Speech Acts, Electronic Commerce, and KQML

Michael A. Covington

Artificial Intelligence Center

The University of Georgia

Athens, Georgia 30602–7415 U.S.A.

mcovingt@ai.uga.edu

Revised 1997 June 9

Abstract

Speech act theory (the study of how utterances function as statements, questions, commands, etc.) is increasingly applicable to software design. Moore has found that the repertoire of speech acts used in electronic com-

munication is close to that used in human speech.

This paper examines the encoding of speech acts in KQML, a knowledge interchange language developed with ARPA funding; contrasts KQML with human speech and conventional EDI; and suggests ways of improving KQML.

I conclude that although speech act theory is highly relevant to electronic communication, the needs of computers are different from those of humans. Computers need to perform concisely speech acts that are clumsy in human speech, such as arranging communication paths. They also need to recognize speech act types as immediately as possible, whereas human language gets along with clumsy encodings of speech acts into grammar.

1 Speech acts in electronic communication

Speech act theory — the study of how utterances function as statements, questions, commands, and so on — is no longer just an area of theoretical linguistics; it is finding increasing applications in software engineering. Several groups of researchers are experimenting with knowledge interchange languages based explicitly on speech act theory [7, 17, 12, 8]. Further, Scott Moore [18] has opened an important line of investigation by comparing the

repertoire of speech acts used in electronic communications with those used in human speech.

Moore analyzed the illocutionary force of X12 EDI transactions, S.W.I.F.T. securities transactions, and Apple Events in the Macintosh operating system. He found, perhaps surprisingly, that all of these electronic messages display much the same variety of speech act types as human speech. There are a few gaps; for example, computers do not normally express condolences to each other. But the applicability of human-language speech act theory to electronic messaging, even the internal messages used within an operating system, is impressive.

Nonetheless, electronic communication is not human speech. It is time to look more deeply at speech act theory from the viewpoint of software engineering as well as linguistic description. In this paper I will raise some methodological points, then examine the usage of speech acts in KQML, a new speech-act-based knowledge interchange language, and briefly contrast KQML with conventional EDI.[1]

[1]An earlier version of this paper was presented at the 1997 Hawaii International Conference on System Sciences. I am indebted to Steve Kimbrough and Roggie Boone for helpful conversations and encouragement relating to this work.

2 The central claim of speech act theory

2.1 The "Vulcan mind meld" theory of communication

Perhaps the best way to introduce speech act theory is to compare it to a naive view of communication that does not recognize speech acts. On that naive view — known to *Star Trek* fans as the "Vulcan mind meld" — communication is simply the transmission of thoughts or knowledge from one mind to another. When you connect your brain to mine, you know what I know.

That is indeed how computer-to-computer communication has often been approached. Networks allow one machine to mount another machine's disk drives; EDI forms such as ANSI X.12 [9] allow one program to stuff data into variables in another program. Distributed databases enable computers to share non-trivial knowledge structures.

But Vulcan mind melds do not occur in human experience, and they are a poor model of how humans actually communicate, for at least three reasons. First, my thoughts are not your thoughts; they are of no use to you unless I express them in a common language, making appropriate assumptions about background knowledge. This, indeed, is the problem that standard EDI

formats and knowledge representations address.

Second, the Vulcan mind meld theory ignores the voluntary nature of communicative acts. I can't give you all my thoughts; I have to select particular things to say at particular times. Thus, alongside syntax and semantics, every language needs rules of PRAGMATICS, the knowledge of what to say when.

Third, and perhaps most importantly, I can't just deliver my thoughts to you; I have to tell you what I want you to do with the propositions that I express. If you can't distinguish statements from conjectures, questions, my utterances will be of no use to you. That is where speech acts come in: effective communication requires accurate recognition of speech acts.

2.2 The $F(P)$ hypothesis

The central claim of speech act theory is that people do not just utter propositions; they perform ILLOCUTIONARY ACTS such as stating, requesting, commanding, and so forth. Every speech act consists of an ILLOCUTIONARY FORCE F applied to a proposition P. This is known as the $F(P)$ hypothesis. The importance of illocutionary force was first made explicit by Austin [2]

Table 1: Various illocutionary acts with the same or nearly the same propositional content.

Illocution	English sentence
Statement	The cat is on the mat.
Question	Is the cat on the mat?
Command	Put the cat on the mat.
Polite request	Could you put the cat on the mat, please?
Promise	I promise that the cat will be on the mat.
Guarantee	I certify that the cat is on the mat.
Offer	I'll put the cat on the mat if you'd like.
⋮	⋮

but was foreshadowed by the semantic theories of the ancient Stoics.[2]

Moore summarizes the $F(P)$ hypothesis as claiming that "the outermost [logical] operator of *every* utterance (everything we could possibly say) is not Boolean, not temporal, not even defeasible — it is an illocutionary force" [18]. Further, this outermost operator is never vacuous; that is, $F(P) \neq P$. Even when stating a fact, you are making a statement, not just voicing a fact.

Table 1 shows several different illocutionary forces applied to the proposition "The cat is on the mat." Many more illocutions are possible, many of them restricted in various ways; to take an extremely specialized exam-

[2]Diogenes Laertius (*Lives of the Philosophers* VII.65–68) divides utterances into statements, yes/no questions, questions seeking information, commands, oaths, acclamations, and exclamations.

ple, christening a ship — which is a speech act — is possible only in a very specific setting.

2.3 Some distinctions

Illocutionary force is distinct from grammar, meaning, and perlocutionary effect. Taking the last of these first, the PERLOCUTION of an utterance is what it actually accomplishes, such as informing, persuading, dissuading, and the like. Illocutions and perlocutions are closely related, but discrepancies can easily arise. For example, by inviting you to do something in a particular way, I may end up actually dissuading you from doing it. The speaker controls the illocution but only attempts to control the perlocution.

Illocution is also distinct from meaning. Questions about the weather are no different, as far as illocution are concerned, from questions about dogs and cats; the only difference is in the propositional content. Whenever a classification of speech acts becomes excessively fine-grained, one suspects that the classification is picking up distinctions of meaning as well as illocution.

Bierwisch [5] points out that this mistake is especially easy to make when the information content of the utterance refers to a speech act – that is, when

P contains another F. Some speech acts refer to others; for example, "Please tell me your name" is a request for a statement. Nonetheless, the utterance itself is one speech act, not a combination of them — it is a request *referring to* a statement, not a request *combined with* a statement. Logically, it is $request(tell(name))$, not $[request + tell](name)$.

Finally, the encoding of illocution into grammar in English is notoriously non-uniform. Some speech acts are encoded by particular syntactic structures (statements, questions, exclamations); others are encoded by particular verbs (promise, accept, nominate); and still others, the most specialized, are performed by asserting that one is performing them, such as "I hereby dub thee knight."

The requisite distinctions can be subtle. "I will go to New York next week" can be a statement, an offer, a promise, or even a threat, depending on the context. Human language requires elaborate inference in order for the hearer to identify speech acts. In electronic communication, we want to keep the necessary inference as simple as possible.

Communicative speech acts

Constatives (statements of fact)
  Assertives, predictives, retrodictives,
  responsives, suggestives...
Directives
  Requestives, questions, requirements,
  prohibitives, permissives, advisories
Commissives
  Promises, offers
Acknowledgments
  Apologize, condole, congratulate, greet,
  thank, bid, accept, reject

Conventional speech acts (declarations)

Effectives
  Appoint, nominate, suspend, demote,
  resign, abdicate, arrest...
Verdictives
  Acquit, certify, disqualify, clear,
  rule, adjudicate...

Figure 1: Speech acts as classified by Bach and Harnish.

3 Classifying speech acts

The study of speech acts begins with classifying them, and many rival classifications have been proposed [4, 21, 20, 1].[3] In his study, Moore used that of Bach and Harnish [3], summarized in Fig. 1.

For Moore's purposes this classification is ideal because it makes as many

[3]See [21] for a particularly good overview up to 1983.

distinctions as possible, thereby enumerating the whole range of human speech acts. But the Bach-Harnish classification is less than ideal for shedding light on how speech acts actually work. In some respects it is more a collection of data than a theory. Notice the large number of verbs that are in classes by themselves.

A more insightful classification will take into account the fact that speech acts differ along more than one dimension [19, 1]. For example, the difference between a command and a polite request, or between a confident assertion and a cautious suggestion, is not a *logical* difference; it is a difference of strength. Similarly, the difference between a promise and a threat is whether the affected person wants the thing to happen. Indeed, one important dimension is whether the speech act pertains primarily to the state of the speaker, the state of the hearer, or both.

Accordingly, not every distinction calls for another leaf in the Bach-Harnish classificatory tree. An alternative approach is to distinguish a smaller number of basic speech act types and equip each speech act with parameters to describe further aspects of its illocutionary force. One then ends up with a matrix rather than a tree.

What does all this imply for engineering? Three things. First, it is

reasonable to want the encoding of speech acts in an artificial language to be syntactically uniform. An utterance should wear its illocutionary force on its sleeve, so to speak, for the convenience of the routines that process it. Second, the set of speech act types should not be too large; instead, parameters should encode subtle variations on basic types. Third, the inference required on the receiving end should be held to a minimum. Entertaining though misunderstandings or "comedies of manners" may be, we do not want them to become a regular part of electronic knowledge interchange.

4 Speech acts in KQML

4.1 The KQML language

KQML (Knowledge Query and Manipulation Language) is a Lisp-based language that was developed as part of the ARPA Knowledge Sharing Effort [10, 11, 14] and has been implemented by several different working groups. I will discuss first the 1993 version [10, 11] and then the proposed 1997 revision [15]. I shall call these KQML 1993 and KQML 1997 respectively.

The main focus of KQML research so far has been the use of intelli-

```
(ask-one
  :content (PRICE IBM ?price)
  :receiver stock-server
  :language LPROLOG
  :ontology NYSE-TICKS)

(ask-all
  :content "price(IBM,[?price,?time])"
  :receiver stock-server
  :language standard_prolog
  :ontology NYSE-TICKS)
```

Figure 2: Examples of KQML messages.

gent agents to arrange transport and handling of messages. Accordingly, the knowledge content of a KQML message need not be written in KQML; it can be expressed in Prolog or some other language. The KQML wrapper indicates the kind of message, the intended recipient, the language, the "ontology" (knowledge base), and other parameters. Fig. 2 shows examples from [11].

4.2 The KQML performative set

Crucially, KQML is speech-act-based; the message types are performatives. The basic performative set is shown in Figs. 3 and 4.

The first thing one notices is that there is some doubling up of basic

Basic informatives (constatives)
`tell` (share a piece of knowledge)
`deny` (retract or negate a speech act)
`untell` (retract a statement; equals `deny tell`)

Database performatives
`insert` (ask recipient to add something to his KB)
`delete` (ask recipient to delete a fact from his KB)
`delete-one` (ask recipient to delete one of the facts that match X)
`delete-all` (ask recipient to delete all facts that match X)

Responses from recipient
`error` (what you said doesn't make sense)
`sorry` (I can't do what you requested)
(also means "no (more) answers" as in Prolog)

Query performatives
`evaluate` (evaluate an expression; details depend on language)
`reply` (I am sending you data to answer your query)
`ask-if` (yes-no question)
`ask-about` (tell me what you know about X; reply with 1 list)
`ask-one` (send me one response that matches my query)
`ask-all` (send me all responses that match my query)

Multi-response query performatives
`stream-about` (like `ask-about`, but reply with a series of messages)
`stream-all` (like `ask-all`, but reply with a series of messages)
`eos` ("end of stream," marks end of series of messages)

Effector performatives
`achieve` (change things to make X true)
`unachieve` (you need not make X true)

Figure 3: Predefined performatives of KQML 1993.

Generator performatives

`standby` (get ready to give me the answers to this question)
`ready` (I am ready to give you the answers)
`next` (give me the next answer)
`rest` (give me all the remaining answers)
`discard` (you need not give me any more answers)
`generator` (like `standby` + `stream-all`)

Notification performatives

`subscribe` (tell me about all future changes to this data item)
`monitor` (like `subscribe` + `stream-all`)

Capability-definition performative

`advertise` (I hereby announce that I can handle such-and-such messages)

Networking performatives

`register` (I hereby announce that I can deliver messages to X)
`unregister` (I retract that announcement)
`forward` (I am forwarding you this message from X; reply to him through me)
`broadcast` (send this to everybody you know, unless it has looped)
`pipe` (establish a communication path to X)
`break` (remove the `pipe` communication path)
`transport-address` (associate a name with a non-KQML address)

Facilitation performatives

`broker-one` (get somebody to process this message; send me the result)
`broker-all` (get everybody to process this message who can do so)
`recruit-one` (like `broker-one` but have him send result to me directly)
`recruit-all` (like `broker-all` but have them send results to me directly)
`recommend-one` (find me somebody who can process this message)
`recommend-all` (find me everybody who can process this message)

Figure 4: Predefined performatives of KQML 1993 (continued).

performatives: `ask-one`/`ask-all`, `delete-one`/`delete-all`, and so on. I shall return to this point.

Some of the KQML performatives correspond closely to basic speech acts in human language, such as `tell`, `ask-if`, `ask-all`, `evaluate`, and `achieve` (the last of these requests a change in the physical world, as opposed to requesting a reply or a change in a knowledge base). The distinction between requests and assertions is blurred; `insert`, for example, means "Put this information in your knowledge base," which is close, but not identical, to what we normally achieve by telling someone a fact.

Other performatives are negative, serving to undo other performatives. For example, after `tell`ing someone something you can `untell` it and thereby retract your statement. Similarly, `unachieve` cancels a request for a physical act, and `deny` cancels any speech act whatsoever. This solves a problem noted by Moore [17], which is that in a conventional EDI system, it is often impossible to tell someone to disregard an earlier message.

Note that `untell` is a nested combination of `deny` with `tell`, and `unachieve` equals `deny achieve`. I do not think the designers of KQML have fallen into the confusion that Bierwisch warned us about; rather, they are making their vocabulary efficient. Some nested combinations occur so regularly that they

deserve their own lexical encodings.

Database technology looms large in KQML, and many of the performatives resemble the user interface of Prolog. KQML provides `insert`, `delete`, and numerous tactics for obtaining multiple responses to a query — in a list, in a stream of messages, or even by standing ready to deliver additional answers when asked (`standby`). Mechanisms for delivering the answers include the `ready`, `eos` ("end-of-stream"), and `sorry` performatives.

Here `sorry` either means "No (more) answers," like Prolog failure, or means "I can't respond to what you said; it's beyond my computational power." It is surprising that the designers tolerated this ambiguity, since there are situations in which it could cause problems.[4]

Still other performatives have to do with establishing communication paths and finding suitable agents to handle a message. Here, perhaps, is where KQML shows the greatest originality. An agent can advertise its own capabilities and ask other agents who can process a particular kind of message, either through "brokering" (you send it somewhere for me and send me the result that you get) or "recruiting" (you tell me whom to send it to).

[4]Technically, a server that responds `sorry` to all communications is KQML-compliant, although the amount of KQML that it implements is essentially zero; a wag has observed that "KQML means always being able to say you're `sorry`."

These are activities that require complicated utterances in human language, but the designers of KQML felt, probably correctly, that they are going to be so common in electronic communication that they should be treated as basic.

The KQML performative set is, of course, a classification of speech acts, although it is quite different from that used by Bach and Harnish. The KQML performatives do fit into the Bach-Harnish classification, albeit with some risk of triviality, since many of them are requests. What we see in KQML is a classification developed for a completely different purpose, not for studying human language but for conveying electronic communications concisely.

4.3 Parameters in KQML

Each KQML performative is accompanied by parameters that given additional information (Fig. 5). Parameters identify the sender and recipient, provide tags for pairing up messages with their replies, and identify the language and ontology being used.

More significantly as far as illocution types are concerned, one parameter,

`:sender`	*symbol identifying the sender*
`:receiver`	*symbol identifying the recipient*
`:reply-with`	*identifier that must appear in the reply*
`:in-reply-to`	*symbol from* `reply-with` *field of the message being answered*
`:content`	*the content of the message, i.e., P in $F(P)$*
`:language`	*language in which* `content` *is expressed*
`:ontology`	*ontology (knowledge base) used by* `content`
`:force`	*true if the sender will never retract (deny) this message*

Some performatives take additional parameters.
There are defaults for parameters that are omitted.

Figure 5: Basic set of KQML performative parameters (1993 draft).

`force`, can be used to mark a speech act as irrevocable. Other parameters could be defined to encode further distictions in illocutionary force.

4.4 KQML 1997

In 1997, Labrou and Finin [15] proposed a revised specification for KQML. The main changes are the following:

- the semantics is cleaner and more Prolog-like, though still not rigorous; it is based on the concept of "virtual knowledge base," i.e., all the knowledge that an agent has or can infer.

- `ask-about` and `stream-about` are gone, presumably because they do not represent feasible Prolog-like queries;

- `deny` no longer means "retract a speech act;" instead, it means "assert that P is false," and `untell` means "assert that P is not known to be true."

- a number of speech acts have counterparts beginning with `un-`, for retracting them; thus `delete` has been renamed `uninsert` and `unadvertise` has been provided to cancel `advertise`.

- `reply`, `generator`, and `monitor` have been subsumed into `tell`, `standby`, and `subscribe` respectively;

- `pipe` and `break` are absent, presumably because they are too low-level, and `transport-address` is redefined to link it more closely to the activities of agents;

- the `:force` parameter is no longer supported, but `:from` and `:to` have been added as parameters for forwarded messages.

4.5 Critique of KQML

Any evaluation of KQML must take into account the fact that KQML is not a theory of illocution, nor an account of the pragmatics of human speech; it is a tool for prototyping agent-based software. Thus, although it should have a solid theoretical basis, theoretical elegance is not its main goal.

Cohen and Levesque [6] point out three weaknesses in KQML (1993 version). First, the semantics is not formalized and is in some cases seriously unclear. For example, in KQML 1993 it was not clear whether `deny tell` meant to retract a statement or to assert a negative one. This has been fixed in KQML 1997 by redefining `deny`; unfortunately, there is no longer a general way to cancel speech acts.

Second, some KQML "performatives" seem to be perlocutions rather than illocutions, or at least have misleading names. For example, `achieve`, `broker`, and `stream-all` have names denoting the intended effect rather than the speech act itself. This is not a fatal flaw, but it does represent a path which, if followed further, could lead to serious confusion.

Third, and more seriously, KQML provides no way to express commissives (promises), the stuff of which commerce is made. (This is especially the case

now that `:force` has been deleted from KQML 1997.) Cohen and Levesque demonstrate that future tense statements are no substitute for promises; e.g., in 1979 I could have told you whom I was (almost certainly) going to marry, but at the time I had not yet promised to do so.

To this I can add another point: the performative set is too large and lacks orthogonality, encoding in the performatives some distinctions that should have been parameters.

This problem manifests itself in two places. First, instead of the pairs of performatives `ask-one` and `ask-all`, `broker-one` and `broker-all`, `recruit-one` and `recruit-all`, `recommend-one` and `recommend-all`, there should be a parameter indicating whether the recipient wants all possible answers or just the first one. One could go further and give this parameter four values: "give me only a boolean (yes/no) answer," "give me one piece of data as an answer," "give me all answers in succession," "give me all answers in a list." In that case the four varieties of `ask` could be combined into one, although there would be some combinations that are not normally used, such as `stream` asking for a single answer.

Second, as noted, there are a number of pairs of the form `X`:`un-X`, but, in KQML 1997, no general mechanism to cancel speech acts. The 1993 sense

of `deny` needs to be reinstated, and the performatives that begin with `un-` removed unless clearly needed for conciseness.

5 Speech acts in ANSI X12

Now consider ANSI X12 [9], a set of standard forms for electronic data interchange already analyzed by Moore [18] and critiqued by others [7, 13, 17].

The X12 standard is a set of encodings of hundreds of business forms, such as purchase orders, invoices, bills of lading, educational transcripts, insurance claims, and so forth. Notoriously, X12 fails to make generalizations about the knowledge in these forms. Each form is entirely *sui generis,* with a separate form number and a separate syntax. Even basic semantic units such as "number" are not defined; instead, there's a three-digit numeric field here, a five-digit field there, and so on.

Moore [18] showed that X12 messages can perform a wide variety of speech act types. However, nothing like the Bach-Harnish classification is built into X12. Instead, the encoding of speech acts into X12 either misses the point totally, or is brilliantly simple, depending on your point of view. Very simply, each form is a different kind of speech act. The form numbers stand for

illocutionary forces together with schemata for the information content. A purchase order is a request, a bill of lading is a constative, and so forth.

The overwhelming advantage of this system — one that should be preserved as far as possible in more sophisticated system — is that any computer can tell at a glance whether a particular X12 message is one that it can process. Apart from that, of course, X12 is quite unsystematic and ripe for replacement with a true knowledge representation langauge.

6 Toward an applicable speech act theory

What can we conclude from all of this? Several things. First, speech act theory provides considerable insight into the workings of electronic communication. More specifically, speech act theory provides an appropriate way to label utterances so the recipient will know, at least roughly, what to do upon receiving a message. The efficiency of X12 comes from a simple, if uninsightful, encoding of speech acts.

Second, although Moore is quite right in pointing out similarities between human and electronic speech acts, the needs of electronic communication are different from those of human speech. Two differences are brought out

by the design of KQML. Computers often need to do concisely things that are clumsy and require many steps in human language, such as arranging communication paths to other agents. Further, computer communications are influenced by the nature of databases and knowledge bases, which is why so much of KQML resembles the Prolog user interface.

Third, although it is speech-act-based, KQML does not exploit speech act theory as fully and elegantly as it might. I noted that the set of basic performatives could be made smaller by treating the handling of multiple answers as a parameter rather than a difference of basic performative type. A similar point could be made about some of the performatives that deal with communication; they make unduly fine-grained distinctions at the top level of classification.

Nonetheless, KQML is a good start, and together with other proposed speech-act-based languages, it demonstrates the applicability of speech act theory to electronic communication.

References

[1] Allan, K. Speech act classification and definition. Asher, R. E., ed., *The encyclopedia of language and linguistics,*, v. 8, 4124–4127. Oxford: Pergamon, 1994.

[2] Austin, J. L. *How to do things with words.* Oxford: Oxford University Press, 1962.

[3] Bach, Kent, and Harnish, Robert M. *Linguistic communication and speech acts.* Cambridge, Mass.: MIT Press, 1979.

[4] Ballmer, Thomas, and Brennenstuhl, Waltraud. *Speech act classification: a study in the lexical analysis of English speech activity verbs.* Berlin: Springer, 1981.

[5] Bierwisch, Manfred. Semantic structure and illocutionary force. In J. R. Searle, F. Kiefer, and M. Bierwisch, eds., *Speech act theory and pragmatics,* 1–35. Dordrecht: Reidel.

[6] Cohen, Philip R., and Levesque, Hector J. Communicative actions for artificial agents. *Proceedings of the International Conference on Multi-Agent Systems* (1995). Copy obtained from `http://www.cs.umbc.edu`.

[7] Covington, Michael A. Toward a new type of language for electronic commerce. *Proceedings of the 29th Annual Hawaii International Conference on System Sciences – 1996,* vol. 4, 329–336.

[8] Dewitz, Sandra K., and Lee, Ronald M. Legal procedures as formal conversations: contracting on a performative network. *Proceedings, Tenth International Conference on Information Systems* (1989), 53–65.

[9] *Electronic data interchange X12 standards,* draft version 3, release 4. New York: American National Standards Institute, 1993.

[10] Finin, Tim; Weber, Jay; et al. Draft specification of the KQML agent-communication language plus example agent policies and architectures. 1993. Manuscript obtained from `http://www.cs.umbc.edu`.

[11] Finin, Tim; Fritzson, Richard; McKay, Don; and McEntire, Robin. KQML as an agent communication language. *Proceedings of the Third International Conference on Information and Knowledge Management* (CIKM '94). Copy obtained from `http://www.cs.umbc.edu`.

[12] Kimbrough, Steven O., and Lee, Ronald M. On illocutionary logic as a telecommunications language. *Proceedings, Seventh International Con-*

ference on Information Systems (1986), 15–26.

[13] Kimbrough, Steven O., and Moore, Scott A. On automated message processing in electronic commerce and work support systems: speech act theory and expressive felicity. Manuscript, University of Pennsylvania, 1997.

[14] Labrou, Yannis, and Finin, Tim. A semantics approach for KQML — a general purpose communication language for software agents. Manuscript obtained from `http://www.cs.umbc.edu`.

[15] Labrou, Yannis, and Finin, Tim. *A proposal for a new KQML specification.* Technical Report CS-97-03. Computer Science and Electrical Engineering Department, University of Maryland Baltimore County, 1997.

[16] Mayfield, James; Labrou, Yannis; and Finin, Tim. Evaluation of KQML as an agent communication language. Manuscript obtained from `http://www.cs.umbc.edu`.

[17] Moore, Scott A. *Saying and doing: uses of a formal language in the conduct of business.* Dissertation, Ph.D., University of Pennsylvania,

1993.

[18] Moore, Scott A. Testing speech act theory and its applicability to EDI and other computer-processable messages. *Proceedings of the 29th Annual Hawaii International Conference on System Sciences – 1996,* vol. 2, 30–38.

[19] Searle, John R. *Expression and meaning: studies in the theory of speech acts.* Cambridge: Cambridge University Press, 1979.

[20] Ulkan, Maria. *Zur Klassifikation von Sprechakten: eine grundlagentheoretische Fallstudie.* Tübingen: Niemeyer, 1992.

[21] Verschueren, Jef. Review article: Speech act classification. *Language* 59:166–175, 1983.