COMPUTING LABORATORY

Conversation Structures as a Means of Specifying Security Policy

J.E. Dobson

TECHNICAL REPORT SERIES
No 295    October, 1989
Conversations Structures as a Means of Specifying Security Policy

J.E. Dobson

Abstract

This paper examines the concept of a security policy. It proposes speech act theory as a language for discussing the semantics of information systems and argues that a security policy must, in addition to specifying constraints on the instrumental acts of authorised access to data, also be capable of expressing the valid sequence of speech acts that constitute the authorisation and create commitment to the policy.

Series Editor: M.J. Elphick

©1989 University of Newcastle upon Tyne.
Printed and published by the University of Newcastle upon Tyne, Computing Laboratory, Claremont Tower, Claremont Road, Newcastle upon Tyne, NE1 7RU, England.
Bibliographical details

DOBSON, John Edward

Conversation structures as a means of specifying security policy. [By] J.E. Dobson
Newcastle upon Tyne: University of Newcastle upon Tyne: Computing Laboratory, 1989.

(University of Newcastle upon Tyne, Computing Laboratory, Technical Report Series, no. 295)

Added entries

UNIVERSITY OF NEWCASTLE UPON TYNE.
Computing Laboratory. Technical Report Series. 295

Abstract

This paper examines the concept of a security policy. It proposes speech act theory as a language for discussing the semantics of information systems and argues that a security policy must, in addition to specifying constraints on the instrumental acts of authorised access to data, also be capable of expressing the valid sequence of speech acts that constitute the authorisation and create commitment to the policy.

About the author

Mr. Dobson is a Principal Research Associate at the University of Newcastle upon Tyne, with particular interest in a system approach to security. He has been active in the field since 1982 and is the author of more than a dozen papers.

Suggested keywords

COMPUTER SECURITY
DATABASE ENGINEERING
INFORMATION ENGINEERING
SPEECH ACTS
SYSTEM MODELLING

Suggested classmarks (primary classmark underlined)
Dewey (18th): 658.47 001.64404
U.D.C. 519.718 519.687
CONVERSATION STRUCTURES
AS A MEANS OF SPECIFYING SECURITY POLICY

John Dobson
Computing Laboratory
University of Newcastle upon Tyne
NEWCASTLE NE1 7RU
United Kingdom

ABSTRACT

This paper examines the concept of a security policy. It proposes speech act theory as a language for discussing the semantics of information systems and argues that a security policy must, in addition to specifying constraints on the instrumental acts of authorised access to data, also be capable of expressing the valid sequence of speech acts that constitute the authorisation and create commitment to the policy.

This is Technical Report number 10 produced by the Computing Laboratory, University of Newcastle upon Tyne
for the Royal Signals and Radar Establishment, Malvern
as part of the research contract entitled 'Security Modelling for Distributed Systems'.

This paper is to be presented at an IFIP WG 11.3 workshop on Security and Databases to be held in Monterey, CA, in September 1989.

Copyright (c) HMSO, May 1989.
1. INTRODUCTION

There seems to be growing understanding[3, 12] that developing the information aspect of computer systems requires two very different activities, which we might call 'database engineering' and 'information engineering'.

The universe of discourse for the database engineer is confined to data, and devices and processes for manipulating data, which is on the whole quite a tractable area of problems — or puzzles, since they have solutions. We can now accept that formal analysis and methods of proof are (or may become) the right tools for dealing with some of the central problems of understanding database engineering. Mathematics and logic are the key disciplines of the database engineer, who has to deliver a system demonstrably capable of effecting the desired data transformations with reasonable efficiency and economy.

The information engineer, on the other hand, has to examine the validity, effectiveness, meaning and reference of the information. The database engineer's ability to manipulate, store and distribute information correctly is no guarantee of the information's usefulness in the context of a business enterprise. The quality of information must be decided by the information engineer in the light of its fitness for some human purpose, by its ability to provide knowledge or guide people in their actions in a sense appropriate to the enterprise and its policies. The information engineer must specify a system whose data have meanings in terms of the real world of the enterprise. We have to accept that logic and mathematics, though helpful, cannot furnish total solutions to the problems — not puzzles, since they have no solutions — of information engineering.

Consider the following example. A statistical database is being interrogated by a statistician to determine the numerical characteristics of some variable of interest. Some simple standard techniques for preventing inference from a statistical database are implemented by slightly adjusting the values as between the conceptual and external schemas. However, a Data Protection Act insists that a data subject has the right to see the unadjusted data about himself or herself. In this example, the task of the database engineer is to arrange for the provision of the appropriate information, possibly adjusted in accordance with some bias function whose mathematical specification is also a matter of database engineering. The information engineer, on the other hand, has to decide on the appropriateness of the information held in the database, the sensitivity of each data item, which data entities attributes and relations fall within the scope of the Act, and so on. Thus a query on a particular record from an application may require or result in some negotiation, hidden from the user, as to whether an adjustment should take place or not — depending on the application semantics. ("Is this request for statistical purposes only or is it in accordance with the provisions of the Act?")
The issue here is that before a request can be satisfied, some hidden negotiation takes place as to what should be said in response to the request. This hidden negotiation does not necessarily have to take place at the time the request is actually made; it could have taken place at some earlier epoch, for example during the system design phase. The important point is that in order to deny or grant the request and to determine the nature of what is granted, some conversation must at some previous stage have taken place between relevant parties to decide upon what information policy the database client/server relation is going to support. This pattern of hidden negotiation is very common: we see it in remote procedure call protocols, security filters, atomic transaction structures, user interface presentation, and many other places. It is in fact part of almost every process of deciding how to interpret data in order to provide information.

2. CONVERSATIONS AS INTERPRETATION STRUCTURES

The particular relevance of this observation regarding conversation structures to issues of security policy is that these negotiations are just as much a matter of security policy as are the details of the data access rules. Who decides, and by what authority, what is deemed to be a subject and what is deemed to be an object? Just what is the process of negotiation that results in this deeming?

Consider another example.

(i) An evaluator requires that software to be certified must be mathematically verified using only a certified theorem prover.

(ii) It is convincingly shown that the only available certified theorem prover is also capable of proving the following

\[ \text{theorem: } FALSE; \]

and hence is mathematically suspect.

The obvious question that now arises is the following:
Under what circumstances is it reasonable to invoke the meta-rule (escape clause)

(iii) If the software developer, design authority, customer authority, and certified evaluator all agree, then (i) and (ii) are deemed not to be in conflict.

The point that we are making here is that in order to invoke the escape clause, the structure of the agreement must be convincing, and be seen to be convincing, and be recorded, in order to satisfy some possible future auditor that the procedures used were both formal and reasonable.

The following section will attempt to address these and other similar questions by showing how to deal with the structure of the exchanges or conversations in which objects are exchanged or operated upon, and — more
importantly — the results of those operations, which are decisions that are made, commitments that are entered into, and so on. Just as the world of data has been articulated into entities, relations and attributes, so the world of information interpretation will be articulated into different sorts of operation and different sorts of result.

The approach to the problem that we are going to adopt is by examining the structure of conversations between a number of parties. The reason for this approach is that, as we hope to show, it brings into the focus of a single framework a number of issues concerned with information usage:

- protocol structures for ascribing meaning to information
- protocol structures for identification and authentication of communicating entities
- notions fundamental to the object metaphor (e.g. trading — the handling of import/export relationships)
- database transaction and commitment protocols, particularly in a distributed environment
- policies (e.g. security policies) over information storage, access and manipulation.

It is impossible to express concepts without language. The purpose of the speech act model presented below is to provide a general vocabulary for expressing important distinctions in information-yielding operations on data objects. A model-less statement of the distinctions could only be so general as to be meaningless. The selection of the speech act model (instead of some other model) is justified on the basis of two considerations:

(i) it is, philosophically speaking, reasonably well-understood;
(ii) it has proved useful in various computer-related disciplines such as AI, office automation, and human-computer interfacing.

3. SPEECH ACT THEORY

This section introduces a method for modelling information systems as systems of communicative action. The actions — making promises, negotiating contracts, stating facts, and so on — are what give the data its meaning. The interpretation that a particular item of data represents an individual's salary is a matter of fact; the interpretation that a particular document is a legal instrument and represents a negotiated contract is not just a matter of fact; it represents also a set of obligations. The processes that are executed in the computer can also represent human actions that create, modify and delete commitments that bind current and future behaviours and give meaning to the information that is exchanged during these actions and processes. There needs to be a language for describing the types and structures of these actions as well as the types and structures of the information
that mediates them. Such a language would regard information systems as social systems that are technically implemented. It would see the information system as human action even when a computer performs calculations, accesses data, or transmits messages. For example, order files are commitments that have been made but not fulfilled, sales invoices create obligations to pay, and so on.

The following subsections introduce speech act theory as a vocabulary for describing the sequences of actions that result in obligations and actions. It is a simplified version of the exposition written by Auromäki et al[1], who also provide a detailed example of its use and some further development which is not discussed here.

3.1. Basic Concepts

In our earlier work[7] we argued for describing an organisation for the purposes of enunciating a security policy in terms of three domains: agents, actions, and data. These constitute the basic entities of our model. The acts performed by the processes that may induce changes in an entity domain can be classified into two disjoint sets: instrumental acts and speech acts.

Instrumental Acts are deeds performed by the human/mechanical process by means of actions. Their preconditions, initial states, and results can thus be described by referring to states in the entity domain. Examples of an instrumental act are to invoke a compiler or to deliver some goods to a customer.

Speech Acts are spoken or written utterances that result in meaning being assigned to a linguistic expression. Speech acts always involve at least two agent roles: speaker and hearer (though these can on occasion be the same individual). Speech acts form larger wholes called conversations, which exhibit systematic regularities that can be studied and analysed. An example of a conversation is to authenticate someone; it consists of a number of speech acts such as requesting a token of identification, confirming its authenticity with an authority, and finally accepting the individual as genuine (or not).

We define an activity as an institutionalised routine governed by fixed or fluid arrangements that involves both instrumental and speech acts.

3.2. Speech Acts

A speech act is the basic unit of communication. There are several types of speech acts, of which the three main categories are propositional acts, illocutionary acts, and perlocutionary acts.

A propositional act simply expresses a fact, such as "It is Christmas Day today" or "There'll always be an England". Propositional acts can be evaluated to be true or false (though it may not be easy to determine which,
as in the second of our examples). We do not say anything more about how the evaluation is performed, nor about the theory of truth we are assuming. Propositional acts can meaningfully be uttered by people or machines.

An *illocutionary act*, or *illocution*, is a basic unit of human communication; it is always performed when a person utters certain expressions with an *intention* — for example, "I promise to write the letter" or "I refuse to pay the bill" (a positive statement of a negative intention). When the intention has been recognised by the hearer(s), the illocutionary act has been successful, and we say its meaning has been understood. Questions of the truth of an illocution do not arise; rather, the act creates a *commitment* that in a moral sense binds the future behaviour of the parties and pledges them to certain activities or expectations. Illocutionary acts can be expressed through mechanical means as well as vocal means. They are, however, always an expression of human concern or intention.

A *perlocutionary act* is an act that produces effects on the feelings, attitudes or behaviour of the hearer(s), for example the act of getting someone to write a letter on request. Again, truth of a perlocutionary act is not an issue; success is, and occurs when the perlocutionary act has its desired effect. Perlocutionary acts are not further considered here.

### 3.3. The Structure of an Illocutionary Act

In general an elementary illocutionary act has two components: an *illocutionary force* and a *propositional content*. These two components can be distinguished in any meaningful utterance. An illocution also takes place in an *illocutionary context*. All these constituents are important in understanding the meaning of an illocutionary act.

The *content* refers to the propositional act embedded in the illocution. For example, the content of "I refuse to pay the bill" is "I will not pay the bill", considered as a (true of false) statement of future fact.

The *force* is the fundamental component of an illocution. It determines the social relationships or commitments established and the way in which the content is related to the world. Illocutionary force has seven components altogether[11], of which three are dealt with in the next subsection. (The other four are less relevant and will be ignored.)

The *context* of an illocution is defined by the speaker, the hearer(s), the time and place of utterance, and the set of possible worlds. The first four are self-explanatory; the fifth is the set of the speaker's and hearers' contexts of interpretation.
3.3.1. Illocutionary Force
The three components of illocutionary force that will be discussed are:
(i) illocutionary point
(ii) propositional content conditions
(iii) preparatory conditions

(i) Illocutionary Point
Each type of illocutionary act has a point or purpose which is of the essence of its being an act of that type. Thus illocutionary point is a categorisation over illocutionary acts. There are a number of categories of illocutionary act; one categorisation[11] claims that there are just five categories or illocutionary points, as follows:

<table>
<thead>
<tr>
<th>category</th>
<th>objective</th>
<th>examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>assertive</td>
<td>to say how the world is</td>
<td>to state, to predict</td>
</tr>
<tr>
<td>commissive</td>
<td>to commit the speaker</td>
<td>to promise, to agree</td>
</tr>
<tr>
<td>declarative</td>
<td>to change the world by saying so</td>
<td>to christen, to deem</td>
</tr>
<tr>
<td>directive</td>
<td>to get the hearer to do things</td>
<td>to order, to request</td>
</tr>
<tr>
<td>expressive</td>
<td>to express the speaker's attitudes</td>
<td>to apologise, to console</td>
</tr>
</tbody>
</table>

Other categorisations are possible, of course.

(ii) Propositional Content Conditions
The content of an illocution cannot be arbitrary, but must satisfy certain propositional content conditions. For example, the propositional content of the sentence "I order you to draw a triangular square" cannot be satisfied. These conditions are analytic conditions (i.e. ones which can be determined from consideration of the propositional content alone).

(iii) Preparatory Conditions
Preparatory conditions specify the state of affairs that a speaker must presuppose to exist in the world if an intended illocutionary act is to be performed. For example, in placing a purchase order the buyer presupposes that the supplier has not gone bankrupt, has the same address as earlier, and still sells the ordered products. These conditions are synthetic, i.e. they depend on empirical facts.

3.4. Conversation Structure
A conversation is a sequence of speech acts which is intended to fulfil a specific social goal (e.g. validating and authorising a new computer user). It is the largest unit of communication in our model.
A *speech act pattern* is a sequence of speech acts that in some sense forms a distinct logical unit in a conversation. The notion of speech act pattern is crucial, since it is used to validate that a conversation is coherent and complete. The relations between the various illocutions that together form a speech act pattern is defined by what we term a *conversation program*. These programs not only include references to linguistic structures but also involve empirical knowledge such as the current states of the speakers, hearers and possible worlds. Conversation programs include procedural rules that define the flow of control and actions taken during each *stage* (see below) in a conversation. They define what speech acts may be performed under specific conditions.

Speech acts in a pattern can be grouped into sets of alternatives, depending on the actual conduct or execution of the pattern. There might, for example, be separate validation procedures depending on whether the candidate for validation is a British citizen, an EEC resident or a foreign national. These alternatives form a *stage* in a conversation. The set of alternative stages obviously depends on the type of the conversation.

*Moves* are speech acts that activate stages in a conversation. They thus control the flow of the conversation process through its various stages.

4. **WHAT IS A SECURITY POLICY?**

'Policy' is a particularly difficult word to give a dictionary definition for. It combines notions of action, desirable future states of affairs, constraints, and pious hopes. Similarly it is so widely used and in so many different contexts that a definition on the basis of use is likely to be equally difficult. There is, however, one specific sort of use of the word that is of importance to security concerns and which our language must be able to express. The example we have in mind is its use in a phrase such as 'Security Policy', where the intention is to be able to state what sorts of patterns of accesses to information are to be regarded as valid or legal, and what sorts of accesses are to be regarded as a violation of the security policy. The seminal work in this area is that of Bell and LaPadula[2].

Bell and LaPadula showed that it was possible to give a fairly simple state-based formalisation of security in terms of the passive objects (e.g. files and directories) held in a computer, and the subjects (e.g. programs and processes) which act upon them. Having produced an abstract model of the state of a computer system, Bell and LaPadula identified classes of operation which changed the system state, then specified the security policy in terms of changes to that state. This gave rise to the well-known *simple security* property that a subject cannot read an object above its clearance, and the *star* (*) property that a subject may not write to an object below its clearance. Bell and LaPadula gave both an abstract definition of the model and showed how to interpret the model for real computer systems, e.g. Multics
Other security policies have been proposed[4, 8] and it is generally agreed that there is no universally applicable security policy equally suited to all kinds of enterprise. What is possible, and what this section explains, is a language for expressing policies of the sort exemplified by the security policies just mentioned. Although for clarity in exposition we shall talk about security policies and use security-related terms (such as authorisation, authentication and so on), it is important to realise that this is just a specific example of one sort of policy that can be expressed in the linguistic apparatus we have so far presented. Other sorts of policy could equally well be stated, as for example safety policies or maintenance policies.

4.1. Application of Speech Acts to Security Policy

We first distinguish between a Security Model, which presents high-level concepts and the nature of a formal language and its associated semantics, in which the abstract property 'security' is defined, and a Security Policy Model, which is a refinement of the security model obtained by incorporating policy rules and assigning real world things and categories of things to the formal elements defined by the formal semantics of the security model.

In the case of the Bell and LaPadula model, the security model is expressed in terms of state machines which define the abstract notion of security: the security policy model for a particular system states what counts as subjects and objects, what rules govern the assignment of security levels, and so on.

In our view of information systems based on speech act theory, we define a Security Model, in its most abstract form, as a relation between a set of illocutionary acts in a conversation and a set on instrumental acts in an activity; and we define a Security Policy Model as a basic unit of the conversation consisting of illocutionary acts in which such things as

- subjects and objects are nominated
- security attributes are asserted
- security classifications are assigned
- security authorisations are allocated

and so on.

We will now explain the rationale for these definitions, and give an example of their use.

Security Model: a relation between a set of illocutionary acts in a conversation and a set of instrumental acts in an activity.

The view of security here being espoused is expressed by "Provided these procedures have been invoked and correctly executed, it is legal for this agent to do this action". Thus in order for a trusted subject to read a classified object, the subject must have been cleared through the appropriate
social procedures, and have been given a token of clearance to present to the authorised guardian of the classified object. Similarly the guardian must be able to prove that it was in fact duly authorised, the object must have been classified by an agent authorised to classify, and so on. All of these matters are issues of security policy (as we are using the term). It is a major defect of the Bell and LaPadula model of security that while it well expresses (one view of) security as a matter of who can do what to whom, it is powerless to express the social procedures by which its access rules come into being and which ascribe authority and security attributes to the subjects and objects of which it treats. In our terms, such models attempt to model security purely in terms of instrumental acts in an activity, since they do not have even the concept of illocutionary acts in a conversation. But both are needed in order to express all aspects of a security policy, and it is the relation between them that provides a more coherent and complete answer to the question What is security?

The close relation between instrumental and illocutionary acts in the expression of policy does have some implications for the database engineer. It means that certain instrumental acts which change data structures do so in order to bear witness to certain acts of policy commitment. This may well generate extra dependability requirements on those instrumental acts (e.g. that they cannot subsequently be undone or repudiated). Thus some dependability requirements can be derived from the roles of the instrumental acts in the commitment structures.

Security Policy Model: a basic unit of a conversation consisting of illocutionary acts in which a security model is instantiated.

To make the discussion somewhat clearer, we shall assume the following toy Universe of Discourse. The world consists of Buyers and Sellers, who exchange Commodities which are Carried by Carriers. Buyers, Sellers, and Carriers are either Trusted or Untrusted. Commodities are either Valuable (e.g. money, paintings) or Worthless (e.g. invoices, Christmas cards). The primary instrumental act is one of Carriage, though in practice there needs to be some secondary ones also.

Associated with this system, there could be obvious security rules in the security policy model of the form Valuable Commodities can be Carried only by Trusted Carriers; a Buyer will buy a Valuable Commodity only from a Trusted Seller, and so on.

From the point of view of our information modelling language, however, what we want to represent is not these sorts of rules, which are easy enough to represent in any security policy model, but the rules which govern the answers to questions such as How does a Carrier become Trusted? How does a Buyer know whether a Seller is Trusted? Even more possibly catastrophic importance of events following a security violation, are the answers to questions such as What happens when it becomes clear that a Carrier is no
longer to be Trusted? when a Buyer discovers that an alleged Valuable Commodity is actually Worthless? It should be clear that, even in our toy Universe of Discourse, there are questions of security policy which cannot be answered simply by looking at the security attributes of the entities but at the procedures by which these attributes are assigned, promulgated and revoked; and the job of the security model is to be able to represent those procedures, not only so that logical problems such as well-foundedness (‘Trust’ can only be assigned by a Trusted Authority, hence a Carrier cannot simply declare herself† Trusted) and termination (hence there must be at least one initial Trusted Authority) can be analysed; but also so that the parties can agree to the representation of the procedures.

These issues of security policy can be cast into the form of a set of relations between illocutionary and instrumental acts. For example, to take the question of how a Carrier becomes Trusted.

(1) a Carrier is vetted by a Trusted Authority by means of a sequence of questions and responses (which may include authentication to the Carrier of the Authority)

(2) if satisfied, the Authority deems the Carrier to be Trusted;

(3) the Carrier is given a token of Trustworthiness;

(4) the newly-Trusted Carrier now legitimately plies her trade.

— and all of these must occur in the order stated.

Of these, (1) (2) and (3) are illocutionary acts, whereas (4) is an instrumental act. (1) (2) and (3) together with the temporal constraint that (1) precedes (2) precedes (3)) form an example of a conversation program. (1) (2) and (3) by themselves are an example of a speech act pattern. The above example is thus a fragment of a security policy model which exhibits the general form of the security model we are advocating, i.e. a relation between speech and instrumental acts.

Similar examples could be set up to show how the other questions we have raised could be answered. We need not do this in detail, since the purpose of our paper is not to exhibit a detailed security policy model but to make the general point that issues of security cannot be modelled without considering both the instrumental acts carried out by the system of agents and the illocutionary acts uttered with some intention by human participants.

5. INFORMATION SEMANTIC LANGUAGE FEATURES

Now that the speech act model has been introduced, we can state some important features of an information processing language that accesses and interprets data items. This discussion will use the terms introduced by

† There are only female Carriers in our Universe
means of the speech act model, but is not meant to be unduly prescriptive on the implementation model. It is recognised that the use of the speech act model in explaining the structure of the information modelling language could bias a developer in various undesirable ways towards implementations similar to the language. This is not intended. Speech act theory provides a useful vocabulary for talking about the issues which an implementor must address, but any implementation which supports the concepts is entirely acceptable.

The following features of an information semantic modelling language are important:

1 \textit{The description of the purpose of the conversation.}

Purpose is described in terms of illocutionary points of the individual speech acts. This paper does not prescribe any particular categorisation of illocutionary points. Experience of the use of speech acts indicates, however, that in order to determine whether a particular conversation is well-formed (with respect to its social function), it is necessary to be able to make certain distinctions, e.g. between requesting and responding to a request, between agreeing to an offer and making a counter-offer, between committing a transaction and aborting a transaction. All of these distinctions are distinctions of illocutionary point. It is an application-dependent decision as to what different sorts of illocutionary point there are.

2 \textit{The description of the conditions for successful conversation.}

Any conversation can only be held on the basis of certain assumptions, which are described in the preparatory conditions for an illocution. Of course the complete set of assumptions can never be stated in its entirety, since it is probably not finite, or only just so. However, the more important assumptions, which in general will be those about the assumed impossibility of certain failure modes, should be made clear.

3 \textit{The emphasis on describing the understandability, coherence and completeness of the conversation.}

Understandability, completeness and coherence of a conversation are expressed in terms of the structuring of a conversation into speech act patterns, sequencing, reference to context, the expression of alternatives, control over dialogue structure, and conditions and constraints. The collection of the individual speech acts together with the description of the superimposed structure form what we have called the conversation program. Representations of the individual speech acts in the program can be installed as data items in the database. The
structure can then be represented as a set of relations over those data items, and can therefore (as is desirable) be kept separate from them.

4 *Simultaneous analysis of communication and organisational tasks.*

Speech act theory makes the distinction between the instrumental act of asking a database for a person's salary in order to write out a monthly pay cheque and the speech act of asking a database whether a given password is valid in order to establish a person's identity. The difference is that the latter act, unlike the former, is concerned with the creation of a commitment that binds the future actions of the requesting program; it is about the establishment of trust rather than the mere transmission of information. The reason why this distinction is important is that it may well influence design decisions concerning the privacy, security, integrity and reliability constraints on the means of handling and transmitting the message; for example, different channels with differing dependability characteristics may well have to be used for instrumental acts and some speech acts.

5 *The ability to take a more complete view of a policy statement.*

The reason for examining speech acts at all is that they, and they alone, establish the basis on which obligations (accountability, responsibility) are established. It is these obligations that are fundamental to a statement of policy.

One of the more interesting uses of the kind of conceptual model of conversations we have introduced is that it makes it possible to perform an analysis of the various failure modes of a security policy. To take our previous toy example, one of the questions that could be asked is What happens if for some reason the supply of tokens of Trustworthiness dries up? Clearly a preparatory condition for (3) is the existence of a token, and one would expect an analysis of the policy to identify situations in which this condition could not be fulfilled. Since the purpose of the token is presumably to prove to all interested parties that the Carrier is now indeed Trusted, then perhaps some authenticated broadcast protocol would have the same effect. However, if — as is often the case — the token serves the dual purpose of being both a token of authorisation and a token of identity, then an authenticated broadcast protocol structure could replace only the first function of the token and not the second. Any analysis of the function of the token would require an analysis of its use in subsequent speech act structures, perhaps those concerned with the instrumental act of submitting a Valuable Commodity for Carriage. It is this kind of analysis that the method presented here readily permits. A systematic approach to the analysis of conversation structures which, it is claimed, is capable of
determining and classifying all the possible failure modes of this kind of structure is outlined in previous work[5].

6. FUTURE RESEARCH DIRECTIONS
The approach advocated here is being used as the basis for an ongoing EEC-funded research project into requirements definition methodology for non-functional requirements (e.g. safety, security), and, as part of another EEC project, for defining security policies for a large scale distributed applications system where a single security policy cannot be assumed – for example because of the lack of a single jurisdictive authority over the whole of the network and all possible application domains. (Some of the problems in this area were identified in a previous workshop in this series [6].) Simple graphical tools to support the application of the method are currently being designed. The major outstanding problem, however, is the choice of an appropriate formalism.

It might seem that for many speech act purposes, some form of modal logic could be required. This certainly is the view of the Alvey FOREST project, for example, which is developing a form of modal logic called MAL (Modal Action Logic [9]). There are, however, a number of reasons why a modal logic is not necessarily best suited for the purpose [10].

The most compelling reason that convinces us that an approach to security modelling based on modal logic (or indeed on any classical logic) is unsatisfactory is that all such approaches implicitly assume as an article of faith the slogan "Formalisation Requires Axiomatisation"[10]. The immediate leap of faith into writing down axiom sets and making deductions from them is all very well and lots of people do it correctly, but correct behaviour does not always imply full understanding.

In a way, the axiomatic method has been too successful. Propositional calculus, first-order logic, Euclidean geometry are all completely axiomatised, and everything is so very neat and precise. But the neatness and precision, the very fascination of the axiomatic approach, makes it difficult to conceive of alternative approaches to calculi, logics and geometries. The proofs of soundness and completeness of the axioms are not sufficient evidence that one's concepts are sound or adequate. (Compare the now widely recognised need in software engineering for both verification – Are we Building the System Right? – and validation – Are we Building the Right System?)

That axiomatisation is not always necessary for formalisation can be seen by considering Intuitionist Logic. The underlying concept of Intuitionism is that of mathematical construction as a cognitive process. Here construction implies proof. Brouwer regarded proofs as abstract objects to be manipulated in the same way one manipulates other mathematical objects. This idea is familiar to computer scientists brought up to believe that development of a program and development of a proof go hand-in-hand. But there
is not an axiom in sight. (Of course I am aware that Intuitionist Logic has its own — very neat — set of axioms due to Heyting. But my point is that these axioms can only be understood and justified by reference to this concept of mathematical construction.)

In the case of security modelling, although I believe one cannot simply start from axioms, formal work is not beside the point. The trouble with the axioms of modal logic is that they are assumed to be the same in all possible worlds. Unfortunately this does not include the real world, in which the way people divide things into subjects and objects, authorise access to such things, set up multi-level categorisations over such things, and so on, often and rightly depends on the role of the person doing the division, authorisation and categorisation. Formally the axioms do depend on the possible world.†

The point here is that modal logic, like all forms of symbolic logic with a set-theoretic possible worlds semantics, is capable of representing only one form of cognitive activity, and that is reasoning. An information system, though, is of wider scope and has to permit other forms of discourse — for example, allocating and accepting responsibilities, making commitments, negotiating over requests. These human activities do not always obey the axioms of modal logic. So although a modal logic can well describe reasoning about a system, it is incapable of describing acting when situated within a system, and it is the latter rather than the former that is of interest to us.

The speech act approach moves the focus of attention away from formal reasoning as the basic cognitive activities undertaken by the human user of the information system, to the social activity by which we generate the space of co-operative actions that the enterprise as a whole undertakes. In taking this perspective, one is not denying the obvious fact that people in their jobs do in fact collect information and reason about that information in the context of making decisions. Rather, one is taking a complementary view that shifts the emphasis to the actions by which groups originate and co-ordinate a network of interconnected activities and obligations, and considers the commitments that individuals impose on each other and on the system. The question that we now wish to investigate, then, is the following: given that speech act theory provides an adequate basis for some aspects of conceptual modelling of this activity, what sort of logical formalism is required, and how do we translate the one to the other?

†We note, incidentally that a symptom of deficiency in modal logic is that even in MAL — as at present described — there is a disparity between the expressive power of the notation and examples of its use, in that in order to be convincing the examples make ad hoc extensions of the defined notation. The technical point seems to be that MAL, like any other modal logic, treats its obligation operators as second-class citizens.
Acknowledgements

I wish to thank Mike Martin and John McDermid for all our long discussions, many of them useful. Security work at the University of Newcastle upon Tyne is funded by RSRE Malvern, and I am grateful for the tolerant support of Ruairidh MacDonald.
References


