocol; i.e. direct exchange of goods and payment. A protocol said to be still used in the dealing of narcotics. The overseas trade in the 15th and 16th century took place in this manner. Traders would usually accompany their goods to the overseas trading partner. As the scale of the trade kept growing, it became no longer feasible to conduct all business in simultaneous exchanges [28]. Thus a separation, both in place and in time, took place between the parties.

This separation led to the introduction of new agents in international trade, such as the captain of a ship and merchant banks who pre-financed the transport. This has lead to complex inter-organisational procedures involving many different parties and the exchange of control information by standard types of paper documents. For instance, shipping a container with chemical products from Germany to the United States may involve over 40 different parties and over 100 different types of documents (see [3]).

Throughout history new developments in trade have required the development of new trade procedures, usually through many years of trial and error. One of the best known examples of such a procedure is the documentary credit procedure, which was developed over 200 years ago and manages the trading risk between overseas buyers and sellers that do not completely trust each other. There are three reasons to develop a theory on trustworthy trade procedures in the context of open electronic commerce.

The first reason is that existing control instruments may be replaced by more efficient and effective electronic equivalents. The current implementation of inter-organisational controls is mainly based on a costly exchange of paper documents. The delays caused by the relatively slow physical transport of such documents are a disruptive factor, as well as the transcription errors that are introduced from transferring data from one paper document to another. Especially with control information, it is crucial to find out as early as possible when something is wrong in order to be able to limit damages (e.g., port of destination vs. port of origin). The difficulties that can arise in this exchange of control information are illustrated by a study of the UK SITPRO organisation [24], which showed that still up to 50% of the control documents needed for the documentary credit procedure contain errors when they are first presented to the controlling bank. The introduction of electronic exchanges of control information can reduce these costs tremendously, provided that such exchanges can be performed in a secure fashion.

The second reason for addressing trustworthy trade procedures is that the introduction of (open) electronic commerce will lead to new ways of doing business for which the existing control structures will no longer work. A distinction has to be made between physical goods and virtual goods such as music, videos, books, software and so forth. Transactions involving the exchange of physical goods can be supported by electronic commerce as information becomes available more timely and accurately, for instance through tracking and tracing systems and *Electronic Funds Transfer* (EFT). For virtual products, even more revolutionary changes will occur as the actual physical delivery can take place electronically. Instead of sending a compact disc through some mail service, it becomes possible to download the music from the record company's server and put it on some kind of storage device locally. If, as Wagenaar claims [28], electronic commerce will be the next evolutionary step in world trade, it is inevitable that new trade procedures will have to be developed. Electronic commerce implies no direct physical observation of each trading partners' activities, but at the same time enables the establishing of new trading relationships. Therefore, more controls have to be imposed on the trade procedures in order to prevent fraud in comparison with the past. On the other hand, there is now ample opportunity with information and communication technology to build in such controls.

Finally, new control mechanisms have to be developed to facilitate the entire range of transactions through electronic means. Currently, the bulk of on-line payments in electronic commerce is done with credit card, and even the credit card is being redesigned for a secure use through the Internet[1]. However, it is very unlikely that high value business-to-business transactions (values over, say, 50000 US dollars, which are traditionally governed by a documentary credit procedure in case of minimal trust) will ever be paid through a credit card transaction, because, first, due to the fee structure of approximately 3% of credit card payments, this is too costly, and secondly, in business-to-business commerce payments are done using purchase orders, invoices and bank transfers etcetera. Hence, in particular electronic versions of purchase orders are important for this type of transactions, but are not provided in the credit card payment scheme. Trustworthy trade procedures will have to be developed for these business-to-business transactions, with possibly different implementations of the control measures.

In open electronic commerce the number of different trading relationships entered and trade procedures used to govern these relationships will be quite high. The parties involved have to be able to assess whether the procedure they are about to use is trustworthy or not. In other words, an 'on the fly' analysis of these trade procedures should be feasible, supported by some automated auditing tools. Such tools may be used by humans, assuming that their system alerts them when a new procedure is going to be used. More advanced systems may have fully integrated the systems running trade procedures and the automated auditing module, delegating the decision whether or not to engage in a certain trade procedure to their information system. This automated reasoning about trustworthy trade procedures requires a formal representation of these procedures.

Design Principles for Trustworthy Trade Procedures.

In [3, 4] a framework for analysing trustworthy trade procedures is developed which identifies three steps. In the first step, it is determined whether the proper control documents are exchanged among the right parties. In the second step of the analysis, the technical requirements are identified that should be satisfied by the underlying communication system used to exchange these control documents. Finally, it is analysed whether or not the actual systems in use by the parties involved do satisfy these requirements. In this paper the focus will be on the first level of analysis, i.e., the question whether or not the right types of documents have been exchanged.Managing the risk of non-performance of the core activities involved in a trade transaction is the most important goal of this analysis.

. These core activities are called ‘primary activities’ in the remainder of this paper. They are defined as those activities trading partners have to perform for one another as specified in the contract governing the transaction at hand, corresponding with ‘primary obligations’ in legal literature.

The exchange of so-called ‘performative’ documents is one of the instruments to control this risk and several functions of such documents can be specified using Speech Act theory [2, 23]. Examples of performative functions are ‘testify’ and ‘promise’, which refer to a statement of a party that it has witnessed the completion of an activity and the firm promise that a party will perform some future action, respectively.

The following set of principles were defined to analyse the use of performative documents in establishing a trustworthy trade procedure. It is beyond the scope of this paper to discuss the theoretical foundations for this set. In [3] these rules are derived from various bodies of literature, including accounting, law and trade practices. Note that the term ‘role’ is used to denote a cluster of activities that is performed by a party. Typical examples of roles in international trade ‘shipper’, ‘bank’, ‘carrier’ and so forth.

General Principle I:
If a primary activity is performed by Role 1, then Role 2 should testify the completion thereof using some document, which should be received by Role 1. If the party playing Role 2 is not trusted by the party playing Role 1, the primary activity should be executed after receiving the document.

General Principle II:
Before Role 1 executes a primary activity it should have witnessed the performance of the counter-activity by some Role 2, if the party playing Role 1 does not trust the party responsible for Role 2, unless it has received evidence that Role 2 has executed its tasks.

General Principle III:
If Role 1 cannot witness the performance of a counter-activity by some Role 2, then another Role 3 should testify the completion of Role 2’s activity, if the party playing Role 2 is not trusted by the party playing Role 1. The testify document issued by Role 3 must be received by Role 1 before the execution of its primary activity, and the party performing Role 3 should be trusted by the party performing Role 1.

General Principle IV:
If a primary activity is outsourced to a Role 1 and the principal Role 2 did not previously witness the counter-performance or receive evidence thereof, the agent Role 1 should witness this counter-performance before it performs the (outsourced) primary activity if the principal does not trust his counter-party. If this is not

possible, the agent Role 1 should at least receive evidence of the counter-performance.

General Principle V:
If the counter-activity by Role 2 to some primary activity of Role 1 consists of only the enabling actions of Role 2 to arrange some third agent Role 3, and not the agent's performance as well, and the party that plays Role 1 does not trust the party that plays Role 2 and has not previously witnessed the counter-activity, then Role 1 should receive an unambiguous promise from Role 3 that it will be the beneficiary of Role 3's performance before it executes its own primary activity. Furthermore, the party performing Role 1 should trust the party that plays Role 3.

In [3] several cases from the domain of international trade have been analysed to validate the correctness and practical relevance of these rules. In one case various international payment methods were analysed based on these general principles. These methods typically differ in the security that is offered to buyers and sellers. For instance, the so-called 'open account post-payment' method, in which the buyer gives his bank an instruction to pay the seller after the buyer has received the goods, does not offer any security for the seller. In terms of the principles, the second general principle is not fulfilled as the seller can not observe the buyer's performance of the payment (delivering the goods and paying the price are counter-activities, to be performed by the parties to fulfill the underlying contract). On the other hand, the aforementioned documentary credit procedure does offer this security since it involves banks that guarantee to pay if the seller presents them sufficient evidence about the shipping of the goods. In [3] it is demonstrated that these differences are made visible by applying this initial set of design principles, proving the relevance of the approach in general and this sample set of principles in particular. It goes beyond the scope of this article to discuss these cases in more detail.

In the remainder of this paper it is shown that automated reasoning about trustworthy trade procedures, based on these type of principles, can be performed using a form of deontic logic.

Logics for Trade Procedures

It is important to note that we have to model three components of the trade procedures:

1. The trade procedure itself.
2. The underlying contracts (e.g. method of shipment and payment in terms of international agreed standards (INCOTERMS)).[3]
3. The audit rules that check the trustworthiness of the procedure given the underlying contracts.

In [3, 17, 18] the components are in fact translated into Prolog, which has the great advantage that one can actually perform the checks automatically. However, Prolog requires that everything is represented in first-order Horn clause logic, and that is not always the most natural format to represent all relevant dynamic and deontic aspects of contracts. Of course, in Prolog it is possible to write predicates as objects again and thereby introducing some nesting of formulas. However, this is usually not (meant to) exceed more than one level of nesting. Therefore it will be difficult to represent nestings of modal operators in a natural way. In the end it will almost always be possible to device an implementation in Prolog. However, it is usually very difficult to show the correctness of the program and that it indeed implements the intended logic. (Note that most nice properties of logic programs don't hold anymore as soon as the predicate to object mapping has been applied!).

In this article we present a modal logic which might serve as an intermediary between the natural language specifications and the Prolog implementation. All three components can be modelled in the same formalism and therefore formally checked at a higher level. Looking at the principles specified above it is clear that the logic should be able to express at least the following elements:

1. Obligations between two parties and general obligations
2. Actions (especially communicative actions)
3. Temporal relationships
4. Knowledge and belief
5. Outsourcing and delegation
6. Trust

Clearly, there is already a logic to perform automated auditing, namely the Prolog-based INTERPROCS implementation that is described in [20, 21 and 3]. What the formalisation presented in this paper adds to that perspective is a more abstract formal specification of the auditing process. The advantage of modal operators compared with first-order predicates is that it is possible to characterize the properties of a modal operator using a range of (well-known) axioms for modal operators. For example, in general one can assume that knowledge is closed under the modus ponens inference rule; i.e. if A is known and from A follows B, then B is also known. However, there are special cases where this axiom does not make sense, and where one would prefer a more elaborated axiom that takes into account the difference between explicit and implicit tacit knowledge. For example, one might have tacit knowledge how to ride a bike, but very few people are able to make explicit inferences from this type of knowledge. A formal analysis can often give a new insight in a theory by making underlying assumptions explicit that are hidden behind the fundamental vocabulary of the theory. In particular, we argue in this paper that there are fundamental axioms about the way people communicate with each other that can give a deeper understanding of the general audit principles. This explanation is based on an investigation of the axioms that hold for the modal operators that we use in this paper to represent these general audit principles.

However, there is a trade-off here. Modal logics might give richer formal specifications than first-order logic, but modal logics are also harder to implement than first-order logics for which already various types of Prolog programming languages have been developed. The issue here is a trade-off between expressiveness and ease of implementation. What we advocate is an approach in which it is possible to formally specify the principles of automated auditing of trade procedures in modal logic, and then find canonical ways to translate these specifications into other formalisms that are easier to implement such as the INTERPROCS implementation. In this way one can have the best of both worlds: very rich and expressive modal logic specifications and at the same time efficient Prolog implementations.

Although the concepts of delegation and trust are fundamental for auditing trade procedures, we use in this paper predicates to describe these concepts without giving full-blown formal definitions for these predicates. The idea is that they can be seen as a black box that can be filled in when the concepts can be described appropriately. We will see that the general audit principles usually presuppose that the parties do *not* trust each other. In these cases it is more important to specify how the procedures should be designed in order to make them trustworthy rather than to describe exactly what is meant by the term ‘trust’. A more detailed explanation of the notion of trust, that underlies the auditing method discussed here, can be found in [3 , Ch. 2], where trust is analysed in relation to transaction cost theory. A more generic model of trust in electronic commerce can be found in [27]. Furthermore, a formal model of fraud is proposed in [14]. The basic idea of this model is that fraud is understood as creating a false belief in another agent that you performed an action, while you actually know that you have not performed the action. Clearly, trust is related to the belief that such fraud situations will not occur.

Epistemic Dynamic Deontic Temporal Logic

The logic consists of a combination of four types of logic: standard deontic logic [29, 1] combined with dynamic logic [16] gives us a variant of dynamic deontic logic as defined in [22]. In [7] we added temporal logic in order to specify deadlines and here we also add a knowledge operator to indicate what both parties (should) know of each other when performing a transaction. In deontic logic we can specify obligations, prohibitions and permissions. There are two basic variations: obligations on conditions and obligations on actions. We use the obligation on conditions as a basis. Subsequently, we define the obligations on actions by adding a dynamic logic component.

The directed obligations are of the form $O_{ij}(p)$, which means intuitively that agent i is obliged towards agent j that p holds. For example, $O_{buyer,seller}$(goods_paid) means that the buyer has the obligation towards the seller to pay for the goods. The main advantage of deontic logic is that we can have the situation in which agent i has the obligation to pay for the goods, but has not paid it yet. This can be represented in the deontic formula O_{ij}(goods_paid)∧¬goods_paid. This formula is not inconsistent, but indicates a non-ideal situation. For further technical details about directed obligations see [25]. General obligations are denoted by O(p), which have the same form but without the subscripts, and mean that it is ideal if p holds. In this case there is no specific agent responsible to make p to hold, and there is also no specific beneficiary; e.g. O(¬fraud) means that in the ideal situation there is no fraud.

Actions play a prominent role in trade procedures. Therefore we need a formalism that can adequately describe them. In particular, the formalism should be able to combine different actions and also describe the effects of the actions. We have chosen dynamic logic for this purpose. One of the reasons for this choice is that there exists already a well-known logic that combines dynamic and deontic aspects [22].

In dynamic logic we specify what happens if an action is performed. The formula $[\alpha]p$ means that the performance of action α leads to a state where p holds. We assume that the action is performed instantaneously, i.e. we cannot express that an action has a specific duration. The performance of an action can be compared to the update of the state of the world. For example, [pay($100)]bank_account<0 means that if $100 is paid at this moment, then the bank account is negative at the next moment after the payment has been done (there was less then $100 in the account). Note that we only give a conditional statement! *If* the action is performed, then it will have some result. It is not claimed that the action will actually be executed next! To specify that an action will actually be performed as the next action we use the formula $DO(\alpha)$. So, DO(pay(John,$100)) means that John will pay $100 at the next moment.[4] In this logic we can also specify that an action has just been performed with the formula DONE(α). For example, DONE(pay(John,\$100)) says that John has just paid \$100. Both the DO and the DONE operator can be defined using the temporal operators from temporal logic.

In order to describe the procedures adequately, we have to specify for each action who performs it. In particular, the actions are parameterised with the roles involved in the action. Dynamic logic also allows us to specify combinations of actions. More specifically, we can express the parallel execution of actions α and β by the formula $\alpha\&\beta$, a choice between α and β by we express by the formula $\alpha\cup\beta$, and an action sequence of α followed by β by the formula $\alpha;\beta$. In addition to these combinations of actions we use the action **any** to denote a choice of any possible (combination of) action. So, [**any**]p means that whatever happens next p will always hold afterwards.

Speech Acts and Epistemic Logic

In [10] we used the above logical framework to model communications between information systems. We introduced special *speech acts* which have standard types of results. For example, an *order* to deliver results in an obligation to deliver (if all pre-conditions hold). In this paper we also use this type of special actions, but follow as close as possible the descriptions from [3]. This leaves us with the following speech acts:

promise(r_1,r_2,α) : r_1 promises to r_2 that she will perform the action α
direct(r_1,r_2,α) : r_1 orders r_2 to perform the action α
testify(r_1,r_2,p) : r_1 testifies to r_2 that p is true

These speech acts do not just communicate some information but also change the state of the world. The first two create obligations the last one creates knowledge. Specifically, we have the following axiom:

$$[\mathrm{promise}(r_1,r_2,\alpha)]O_{r1r2}(\alpha)$$

i.e. after r_1 promises to perform α to r_2, then r_1 is obliged towards r_2 to actually perform the action α. Analogously, we have the axiom:

$$[\mathrm{direct}(r_1,r_2, \alpha)]O_{r2r1}(\alpha),$$

i.e. after r_1 orders r_2 to perform α, r_2 is obliged towards r_1 to perform the action. Of course, this will not be true in general for any two agents. There should at least exist some relation between r_1 and r_2 prior to the order that allows r_1 to order r_2 around. This can be an existing contract, or a working relationship between r_1 and r_2. We do not specify these preconditions in this paper for simplicity. The interested reader is referred to [10] for more details. The last type of speech act, the *testify*, creates (new) knowledge in the hearer. The fact that a person i knows the fact p is denoted by $K_i(p)$. For the testify speech act we assume the following axioms to hold:

$$[\mathrm{testify}(r_1,r_2,p)]K_{r2}(p) \wedge K_{r2}(K_{r1}(p)) \wedge G\neg DO(\mathrm{testify}(r_1,r,\neg K_{r1}(p))),$$
$$DO(\mathrm{testify}(r_1,r_2,p)) \rightarrow K_{r1}(p),$$

i.e. after r_1 has testified p towards r_2, then r_2 actually knows p. This assumes of course that r_1 can only testify p in case p is true. Besides the fact that r_2 knows p, r_2 also knows that r_1 knows p. Finally, by testifying p, r_1 also 'commits' himself to p. That is, he cannot testify later on that he does not know about p. ($G\neg DO(\alpha)$ stands for "always from now on holds that not α will be done next" (see also definition 4 of the appendix)). Therefore, the testifying counts as a kind of evidence that cannot be denied later on. The second axiom states that r_1 also has to *know* that p is true before it can testify about it. The *testify* speech act is therefore more than just stating that something is true. This can be done at any time, but does not have such far reaching consequences as the testify. We use the testify to check whether actions have actually been performed and sometimes one of the participants must also be aware of that fact before he will perform the counter-action.

For the full technical details of dynamic deontic logic the reader is referred to the Appendix. In the following section we give a formal description of the five general auditing principles to give an idea how the logic sketched above can be used to formalize the auditing of trade procedures.

Formal Description of the General Audit Rules

Before we represent the principles in our logic we need to define a number of special predicates and actions that are useful to describe certain aspects of the audit rules:

> *Primary_activity*(r_1, α) : α is a primary activity of r_1
> *Counter_activity*(α, β) : in return for performing action α it is assumed that action β is performed (e.g. delivering a product has as counter activity paying).
> *Trust*(r_1,r_2,α) : r_1 trusts r_2 with respect to the performance of α

We also have the following special actions:

> *receive_evidence*(r_1,p) : r_1 receives some form of proof that p holds
> *witness*(r_1, α) : r_1 actually observes that α is performed
> *outsource*(r_1,r_2,α) : r_1 does not perform α herself, but delegates the execution to r_2

The following properties hold for the above relations:

$$\text{Counter\_activity}(\alpha(r_1), \beta(r_2)) \rightarrow ([\alpha(r_1)]O_{r2r1}(\beta(r_2)) \wedge [\beta(r_2)]O_{r1r2}(\alpha(r_1)))$$
$$[\text{receive\_evidence}(r_1,p)]K_{r1}(p)$$
$$[\text{witness}(r_1, \alpha)]\ K_{r1}(\text{DONE}(\alpha))$$

The first property states that if two actions are counter activities, then if one of them is performed the other one should also be performed. The second property says that a 'testify' from another agent is one way in which an agent can receive evidence. There are of course many differences, but they are not relevant for the present paper. The last property states that if r_1 witnesses α, then it is also performed. (One can only witness actions that are performed!) The additional property that is specified here is that r_1 also knows now that the action α is performed. Using the above special predicates and actions we can now formally specify the five general audit principles.

General Principle I

$Primary\_activity(r_1,\alpha) \rightarrow [\alpha(r_1)]O(testify(r_2,r_1,DONE(\alpha(r_1))))$
$Primary\_activity(r_1,\alpha) \wedge\neg\ Trust(r_1,r_2,\alpha) \rightarrow O(testify(r_2,r_1,DONE(\alpha(r_1)))\&\alpha(r_1))$

This expresses that if α is a primary activity of r_1, then after α has been performed by r_1, r_2 should testify that the activity has actually been performed. If r_1 does not trust r_2 then the testifying and performance of the action α should be done in parallel.

General Principle II

$Counter\_activity(\alpha(r_1),\beta(r_2)) \wedge \neg Trust(r_1,r_2, \beta) \rightarrow$
$O((receive\_evidence(r_1,DONE(\beta(r_2))) \cup witness(r_1, \beta(r_2)))< DO(\alpha(r_1)))$

If r_1 does not trust r_2 and α and β are counter activities then either r_1 should witness β or get evidence that it has been done before it starts to perform α. The "$\cup$" stands for a kind of logical "or" operator on actions instead of formulas. The "<" is a precedence relation. It states that an action is performed before a condition holds. (see appendix, definition 7, for a formal description). It does not mean that the action *cannot* be performed before receiving evidence or witnessing the performance of β. I.e. it is possible for the trade procedures to violate the general principles. However, it is possible within this logic framework to establish that an obligation has been violated and define special actions on the basis of such a violation.

General Principle III

$Counter\_activity(\beta(r_2), \alpha(r_1)) \wedge \neg Trust(r_1,r_2,\beta) \wedge \neg DO(witness(r_1, \beta(r_2))) \rightarrow$
$[\ O(testify(r_3,r_1, \beta(r_2))) \wedge Trust(r_1,r_3,testify(r_3,r_1, \beta(r_2))) \wedge$
$O(testify(r_3,r_1, \beta(r_2)) < DO(\alpha(r_1))))]$

Note that we use ‘$testify(r_3,r_1, \beta(r_2))$’ here in two ways. First, we state that some r_3 should testify to r_1 that the action β has been performed by r_2 if r_1 could not witness it and he does not trust r_2. Secondly, we state that this is a precondition for r_1 to perform its own activity α.

General Principle IV

$[Counter\_activity(\alpha(r_1),\beta(r_2)) \wedge \neg Trust(r_1,r_2,\beta) \wedge outsource(r_1,r_4,\alpha(r_1)) \wedge$
$\neg DONE(receive\_evidence(r_1,DONE(\beta(r_2)))) \wedge \neg DONE(witness(r_1, \beta(r_2)))\ ]$
$\rightarrow$
$O([receive\_evidence(r_4,DONE(\beta(r_2))) \cup witness(r_4, \beta(r_2))] < DO(\alpha(r_4)))$

Intuitively, this states that if r_1 outsources an action α to r_4, and r_1 could not check principle II yet, then the responsibility to check principle II is also delegated to r_4 (even though r_4 might trust r_2!)

General Principle V

$\mathrm{Primary\_activity}(r_1,\alpha) \wedge \mathrm{Counter\_activity}(\alpha(r_1),\mathrm{enable}(\beta,r_3)(r_2)) \wedge \neg\mathrm{Trust}(r_1,r_2,\beta) \wedge \neg\mathrm{DONE}(\mathrm{witness}(r_1, \beta))$
$\rightarrow$
$O(\mathrm{promise}(r_3,r_1,\beta(r_3)) < \mathrm{DO}(\alpha(r_1))) \wedge O(\mathrm{Trust}(r_1,r_3,\beta))$

This expresses that if r_1 does not trust r_2, and r_1 did not witness the performance of β, which is a counter activity of the primary activity α of r_1, and r_2 does not perform β herself but only enables r_3 to perform β, then r_1 should not perform α before it gets a promise of r_3 that she will perform β, and besides that r_1 should also trust r_3. We use 'enable(β,r_3)(r_2)' to indicate that r_2 performs the enabling actions for r_3 to perform β.

Analysis of the General Auditing Rules

In the previous section we have shown how the general auditing rules can be represented in our logic. For many people, however, this is mainly a translation from a readable format into an unreadable format. So what are the advantages of doing this? The main advantage is that it becomes possible to analyse the rules and try to find some underlying principles of these rules and show that they indeed are implemented by the rules described above. We show the potential of this analysis by sketching the underlying principles of the rules I and II, and showing how these underlying principles are implemented by the rules.

The first rule states that if a party performs a primary activity, the other party should testify that he has actually performed the activity. What is the actual reason for this rule? In fact, the party that performs the primary activity does not want that the other party can later on testify that he did not know at all that he had performed the activity (and thus refusing to pay after a delivery). The testifying is used as a safeguard against a later denial. This can be described in our logic as follows. Ideally, r_1 wants the following to hold:

$$[\alpha(r_1)]\ \neg\Diamond \mathrm{DO}(\mathrm{testify}(r_2,r, \neg K_{r2}\ (\mathrm{DONE}(\alpha(r_1)))))),$$

i.e. after the activity $\alpha(r_1)$ has been performed it is not possible in the future that r_2 testifies that he does not know that it has been performed. (The symbol $\Diamond\varphi$ represents that φ is possible.) One way of achieving this is by letting r_2 testify that the activity has been performed. From the axiom of testifying it follows immediately that:

$$[\mathrm{testify}(r_2,r_1, \mathrm{DONE}(\alpha(r_1)))]\ \neg\Diamond \mathrm{DO}(\mathrm{testify}(r_2,r, \neg K_{r2}\ (\mathrm{DONE}(\alpha(r_1)))))$$

Now there are two cases to consider. If r_1 trusts r_2 he will perform the activity before getting the testification from r_2. In that case r_1 cannot force r_2 to actually testify, but trusts that r_2 will do it. This case is represented in the first clause of general principle I. In the other case r_1 does not trust r_2. In that case r_1 does not want to perform the activity before he is sure that it will be testified by r_2 that he actually performed the activity. One way of ensuring this is by having the activity and testifying be performed at the same time. This is described in the second clause of general principle I.

Principle II states that r_1 should receive evidence or witness that r_2 performs its counter-activity before performing its own activity. The underlying principle is that r_1 does not want to perform an activity without knowing whether the counter-activity will ever be performed by r_2. E.g. delivering a product while not knowing whether it will ever be paid for. In our logic this is described as follows:

$$\text{Counter\_activity}(\alpha(r_1),\beta(r_2)) \rightarrow O(K_{r1}(\text{DONE}(\beta(r_2))) < \text{DO}(\alpha(r_1)))$$

If r_1 trusts r_2 with respect to β then r_1 knows that β is done already. (One might argue that it is more appropriate to use 'believe' in this case, but we use 'know' here to keep things as simple as possible). I.e.

$$\text{Trust}(r_1,r_2, \beta) \rightarrow K_{r1}(\text{DONE}(\beta(r_2)))$$

And thus

$$\text{Counter\_activity}(\alpha(r_1),\beta(r_2)) \wedge \text{Trust}(r_1,r_2, \beta) \wedge \text{DO}(\alpha(r_1))$$

does not create a violation for the above obligation. However, when there is no trust something else should be done to create the knowledge in r_1 that the activity β has been performed. We have identified two actions that could be used to establish this fact (but others could be used as well, as long as they have the same effect), the 'witness' and 'receive_evidence' actions. So, it follows that general principle II is actually an implementation of the above deeper principle which states that r_1 should have knowledge about activity β before he performs its counter-activity α.

The above analysis shows that the first two principles are in fact based upon more fundamental principles that are about 'evidence' and 'knowledge' of the parties involved. Using modal logic assists us in finding these more fundamental principles and subsequently shows that the principles defined earlier are indeed implementations of these underlying principles. Due to space limitations we refrain from an analysis of the other principles, but they can be analysed in the same way as the first two principles.

Modelling Trade Procedures

The same logic that is used to represent the auditing principles can also be used to represent the trade procedures. A trade procedure typically consists of a fixed sequence of actions that have to be performed by the different agent roles that are involved in the trade transaction. Examples of such actions are sending a purchase order, sending goods from seller to buyer, sending payment from buyer to seller. Clearly, the fixed sequence of actions is important here. To illustrate how such a sequence of actions is represented in our formalism, consider the following example. Consider a procedure in which (1) when an order is send by the buyer to the seller he should ship the goods to the buyer, and (2) when the goods are shipped the buyer should arrange payment of the seller. These two steps in the procedure can be modelled in our logic as follows.

1) $[\text{send}(r_1,r_2,\text{order})]O_{r2r1}(\text{ship}(r_2,r_1,\text{goods}))$

2) $[\text{ship}(r_2,r_1,\text{goods})]O_{r1r2}(\text{arrange}(r_1,r_2,\text{payment}))$

These formal representations of trade transaction actions can be combined into a formal representation of the complete procedure. Automated auditing is done by checking whether the formal representation of the procedure complies with the formal representation of the audit principles. We say that a trade procedure complies with the audit rules, if after any series of events that the trade procedure applies to we *cannot* reach a state which satisfies the antecedent of an audit rule and at the same time does not violate the obligations from the trade procedure but *does* violate the consequence of the audit rule. For example, assume that arranging payment is the counter-activity of shipping goods (i.e. $Counter\_activity(arrange(r_1,r_2,payment),ship(r_2,r_1,goods))$) and that there is no trust between agent r_1 and r_2, with respect to the shipment action (i.e. $\neg Trust(r_1,r_2, ship(r_2,r_1,goods))$), then from these two assumptions and the two formulas that describe the trade procedure we can get into a state where holds that:

$$\neg DONE(ship(r_2,r_1,goods)) \wedge$$
$$DO(arrange(r_1,r_2,payment)) \wedge$$
$$\neg DONE(receive\_evidence(r_1,DONE(ship(r_2,r_1,goods)))) \wedge$$
$$\neg DONE(witness(r_1, ship(r_2,r_1,goods)))$$

which do not violate any of the obligations in the trade procedure but which does violate the obligation:

$$O((receive\_evidence(r_1,DONE(ship(r_2,r_1,goods))) \cup witness(r_1, ship(r_2,r_1,goods))) < DO(arrange(r_1,r_2,payment))) .$$

Hence, this procedure does not comply with the second auditing rule, because there is no requirement to either receive evidence about or witness shipment before payment is done. At face value the procedure seems correct, but if we take a closer look at the procedure, then it is obvious that it is not trustworthy for the buyer in case that the shipment takes place at the other end of the world. Suppose the buyer is in Hong Kong and the seller in Rotterdam. According to the procedure the buyer should pay at the moment the seller has shipped the goods on board of a ship in the port of Rotterdam. But the buyer will never trust such a procedure if he has absolutely no evidence that the seller indeed shipped the goods to him. One way to make this procedure consistent with the second auditing rule is to introduce an extra step in which first the seller has to send evidence to the buyer that he indeed shipped the goods. (Sufficient evidence here could be that the captain of the vessel that carries the goods issues a Bill of Lading document that he received the goods.) This would be the following procedure.

$$[send(r_1,r_2,order)]O_{r2r1}(ship(r_2,r_1,goods)) \wedge O_{r2r1}(testify(r_2,r_1,(ship(r_2,r_1,goods))))$$

$$[receive\_evidence(r_1,DONE(ship(r_2,r_1,goods))]O_{r1r2}(arrange(r_1,r_2,payment))$$

and this trade procedure does comply with the second auditing principle. I.e. there is no state that does not violate the obligations from the procedure but does violate the following obligation:

$$O([receive\_evidence(r_1,DONE(ship(r_2,r_1,goods))) \cup witness(r_1, ship(r_2,r_1,goods))] < DO(arrange(r_1,r_2,payment))).$$

Hence, it does comply with the second auditing principle. Of course, this example is only a simplified version of a small part of the real trade procedure. However, it shows how the modeling and checking could be done.

Conclusions

In this paper we have argued that for open electronic commerce applications there is usually a very low level of trust between the trading partners. Therefore it is necessary to create this trust through the use of trustworthy trade procedures. These procedures guarantee a certain level of trust even though the partners do not trust each other. We have given five general audit rules with which the trade procedures should comply in order to be trustworthy.

We have argued that it is beneficial to have a formal representation of these procedures in order to verify the rules. We have given an indication (by example) that both the audit rules as well as the procedures and contracts can be modelled using the same (rich) modal logic. This makes it possible to formally check the procedures against the audit rules at an abstract level. It also enables us to verify the implementation of the general principles in INTERPROCS. A formal analysis can often give a new insight in a theory by making underlying assumptions explicit that are hidden behind the fundamental vocabulary of the theory. In particular, we argued in this paper that there are fundamental axioms about the way people communicate with each other that might explain the general audit principles. This explanation is based on an investigation of the formal properties and axioms that hold for the basic vocabulary of modal operators that we used in this paper to represent these general audit principles.

The modal logic that is presented in this paper is quite expressive, but still need to be expanded with some extra concepts to model all the subtle points that arise in open electronic commerce. One thing that has not been addressed yet is the possibility of fraud. We have assumed that after one party testifies that an action has been performed the other party knows it has been performed. Actually, this should be that the other party *believes* it has been performed. So, we actually should extend the logic with a belief operator to capture this difference. We leave this and other extensions for future work.

In this paper we have actually shown two separate things using the modal logic that we developed. First it is possible to analyze the general audit rules using this logic to find some more fundamental principles. This analysis shows that the audit principles themselves are a type of implementation of more general prinicples that underlay trade relations. Although this analysis requires quite some background in (modal) logic it is possible to perform it on all audit principles and thus recover all underlying principles, check on consistency and also on some form of completeness.

However, a different point is the application of the logic to perform the actual auditing of the trade procedures. The example in the previous section showed some interesting results. However, it does not seem to be practical to check complete procedures in this way. In order to make this more than a theoretical exercise one would need a tool (like INTERPROCS) to automate this analysis based on the logic given. One way would be to actually have a kind of multi-modal theorem prover that could derive the required results. However, this is not practically feasible with the current state of the art tools available. Another way to proceed would be to start from INTERPROCS and show that the programs used in INTERPROCS comply completely (and very rigourously) to the logical principles of the multi-modal logic. It means in fact that we should verify and probably extend INTERPROCS on the basis of the logical theory. This is not an easy task, but once completed would give us a very powerful and still usable tool to audit the tradeprocedures.

Acknowledgements

The research on designing trustworthy trade procedures for open electronic commerce has been conducted in close co-operation with René Wagenaar.

Appendix

Syntax and Semantics of Dynamic Deontic Logic

We give a more formal description of the syntax and semantics of the combined dynamic deontic logic. We add one new operator to the language defined in [22]. The formulas defined here define the standard deontic logic as introduced by Von Wright [29]. Let *Prop* be a set of atomic propositions and *Act* a set of action expressions (see below).

Definition 1.

The set *Form* of formulas is the smallest set closed under the following production rules:

1. $Prop \subseteq Form$
2. $\varphi_1, \varphi_2 \in Form \rightarrow \varphi_1 \wedge \varphi_2 \in Form$
3. $\varphi \in Form \rightarrow \neg\varphi \in Form$
4. $\alpha \in Act, \varphi \in Form \rightarrow [\alpha]\, \varphi \in Form$
5. $\varphi \in Form \rightarrow O(\varphi) \in Form$

Other propositional connectives such as $\vee$ and $\rightarrow$ are assumed to be introduced as the usual abbreviations. Also the special proposition *false* is introduced as the abbreviation of $p \wedge \neg p$ for some $p \in Prop$. The semantics of the formulas in *Form* is given in two stages. First we give the syntax and semantics of the action expressions, which we subsequently use to define the semantics of the formulas.

Action Expressions and their Semantics

First we give a definition of *action expressions*, which we shall typically denote by the symbol α, possibly with subscripts. Let *Act* be the set of atomic action expressions that are denoted by the letters *a,b,...*

Definition 2.

The set *Act* of action expressions is given by the following BNF:

$$\alpha ::- a \mid \textbf{any} \mid \textbf{fail} \mid \alpha_1 \cup \alpha_2 \mid \alpha_1 \,\&\, \alpha_2 \mid \overline{\alpha}$$

The semantics of action expressions has two components. The first component is an algebra of uninterpreted actions of a so-called uniform semantics, which allows us to interpret equalities between action expressions without taking their effect into account. Due to lack of space we refer to [9] for a formal account of this part. The following intuitions should suffice here. In the algebraic semantics, each action expression is interpreted as a choice over possible steps, where a step is a set of events that can be performed simultaneously. We write $\lfloor\alpha\rfloor$ to denote the set of steps that form the uniform semantics of the action α. For example, the expression *a & b* is interpreted as the set of steps, where each step contains at least {a, b}, whereas $a \cup b$ is interpreted as the choice from possible steps containing at least {a} or containing at least {b}. The 'at least' constraint concerns a technical issue related to open versus closed systems which is elaborated on in [9] and not further explained here.

The second component of the semantics is a state-transition semantics of action expressions where we define the effect of steps on the state of the world. Most notable here is that the effect of any action can always be a set of new states (one for each step in its semantics).

Semantics of Formulas

Having defined the semantics of the action expressions within the formulas, we can now give the semantics of formulas in *Form* by means of the notion of a Kripke structure.

$M = (A, \Sigma, \pi, R_A, R_O)$.
Σ is a set of states (worlds).
A is a finite set of events.
π is a truth assignment function to the atomic propositions relative to a state; i.e.
π is a function $\Sigma \rightarrow (Prop \rightarrow \{tt, ff\})$, where tt and ff denote truth and falsehood.

Thus, for $p \in Prop$, $\pi(\sigma)(p) = tt$ means that the atomic proposition p is true in state σ. The accessibility relation R_A specifies how actions can change states. The relation R_A is defined as follows: $R_A = \{ R_t \mid t \in A \}$, reflecting that R_t is the relevant entity. We write, slightly abusing notation, $R_A(\sigma, \sigma') \leftrightarrow \exists t : R_t(\sigma, \sigma')$

The relation R_O relates states σ to states σ' that are deontically ideal with respect to σ. Intuitively, in the states σ' all the obligations are fulfilled and no prohibitions have been violated. We assume the relation R_O to be serial; i.e. for every world σ there exists at least one world σ' such that $R_O(\sigma, \sigma')$ holds. We give the interpretation of formulas in *Form* in Kripke structures. We interpret formulas with respect to a structure M and a state $\sigma \in \Sigma$.

Definition 3.
Given $M = (A, \Sigma, \pi, R_A, R_O)$ as above and $\sigma \in \Sigma$, we define:

1. $(M,\sigma) \models p \Leftrightarrow \pi(\sigma)(p) = tt$ (for $p \in Prop$)
2. $(M,\sigma) \models \varphi_1 \wedge \varphi_2 \Leftrightarrow (M,\sigma) \models \varphi_1$ and $(M,\sigma) \models \varphi_2$
3. $(M,\sigma) \models \neg\varphi \Leftrightarrow$ not $(M,\sigma) \models \varphi$
4. $(M,\sigma) \models [\alpha]\varphi \Leftrightarrow \forall t \in \lfloor\alpha\rfloor\ \forall \sigma' \in \Sigma [\, R_t(\sigma,\sigma') \Rightarrow (M,\sigma') \models \varphi]$
5. $(M,\sigma) \models O(\varphi) \Leftrightarrow \forall \sigma' \in \Sigma\ [R_O(\sigma,\sigma') \Rightarrow (M,\sigma') \models \varphi\,]$

The formula φ is *valid* w.r.t. a model $M = (A, \Sigma, \pi, R_A, R_O)$, denoted by $M \models \varphi$, if $(M,\sigma) \models \varphi$ for all $\sigma' \in \Sigma$. φ is *valid*, denoted by $\models \varphi$, if φ is valid with respect to all models M of the form considered above.

The definitions are quite standard and we do not explain them in detail here. The following axioms can be proven valid on the dynamic logic formulas:

$$[\alpha](p \rightarrow q) \rightarrow ([\alpha]p \rightarrow [\alpha]q)$$
$$[\alpha_1 \cup \alpha_2]p \leftrightarrow [\alpha_1]p \wedge [\alpha_2]p$$
$$[\alpha_1 \& \alpha_2]p \leftarrow [\alpha_1]p \wedge [\alpha_2]p$$
$$[\alpha_1;\alpha_2]p \leftrightarrow [\alpha_1]\,[\alpha_2]p$$
$$[\mathbf{any}]\mathbf{true}$$

The last axiom says that it is always possible to perform some action (and arrive in a next state). For the deontic operator at least the following axioms hold:

$$(O(p) \wedge O(p \rightarrow q)) \rightarrow O(q)$$
$$O(p \wedge q) \leftrightarrow O(p) \wedge O(q)$$
$$O(p \vee q) \leftarrow O(p) \vee O(q)$$

In deontic logic one can also express the prohibition and permission in terms of the obligation:

$$F(p) \equiv O(\neg p)$$
$$P(p) \equiv \neg O(\neg p)$$

Because we do not need the prohibition and permission for the specification of the rules and trade procedures in this paper we do not elaborate on this issue. Using a combination of deontic and dynamic logic we can now also describe (directed) obligations on actions:

$$O_{ij}(\alpha) \equiv [\mathbf{any}]O_{ij}(DONE(\alpha))$$

In other words, agent i is obliged towards j that α is performed is defined as the fact that whatever happens next, afterwards the action α should have been performed as the last one (possibly together with many other actions). Note that we do not require that i performs the action herself! For the obligation on actions similar axioms hold as for the obligation on conditions. So we have:

$$O(\alpha_1 \;\&\; \alpha_2) \leftrightarrow O(\alpha_1) \wedge O(\alpha_2)$$
$$O(\alpha_1 \cup \alpha_2) \leftarrow O(\alpha_1) \vee O(\alpha_2)$$

Because there is no notion of implication between actions, we also have no axiom for that case. Note that the definition also does not allow obligations on sequences of actions. (Due to the fact that the DONE operator is only defined for actions. Extending the definition for the DONE operator for arbitrary transactions complicates the theory considerably, as is shown in [6]). It is possible to give an extra definition for the obligation of transactions in terms of actions as follows:

$$O(\alpha_1;\alpha_2) \equiv O(\alpha_1) \wedge [\alpha_1]O(\alpha_2)$$

The above definitions of the obligation imply that when an action is obligated in a certain state, then it should be performed immediately afterwards. Since obligations usually do not have to be fulfilled instantaneously, we introduce temporal operators into this framework that allow us to specify that an action should be performed before a certain condition holds.

Syntax and Semantics of Dynamic Deontic Linear Time Temporal Logic (DDLTLB)

The monadic Next operator *X*, and the binary until operator *Until* are used as temporal future tense operators. The meaning of *X* is the obvious one, the Until states that some property is preserved until the first time that some other property is fulfilled. In the literature also an *Unless* operator is defined in terms of the Until operator, which basically means the same but that does not require that this fulfilment needs ever to occur. However, we do not need this operator in this paper. For the past time we introduce the mirror operators *P* and *Since*, indicating that a certain formula held in the previous state and that some property held since the first time some other property was fulfilled. We also formally introduce the operators DO and DONE over actions, indicating that a certain action will be performed next or was performed last.

Syntax of DDLTLB

We do not define PLTLB separately, but immediately define DDLTLB as an extension of the dynamic deontic logic defined above. Thus, Definition 2 is extended with the following six production rules.

Definition 4. (extension of Defintion 1.)

6. $\varphi \in \mathit{Form} \rightarrow X\varphi \in \mathit{Form}$
7. $\varphi \in \mathit{Form} \rightarrow P\varphi \in \mathit{Form}$
8. $\varphi_1, \varphi_2 \in \mathit{Form} \rightarrow \varphi_1\ \mathit{Until}\ \varphi_2 \in \mathit{Form}$
9. $\varphi_1, \varphi_2 \in \mathit{Form} \rightarrow \varphi_1\ \mathit{Since}\ \varphi_2 \in \mathit{Form}$
10. $\alpha \in \mathit{Act} \rightarrow \mathrm{DO}(\alpha)\ \varphi \in \mathit{Form}$
11. $\alpha \in \mathit{Act} \rightarrow \mathrm{DONE}(\alpha)\ \varphi \in \mathit{Form}$

The derived operators ◊(eventually) and G(always from now) are defined as follows:

$$\Diamond\varphi \equiv \text{true}\ \mathit{Until}\ \varphi$$
$$G\varphi \equiv \neg\Diamond\neg\varphi$$

Other operators like ‘sometime in the past’ and ‘always in the past’ can be defined similarly based on the *Since* operator. Because we do not use these operators in this article we do not discuss further techincal details here.

Semantics of DDLTLB

The first question is, how to interpret the temporal operators on the Kripke models that contain transitions from R_A and R_O. Traditionally, linear time temporal logic formulas are interpreted over state sequences. For systems with a Kripke structure semantics a formula is then defined to hold for the system if it holds for all the paths through its Kripke structure. Because of the presence of transitions from R_O, and possible mixtures of deontic and temporal operators, we have to sequentialise the Kripke structure into a set of paths after each occurrence of either a dynamic logic operator or the deontic O operator. This situation is similar, but not identical, to the situation in branching time temporal logic (cf. [11]). We interpret every formula in a triple (M,ρ,σ). In this triple the state σ can be seen as an index of the path ρ, making the interpretation similar to the one for the logic PLTLB in [11].

As said before, the relation R_O does not depend on the temporal paths. However, it seems natural that in an ideal state with respect to the present state, also the ideal temporal path is different from the present temporal path. Therefore, in the Kripke structure the definition of the R_O relation is reinterpreted to be a relation between *pairs* of state/path (σ,ρ) to relatively deontic ideal state/path pairs (σ',ρ'). We introduce some notation on paths to facilitate the definitions. Let $\rho = <\ldots\ \sigma_{-1}, t_{-1}, \sigma_0, t_0, \sigma_1, t_1 \ldots>$ be an indexed state/transition path through M, then the set of all such paths ρ is called *Path*(M). The state σ_i on the path ρ is denoted by $\rho(i)$. We use $\rho^-(\sigma_i)$ for the prefix of ρ that ends at σ_i and $\rho^+(\sigma_i)$ to denote the suffix of ρ that starts with σ_i.

Definition 5.
$Path(M,\sigma) = \{\rho \mid \rho \in Path(M) \text{ and } \exists i: \sigma=\rho(i)\}$.
$Path^+(M,\sigma) = \{\rho \mid \rho \in Path(M) \text{ and starts with } \sigma\}$.
$Path^-(M,\sigma) = \{\rho \mid \rho \in Path(M) \text{ and ends with } \sigma\}$.

The following definition defines the concatenation of (sets of) paths.

Definition 6.
Let $\rho = <\ldots\ \sigma_{-1}, t_{-1}, \sigma_0>$ and $\rho' = <\sigma_0, t_0, \sigma_1, t_1 \ldots>$ then $\rho\circ\rho' \equiv <\ldots\ \sigma_{-1}, t_{-1}, \sigma_0, t_0, \sigma_1, t_1 \ldots>$
Let $\rho = <\ldots\ \sigma_{-1}, t_{-1}, \sigma_0>$ and $\Omega \subseteq Path^+(M,\sigma_0)$ then $\rho\circ\Omega = \{\rho' \mid \rho'^-(\sigma_0)=\rho \text{ and } \rho'^+ \in \Omega\ \}$
Let $\rho = <\sigma_0, t_0, \sigma_1, t_1 \ldots>$ and $\Omega \subseteq Path^-(M,\sigma_0)$ then $\Omega\circ\rho = \{\rho' \mid \rho'^+(\sigma_0)=\rho \text{ and } \rho'^- \in \Omega\ \}$
Let $\Omega \subseteq Path^-(M,\sigma_0)$ and $\Omega' \subseteq Path^+(M,\sigma_0)$ then $\Omega\circ\Omega' = \{\rho' \mid \rho'^- \in \Omega \text{ and } \rho'^+ \in \Omega'\}$

The semantics of formulas in DDLTLB is given as follows.

Definition 7.
Given $M = (A, \Sigma, \pi, R_A, R_O)$ as above and $\rho \in Path(M,\sigma)$, we define:

1. $(M,\rho,\sigma) \models p \Leftrightarrow \pi(\sigma)(p) = tt$ (for $p \in Prop$)
2. $(M,\rho,\sigma) \models \varphi_1 \wedge \varphi_2 \Leftrightarrow (M,\rho,\sigma) \models \varphi_1$ and $(M,\rho,\sigma) \models \varphi_2$
3. $(M,\rho,\sigma) \models \neg\varphi \Leftrightarrow$ not $(M,\rho,\sigma) \models \varphi$
4. $(M,\sigma) \models [\alpha]\varphi \Leftrightarrow$
   $\forall t \in \lfloor\alpha\rfloor\ \forall\sigma' \in \Sigma\, \forall\rho' \in \rho^-(\sigma)\circ<\sigma,t,\sigma'>\circ Path^+(M,\sigma')\ [\, R_t(\sigma,\sigma') \Rightarrow (M,\rho',\sigma') \models \varphi]$

i.e. basically it means that if α is performed then φ holds true afterwards. The paths that can be considered in the new state are fixed to the past (equal to ρ followed by t) and open to the future.

5. $(M,\rho,\sigma) \models O(\varphi) \Leftrightarrow \forall\sigma'\in\Sigma\forall\rho'\in Path(M,\sigma')\ [R_O((\sigma,\rho),(\sigma',\rho')) \Rightarrow (M,\rho',\sigma') \models \varphi]$
   i.e. φ is obligated if in all states and paths that are ideal with respect to the present state and path φ is true.
6. $(M,\rho,\sigma_i) \models X(\varphi) \Leftrightarrow (M,\rho,\sigma_{i+1}) \models \varphi$
7. $(M,\rho,\sigma_i) \models P(\varphi) \Leftrightarrow (M,\rho,\sigma_{i-1}) \models \varphi$
8. $(M,\rho,\sigma_0) \models \varphi\mathit{Until}\psi \Leftrightarrow \exists k(\ (M,\rho,\sigma_k) \models \psi$ and $\forall j,\ 0\leq j<k\ (M,\rho,\sigma_j) \models \varphi$
   i.e. φ holds along the path ρ until ψ becomes true.
9. $(M,\rho,\sigma_0) \models \varphi\mathit{Since}\psi \Leftrightarrow \exists k(\ (M,\rho,\sigma_k) \models \psi$ and $\forall j,\ k<j\leq 0\ (M,\rho,\sigma_j) \models \varphi$
   i.e. φ holds along the path ρ since ψ became true.
10. $(M,\rho,\sigma_0) \models DO(\alpha) \Leftrightarrow \rho = <\sigma_0, t_0, \sigma_1, t_1 \ldots>$ and $t_0\in\lfloor\alpha\rfloor$
    i.e. α is the action performed next.
11. $(M,\rho,\sigma_0) \models DONE(\alpha) \Leftrightarrow \rho = <\ldots\sigma_{-1}, t_{-1}, \sigma_0 \ldots>$ and $t_{-1}\in\lfloor\alpha\rfloor$
    i.e. α is the last action performed.
12. φ is *atemporal* w.r.t. model $M = (A, \Sigma, \pi, R_A, R_O)$, notation $(M,\sigma) \models \varphi$ if $(M, \rho,\sigma) \models \varphi$ for all $\rho\in Path(M,\sigma)$.
13. φ is *valid* w.r.t. model $M = (A, \Sigma, \pi, R_A, R_O)$, notation $M \models \varphi$, if $(M, \rho,\sigma) \models \varphi$ for all $\sigma\in\Sigma$ and $\rho\in Path(M,\sigma)$.
14. φ is *valid*, notation $\models \varphi$, if φ is valid w.r.t. all models M of the form considered above.

The addition of the *Until* operator makes it possible to specify that an action α is performed before a certain condition p holds:

$$\alpha<p \equiv \neg p\mathit{Until}DONE(\alpha)$$

I.e. p does not hold before α is performed, which implies that α is performed before p holds. This formula can be used straight away to formulate the obligation that an action α is performed before p holds:

$$O_{ij}(\alpha<p) \equiv O_{ij}(\neg p\mathit{Until}DONE(\alpha))$$

In [7] we have shown how different types of deadlines can be modelled using the temporal operators defined above.

Epistemic Dynamic Deontic Temporal Logic

In this section we incorporate the speech acts and knowledge operator in the logic. The syntax of the formulas changes very little by incorporating the knowledge operator $K_i(\varphi)$, which means that agent i knows φ. We use here the standard modal operator from epistemic logic, which has the following axioms[5]:

$$(K_i(p) \wedge K_i(p \rightarrow q)) \rightarrow K_i(q)$$
$$K_i(p) \rightarrow p$$
$$K_i(p) \rightarrow K_i(K_i(p))$$
$$\neg K_i(p) \rightarrow K_i(\neg K_i(p))$$

The second one is of special interest here. It states that an agent i can only know p, if p is true. This is the main difference between knowledge and belief. You can belief something, which is not true. We simply add the following production rule to Definition 1 and 4:

Definition 8. (extension of Definitions 1 and 4)
12. $\varphi \in \mathit{Form}$, $i \in \mathit{Ag} \rightarrow K_i(\varphi) \in \mathit{Form}$

where *Ag* is a (finite) set of indices indicating the agents in the system. In order to incorporate the speech acts in the logic we have to extend the definition of action expressions, because speech acts are actions that contain other actions and formulas as parameter. The extension can be made by changing the definition for action expressions as follows. Let the set *Act* of action expressions be defined as in Definition 2.

Definition 9.
The set of basic speech acts is given by the following rule:

$$\alpha \in \mathit{Act},\ \varphi \in \mathit{Form},\ i,j \in \mathit{Ag} \rightarrow \text{promise}(i,j,\alpha),\ \text{direct}(i,j,\alpha),\ \text{testify}(i,j,\varphi) \in \mathit{BSact}$$

The set of speech acts *SAct* is now given by the following rules:

1. $\alpha \in \mathit{BSAct} \rightarrow \alpha \in \mathit{SAct}$
2. $\alpha_1 \in \mathit{SAct},\ \alpha_2 \in \mathit{SAct} \rightarrow \alpha_1 \cup \alpha_2$, α_1 & α_2, $\overline{\alpha}_1 \in \mathit{SAct}$
3. $\alpha \in \mathit{SAct} \rightarrow \text{promise}(i,j,\alpha),\ \text{direct}(i,j,\alpha),\ \text{testify}(i,j,\varphi) \in \mathit{SAct}$

Note that this is a recursive definition. So, we can have speech acts about speech acts, etc.
The set of all actions *GAct* is now given by $\mathit{GAct} = \mathit{SAct} \cup \mathit{Act}$
The above set of speech acts is not the universal one introduced in [23]. Searle distinguishes five basic types: directives, commissives, assertives, declaratives and expressives. We have, in this paper, chosen a subset that was sufficient for developing the theory, while trying to stay close to the terminology used in [3]. The promise is a (prototypical) element of a commissive, the direct is a directive and the testify is a special kind of assertive. For a more general and complete description one can consult [8, 10].
The following axioms hold for speech acts:

$$\text{promise}(i,j,\ \alpha_1 \& \alpha_2) \leftrightarrow \text{promise}(i,j,\ \alpha_1) \ \&\ \text{promise}(i,j,\ \alpha_2)$$
$$\text{direct}(i,j,\ \alpha_1 \& \alpha_2) \leftrightarrow \text{direct}(i,j,\ \alpha_1) \ \&\ \text{direct}(i,j,\ \alpha_2)$$

Note that these equivalences are a direct consequence of the fact that the resulting obligations hold immediately after the speech act and have to be fulfilled immediately.

$$\text{testify}(i,j,\varphi) \cup \text{testify}(i,j,\psi) \leftrightarrow \text{testify}(i,j,\varphi \wedge \psi)$$

The speech acts of *SAct* as defined above do not contain all elements of speech acts that are identified in illocutionary logic as introduced in [10]. The propositional content conditions are not modelled here. They can be modelled through a refinement of the language *SAct* which renders only those speech acts syntactically correct that comply with the propositional content conditions. In an Electronic Trading environment, the propositional content conditions are contained in the data model of the trade procedures.

To include the semantics of the epistemic operator in the model we extend the model with one extra set of relations (relative to a set of agent indicators). We also have to include the set of agent indicators now. So the model M is given as $M = (A, \mathrm{Ag}, \Sigma, \pi, R_A, R_O, RK_{Ag})$ The set of semantic events A now also contains the semantic counterparts of the basic speech acts. These are functions that define exactly what the consequence of a speech act in a certain state is. These functions are quite complicated and are not explained further in this paper. We refer to [8] for further details on these functions. The semantics of the epistemic operator are defined through the following definition:

$$(M,\rho,\sigma) \models K_i(\varphi) \Leftrightarrow \forall\sigma'\in\Sigma\forall\rho'\in Path(M,\sigma')\ [RK_i((\sigma,\rho),(\sigma',\rho')) \Rightarrow (M,\rho',\sigma') \models \varphi]$$

This concludes the description of the epistemic dynamic deontic temporal logic that is needed to represent the audit rules and trade procedures.

Notes

[1] See, for example, the Secure Electronic Transaction (SET) protocol that is developed by VISA/Master card.
[2] Formerly called CASE/EDI.
[3] For a detailed explanation of the role of the INCOTERMS see [13]. This paper also contains a description of the expert system INCAS, that was developed at EURIDIS, for advice on the use of Incoterms in international trade contracts.
[4] For simplicity we assume here that the underlying structure of time is discrete, i.e. for any moment in time t there is an immediate predecessor time point t-1 and an immediate successor time point t+1.
[5] These axioms for knowledge are idealizations. Especially, the first axiom and fourth axiom have been much criticized (see e.g. [12]). In future research we hope to develop axiomatisations of knowledge that are more adequate for our purposes.

References

1. Åqvist, L. Deontic logic. In D.M. Gabbay and F. Guenthner (editors) *Handbook of Philosophical Logic II*, pages 605-714. Dordrecht: Reidel, 1984.

2. Austin, J.L. *How to DO things with words*, Cambridge, MA: Harvard University Press, 1962.

3. Bons, R.W.H. *Designing Trustworthy Trade Procedures for Open Electronic Commerce*, PhD thesis, Erasmus University Rotterdam, October, 1997.

4. Bons, R.W.H.; Lee, R.M. and Wagenaar, R.W. Designing trustworthy inter-organisational

trade procedures for open electronic commerce, *Proceedings of the 10th international Bled Electronic Commerce Conference*, pp. 39-70, Slovenia, 1997

5. Chen, Kuo Tay. *Schematic Evaluation of Accounting Control Systems*, Ph.D. Dissertation, University of Texas at Austin, 1992.

6. Dignum, F. Using transactions in integrity constraints. *Proceedings First International Workshop on Applied Logic: Logic at Work*, CCSOM, University of Amsterdam, 1992.

7. Dignum, F. and Kuiper, R. Combining dynamic deontic logic and temporal logic for the specification of dead lines. In: R. Sprague Jr (editor) *Proceedings of 30th Hawaii International Conference on System Sciences (HICSS'97*), Los Alamitos, CA: IEEE Computer Society, 1997.

8. Dignum, F. and Linder, B. van Modeling Social Agents: Communications as Actions. In M. Wooldridge; J. Muller and N. Jennings (editors) *Intelligent Agents III (LNAI 1193*), pages 205-218, Berlin: Springer Verlag, 1997.

9. Dignum, F.; Meyer, J.-J. and Wieringa, R. Contextual permission. a solution to the free choice paradox. In A. Jones and M. Sergot (editors) *Second International Workshop on Deontic Logic in Computer Science*, pages 107--135, Oslo: Tano A.S., 1994.

10. Dignum, F. and Weigand, H. Communication and deontic logic. In R. Wieringa and R. Feenstra (editors) *Information Systems, Correctness and Reusability*, pages 242--260. Singapore: World Scientific, 1995.

11. Emerson, E.A. Temporal and Modal Logic. In J. van Leeuwen (editor) *Handbook of Theoretical Computer Science*, pages 995-1072, Amsterdam: North-Holland, 1989.

12. Fagin, R.; Halpern, J.Y.; Moses, Y. and Vardi, M.Y. *Reasoning About Knowledge*, Cambridge, MA: MIT Press, 1995.

13. Foekens, A.; Mitrakas, A. and Tan, Y.H. Facilitating International Electronic Commerce by Formalizing the Incoterms, *Proceedings of the 30th Hawaii International Conference on System Sciences* (*HICSS'97*), Los Alamitos, CA: IEEE Computer Society Press, 1997.

14. Firozabadi, B.S.; Tan, Y.H. and Lee, R.M. Formal definitions of fraud, in: P. McNamara and H. Prakken (eds.) *Norms, Logics and Information Systems, New Studies in Deontic Logic and Computer Science*, Amsterdam: IOS Press, 1999. pp.275-287.

15. Gabbay D. *Fibring Logics*, Oxford University Press, to appear.

16. Harel, D. *First Order Dynamic Logic.* LNCS 68, Berlin: Springer Verlag, 1979.

17. Lee, Ronald M. *Auditing as Pattern Recognition*, Working Paper, Department of Management Sciences and Information Systems, University of Texas at Austin, August 1991.

18. Lee, Ronald M. *Dynamic Modeling of Documentary Procedures: A CASE for EDI*, Proceedings of Third International Working Conference on Dynamic Modeling of Information Systems, Noordwijkerhout, The Netherlands, June 1992.

19. Lee, Ronald M. and Dewitz, S.D. Facilitating International Contracting: AI Extensions to EDI, *International Information Systems*, January, 1992.

20. Lee, Ronald M and Bons, Roger W.H. Soft-Coded Trade Procedures for Open-edi, *International Journal of Electronic Commerce*, Vol 1, No 1, 1996, pp. 27-49.

21. Lee, Ronald M. INTERPROCS: A Java-based Prototyping Environment for Distributed Electronic Trade Procedures, *Proceedings of the 31th Hawaii International Conference on System Sciences* (*HICSS'98*), Los Alamitos, CA: IEEE Computer Society, 1998.

22. Meyer, J-J. Ch. A different approach to deontic logic: deontic logic viewed as a variant of dynamic logic, *Notre Dame Journal of Logic* 29(1), 1988, pp. 109-136.

23. Searle, J. *Speech Acts: An Essay in the Philosophy of Language*, London, UK: Cambridge University Press, 1969.

24. SITPRO, *Exporter's Problems with Commercial Letters of Credit*, SITPRO, London, UK, 1991.

25. Tan, Y.H. and Thoen W. Modeling directed obligations and permissions in trade contracts, *Proceedings of the 31th Hawaii International Conference on System Sciences* (*HICSS'98*), Los Alamitos, CA: IEEE Computer Society, 1998.

26. Tan, Y.H. and Thoen, W. A logical model of directed obligations and permissions to support electronic contracting, *International Journal of Electronic Commerce*, Vol. 3, Nr. 2, 1999. pp. 87-104.

27. Tan, Y.H. and Thoen, W. A generic model of trust in electronic commerce, *Proceedings of the 13th Bled Conference on Electronic Commerce*, Bled, Slovenia, 1999.

28. Wagenaar, R.W. *The Virtual Merchant; fiction or reality*, Inaugural address Free University Amsterdam, Amsterdam, Netherlands, March, 1997. (in Dutch)

29. Wright, G.H. von Deontic Logic. *Mind*, Vol.60, pages 1—15, 1951.

ROGER W.H. BONS (roger.bons@philips.com) completed his dissertation in 1997 at the Erasmus University Rotterdam (working at both the EURIDIS institute and the Rotterdam School of Management). His research primarily focuses on analysing and modelling of 'trustworthy trade procedures', studying the role of information

in implementing transactional controls in electronic commerce and the role and organization of trusted third parties. In 1998 he joined the consulting firm Multimedia Skills where he worked with some of the major European telecommunications providers on customer relationship management, supply chain management and electronic commerce. Since June 1999 he works for the Corporate IT department of Royal Philips Electronics and is responsible for organizing the corporate eCommerce IT infrastructure.

FRANK DIGNUM (dignum@win.tue.nl) got his Ph.D. in 1989 from the Free University of Amsterdam. After this he set up the Computer Science department of the University of Swaziland. After one year in the AI department of the Technical University of Lisbon he is now working at the Eindhoven University of Technology.). His main interests lay in the area of formal modelling communication and social relations. The use of several modal logics is of prime importance in this research. The results are applied in the area of electronic commerce and agent communications.

RONALD M. LEE (R.Lee@fac.fbk.eur.nl) is currently Director of the Erasmus University Research Institute for Decision Information System (Euridis) in Rotterdam, the Netherlands. Prior to that, he was Associate Professor of Information Systems at the Management Science and Information Systems Department at the University of Texas at Austin. He has a PhD in Decision Sciences (Wharton, University of Pennsylvania, 1980), and has previously served as a research scholar at the International Institute for Applied Systems Analysis in Vienna, Austria, and as Visiting Professor of Management at the Universidade Nova de Lisboa, in Lisbon, Portugal. Current research focuses on applications of artificial intelligence to business; special focus is 'logic modeling', the use of formal logic representations for management science applications; current projects involve the use of logic modeling and Petri nets to represent and manage formal business communications, specifically electronic document interchange (EDI), focusing on electronic commerce and governmental bureaucracies. Theoretical aspects include the role of deontic and illocutionary logic in representing formal business conversations. Practical value is the reduction of bureaucratic and legalistic red tape. This work also includes multi-lingual business communications, supporting structured communications between parties with different native languages.

YAO-HUA TAN (Y.Tan@fac.fbk.eur.nl) studied mathematical logic and computer science at the universities of Amsterdam and Groningen in the Netherlands, and received his Ph.D. from the department of Mathematics and Computer Science of the Free University of Amsterdam. Currently, he is associate professor at the Rotterdam School of Management and associate director of EURIDIS at the Erasmus University Rotterdam. He is also scientific director of the Erasmus Center for Electronic Commerce (ECEC) of the Erasmus University. He served as the Reynolds Visiting Professor at the Wharton School of the University of Pennsylvania. His current research focuses on formal models for electronic commerce, which can be used to develop software for a wide range of activities such as electronic contracting and automated auditing of electronic trade procedures.