On Automated Message Processing in Electronic Commerce and Work Support Systems: Speech Act Theory and Expressive Felicity

STEVEN O. KIMBROUGH
University of Pennsylvania
and
SCOTT A. MOORE
University of Michigan

Electronic messaging, whether in an office environment or for electronic commerce, is normally carried out in natural language, even when supported by information systems. For a variety of reasons, it would be useful if electronic messaging systems could have semantic access to, that is, access to the meanings and contents of, the messages they process. Given that natural language understanding is not a practicable alternative, there remain three approaches to delivering systems with semantic access: electronic data interchange (EDI), tagged messages, and the development of a formal language for business communication (FLBC). We favor the latter approach. In this article we compare and contrast these three approaches, present a theoretical basis for an FLBC (using speech act theory), and describe a prototype implementation.

Categories and Subject Descriptors: I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*representation languages*

General Terms: Languages, Theory

Additional Key Words and Phrases: Electronic commerce, formal language for business communication, speech act theory

1. INTRODUCTION

By now, all habitués of the Internet are familiar with the process of joining a discussion group. It works roughly as follows. You hear about an

This work was funded in part under Contract DTCG39-86-C-80348, between the U.S. Coast Guard and the University of Pennsylvania, with S. O. Kimbrough as principal investigator.
Authors' address: S. O. Kimbrough, The Wharton School, University of Pennsylvania, Philadelphia, PA 19104-6366; email: kimbrough@wharton.upenn.edu; S. A. Moore, University of Michigan Business School, Ann Arbor, MI 48109-1234; email: samoore@umich.edu.

interesting network forum, and you send an inquiry, via email, to the group's information account, say, electronic-commerce-info@ . . . Back comes a message quickly. Your inquiry has not been read by any person or parsed by any computer program. Instead, a standard reply is sent automatically to all who correspond to the address, and your inquiry message is simply dropped. The standard reply normally is brief and contains essential information about how to join the group. Typically, you are instructed to resend a message containing one line of body: subscribe. When you do this, a process at electronic-commerce-info@ . . . notices that the first line in the body of your message consists of only the word subscribe. The program then parses the header of your email message and adds you to the subscriber list for the discussion group. Thus, your (second) message has been automatically processed, saving time and effort for all involved. Automated message handling makes the subscription process better (no one miskeys your address), faster (everything happens more or less immediately), and cheaper (once things are set up, no human intervention is required). Things being what they usually are, all this is to be welcomed.

The work we report on in this article is motivated by two observations. First, automated processing of messages, illustrated by the example of subscribing to an Internet discussion group, can often be enormously valuable. The point—about automated subscription processing being better, faster, and cheaper—generalizes richly in the contexts of electronic commerce and work support systems. Second, automated message handling relies essentially on the processing of structured messages, and the sophistication and richness of the subscription messaging scheme, just described, leave a lot to be desired. In a typical Internet discussion group system, only two messages are recognized: subscribe and unsubscribe. There have to be better, more powerful and general, ways to encode a formal message. The message creation and handling system here is terribly ad hoc (a topic treated in Kimbrough and Moore [1992]). Also, and speaking to the main subject of this article, there is a lot more that needs to be said in the conduct of business, even confining our attention to what needs to be said for automated message processing.

Of course, formal message encoding schemes of some sophistication are used daily, and their penetration is growing. Principal among these schemes are EDI (electronic data interchange) protocols, which we discuss in the sequel. It is our thesis, however, that a form of structured messaging—much richer than is typically encountered in current systems—would be desirable, practicable, and useful in many commercial and work support contexts. The principal aims of this article are to show why this is plausible, to show how it may be done, and to demonstrate (a degree of) feasibility. More concretely, our main points in support and elaboration of the thesis are as follows (we repeat our two observations as the first two points):

(1) the need for automated message handling point,
(2) the need to say more and say it felicitously point,

(3) the four approaches point,
(4) the discernment, iteration, and composability points,
(5) the need for theoretical soundness point,
(6) the aptness of speech act theory point,
(7) the limitations of speech act theory point, and
(8) the practicability point.

We now devote a brief subsection to explaining each of these main points. Of course, a full exploration of any of these points is beyond the scope of any single article. Our goal in *this* article is to present a prima facie case for the main points and then to focus on (a) the foundational significance of the speech act theory point and (b) the practicability point.

1.1 The Need for Automated Message Handling

This is our first observation, above. We think the point is a pretty obvious one. In addition, there are at least three sorts of evidence for it. First, practice confirms it. EDI, and other forms of electronic commerce based on automated processing of structured messages, are growing and are gaining a great deal of favorable attention. The market has spoken in favor of the general point. There are about 70,000 businesses worldwide that are using some form of EDI [Steel 1994]. SWIFT (Society for Worldwide Interbank Financial Telecommunications) alone, for example, switched about 2.8 million EDI messages per day during 1996.

Second, in this article we present several examples of formal messaging. Our purpose in these examples is mainly to illustrate other substantive points in the article, but we submit that the examples are instances of useful applications of structured messaging.

Third, a large number of observers (e.g., Gerlach [1992], Harrison [1994], Powell [1990], and Reich [1991]) have noted a strong secular trend for businesses to disintegrate vertically and to establish working, flexible (often temporary) relationships with many other firms, serving at many points in the value chain. The phenomenon is associated with the Japanese *keiretsu* system and currently goes by many different names, including concentration without centralization [Harrison 1994], alliance capitalism [Gerlach 1992], network forms of organization [Powell 1990], and operational webs [Reich 1992]. The following passage is representative of what these observers see [Harrison 1994, p. 127]:

> Of all the reactions [to "the trauma of the worldwide economic crisis of the 1970s and early 1980s"], all the experiments, the most far-reaching may well turn out to be the creation by managers of boundary-spanning networks of firms, linking together big and small companies operating in different industries, regions, and even countries. *This* development—not an explosion of individual entrepreneurship or a proliferation of geographically concentrated industrial districts, per se—is the signal economic experience of our era.

If, indeed, the imperatives favoring alliance capitalism are as powerful as suggested in this literature (see Harrison [1994, p. 166] for a list of "motives for technology-oriented companies to seek cooperation via networks"), then it is surely easy to see why there should be a strong, ongoing need for better systems to handle business messaging automatically.

1.2 The Need to Say More and Say It Felicitously

This is our second observation. It is surely obvious that the expressive power inherent in the discussion group subscription system is far from adequate for general commercial purposes. The question is, What is? In particular, are EDI protocols and other extant systems sufficient? We think not. We are not alone in this view (and see the detailed treatment in Kimbrough and Moore [1993]). The received view, typified in the following passage, is that there is a strong need to expand the scope of present-day electronic commerce activities [Foster 1994, p. 29]:

> Generically, it is often useful to view a manufacturing enterprise in terms of five basic processes:
>
> - Develop the Product
> - Sell the Product
> - Make and Deliver the Product
> - Collect and Disburse Funds
> - Support the Product
>
> Most EC [electronic commerce] focus to date has been on what we have termed the make/deliver process, which involves ordering processing, procurement, manufacturing, and logistics—everything necessary to transform an order into a delivered product. But emerging technologies are broadening EC's scope of application to include the other processes . . .—while increasing even further its value for the make/deliver process. At the present time, there are major EC opportunities in every one of the key business process arenas.

And with expanded scope for electronic commerce inevitably come demands to be able to express and—especially—to interpret a broader range of meanings. We shall call this the requirement for *expressive felicity*. Not only must messaging systems for work support and electronic commerce be able to express what needs to be said, but they need to facilitate (rather than to hinder) machine-based interpretation of messages and automated extraction of information from archived collections of messages.

Our approach to supporting this point will be indirect. The subsequent main points, beginning with our discussion of the four general ways of creating automated message handling systems (see Sections 1.3 and 2), can all be taken to lend credence to the general claim that greater expressive felicity is needed. Furthermore, after illustrating our approach in Sections 4–6, we directly address the felicity point in Section 7.

1.3 The Four Approaches

There are four general approaches to automated message handling in electronic commerce, which are now in general use or under general discussion. These are natural language processing, EDI (Section 2.1), tagged-message systems (Section 2.2), and FLBC (formal language for business communication, Section 2.3). In Section 2 we briefly describe and discuss each of these generic alternatives, with the exception of natural language, which we do not view as currently practicable (but see Young and Hayes [1985] and Pollack [1989]).

We believe there is a strong in-principle case to be made for an FLBC approach. The FLBC approach offers (among other things) greater flexibility and superior expressive felicity compared to the other three approaches; consequently, it will often be the preferred approach in applications (at least when speed and resource limitations are not dominant). This should especially be the case once sufficient infrastructure, that is, conventions and software for creating, reading, and generally processing messages, is in place. It is this eventuality to which the research we describe here aims to contribute.

1.4 Discernment, Iteration, and Composability

A fundamental reason for favoring an FLBC approach (to design messages for an automated message handling regime) is to obtain greater expressive felicity. What exactly does this mean? Here, we offer a partial characterization, based on three criteria. First, a messaging regime should be able to discern, or express, a rich variety of messages. Clearly, lack of discernment is apparent in the discussion group subscription system, since it really only recognizes three messages: (a) subscribe, (b) unsubscribe, and (c) neither (a) nor (b).

Second, there is a practical need for messaging systems to express and exploit iterated message operators. For example, existing message systems can, in effect, say such things as "Jones requests that Jones be put on the subscription list" and "Smith said that Smith is on the list." What they cannot say are such things as "Smith said that Jones requests that Smith be put on the subscription list," in which the message operators (". . . requests that . . ." and ". . . said that . . .") are explicitly (and decomposably) iterated.[1] In the discussion of our prototype language and implementation, Sections 4–7, we give examples of situations in which the need for iterated message operators naturally arises.

In the interests of brevity, we leave the discussion of composability, our third criterion, to Sections 3 and 7, in which the other two criteria are also presented.

1.5 The Need for Theoretical Soundness

As noted earlier, the message creation and handling system for discussion group subscription is terribly ad hoc and unsystematic. Similar complaints have been lodged, with some justice, against existing systems for automated message handling, particularly EDI systems. Here is a representative comment [Covington 1996] (see also Lehmann [1996]):

[1]Of course, any system can code iterated operators atomically simply by numbering them. As we shall demonstrate in Section 7, this is pretty much what X12 and other EDI standards do, to unhappy effect.

> A striking characteristic of X.12 and EDIFACT is their bloated ontology. When the same entity or type of entity turns up in more than one place, the sameness is not recognized. To take an extreme case, EDIFACT has no concept of "number"—instead, there are 3-digit numeric fields in some places, 4-digit numeric fields in others, 10-digit numbers somewhere else, and so on. The problem, of course, is that EDIFACT does not distinguish concepts from their physical representations. In essence, EDIFACT is a language for depositing character strings into particular places on a remote computer, rather than a language for exchanging knowledge. X.12 is largely the same.

Considerable benefits, especially generality and robustness under change, can be expected from a theoretically sound approach, were one to be found. We elaborate on these points in Sections 3 and 7.

1.6 The Aptness of Speech Act Theory

Our main goal in Section 3 is to argue for the in-principle appropriateness of speech act theory (SAT). Our claim is that SAT should be accepted as the foundational theory for representation schemes for automated message handling. We have the following main reasons: SAT in some version or other is widely accepted in linguistics, philosophy, and information systems; SAT tells us a great deal (or at least something essential and important) about the logic of what can be said (thereby offering generality and robustness, as with any good theory); and there really is not any close competitor to SAT for present purposes.

1.7 The Limitations of Speech Act Theory

SAT is foundational for present purposes in at least two ways. First, as mentioned in Section 1.6, SAT is a (nearly *the*) fundamental theory for linguistic communication. Vanderveken's [1990, p. 5] comment is, if anything, an understatement:

> In the past few decades, speech act theory and formal semantics have influenced the development of several disciplines, including not only philosophy, linguistics, and cognitive psychology, but also logic, artificial intelligence, law, business, translation, education, literary studies, and engineering. Moreover, speech act theory has also become a focal point of creative theoretical interactions in interdisciplinary research centers of cognitive science.

If we are to develop general message handling systems, it would be wise to attend to SAT. Second, and more relevant to the current point, SAT tells us something about the logical structure of messages. It tells us something, but hardly everything. A great deal is left open. We elaborate on this point in Section 3. The subsequent sections on our prototype implementation (Sections 4–6) illustrate how SAT may be augmented for practical purposes.

1.8 Practicability

The theory is nice, but will it work? The bulk of this article, especially Sections 4–6, describes a practical application, and prototype implementation, of these ideas. Extending previous work [Kimbrough and Thornburg 1989], we develop a general system, specialized for automatic processing of messages in an Army office environment. Following this, we demonstrate in Section 7 how the same language, augmented by an enlarged lexicon, can be used to express more felicitously standard EDI messages.

2. THREE KINDS OF APPROACHES TO FORMALIZED MESSAGING

Our purpose in this section is to present and discuss the three main approaches extant for formalizing, and computerizing, business communications.

2.1 EDI: Electronic Data Interchange

EDI protocols were, and are being, developed for the purpose of replacing the interfirm (and intrafirm) flow of standard paper documents, such as purchase orders and bills of lading, with computer-to-computer exchange of information (see Adam and Yesha [1996], Bleakley [1994], Edwards [1987], Emmelhainz [1993], Frost and Sullivan [1988], Kimberley [1991], Raymond and Bergeron [1996], Salminen [1995], Sivori [1996], Sokol [1995], Straub and Wetherbe [1989] for general information on EDI in practice). Such protocols are quite commonly used in the grocery, automotive, warehousing, transportation, distribution, and general manufacturing industries, and the use of these protocols is growing.

There are at least five major EDI protocol standards, but nationally there is a general movement toward a common EDI standard, called X12, which is under development by ANSI (American National Standards Institute: ANSI Accredited Standards Committee X12, Alexandria, Va.).[2] In the case of X12, and all the other existing EDI standards, various paper documents, for example, purchase orders and invoices, are identified as transaction sets, and carefully structured definitions are developed for the sake of representing them electronically. Once the standards are in place, organizations write software for creating and interpreting documents conforming to the standards.

Although EDI systems have been extensively and successfully implemented, and are growing in popularity, it is clear that (1) the protocol orientation of EDI[3] continues to be a hindrance to further use because of inflexibility; and (2) the document, as opposed to message, orientation of EDI protocols also hinders flexibility and expressive felicity.[4] While EDI is

---

[2]Internationally, there is a general movement toward the UN/EDIFACT standards. X12 is part of that general movement.

[3]By which we mean that EDI messages, under current standards, consist of logically very simple, elementary structures.

[4]EDI protocols, a.k.a. transaction sets, are typically conceived as more or less direct replacements of existing paper documents used in commerce, e.g., bills of lading, receiving reports, and invoices.

a good and growing thing, other technical approaches may yield greater functionality.[5]

2.2 Tagged Messages

There is an intriguing, somewhat dispersed literature focusing on computer-mediated communications in which messages are *tagged* (the term is ours) in some way and the tags are used for various purposes. Much of this work is oriented toward developing intelligence-based electronic mail systems [Case 1982; Chang and Leung 1987; Comer and Peterson 1985; 1986; Hiltz and Turoff 1985; Malone et al. 1987a; 1987b]. A general complaint with existing electronic mail systems has been that they foster "information overload" by inundating the subscriber with "junk mail." By tagging messages and giving subscribers procedures for processing the tags, one could hope that the resulting system would help subscribers to "filter, sort, and prioritize messages that are already addressed to them, and . . . [help] them find useful messages they would not otherwise have received" [Malone et al. 1987a]. The state of the art here is that a number of prototype systems have been built, installed, and studied (with generally quite positive results), but the widely used electronic mail (and, more inclusively, electronic mail, computer conferencing, and electronic bulletin board) systems do not make significant use of information about messages; what use is made of such information is limited to data stored in the message header and is normally not available to a user's procedures.[6] What semantic access is available in these systems is available through the message tags; the contents of the messages are not semantically accessible. Furthermore, the message tags are not defined in a recursive fashion, as in a full-fledged language, so that with each additional meaning indicator, or tag, a new symbol must be defined. Under a linguistic regime for expressing semantic content, however, an infinite number of meaningful sentences are implicitly defined by the rules of formation and interpretation (see, e.g., Section 5).

A second area, outside electronic mail systems, in which the tagged-message idea has been explored may loosely be described as office, or work, support. There has been some (indirect) speculation in the literature of group decision support system research that message properties need to be captured and processed (e.g., DeSanctis and Gallupe [1987]). Others (e.g., Flores et al. [1988], Kaye and Karam [1987], Mantelman [1987], Winograd and Flores [1987], and Winograd [1988]) have designed and developed prototype and commercial office support systems that can direct and

[5]The critique we make here is broadly shared among practitioners. In the EDI industry, there is an oft-repeated lament that "the nice thing about EDI standards is that there are so many to choose from." For a more detailed and technical critique, see Adam and Yesha [1996], Kimbrough and Moore [1993], Moore [1993], and Moore and Kimbrough [1995].

[6]It is instructive in this regard to read technical manuals for popular email systems, e.g., Lotus's ccMail.

coordinate the functioning of multiple, distributed processes in support of a given office task.

2.3 FLBC: Formal Language for Business Communication

Finally, there is a small but growing literature aimed at developing what we call a formal language for business communication (e.g., Covington [1997], Finen et al. [1993; 1994], Kimbrough [1990a], Kimbrough and Lee [1986], Kimbrough and Moore [1992; 1993], Kimbrough and Thornburg [1989], Labrou and Finin [1994], Lee [1980; 1984; 1988a; 1988b], Lee and Bose [1988], Lee and Ryu [1989], Lee and Widmeyer [1986], Mayfield et al. [1995; 1996], McCarthy [1982], Moore [1993], Moore and Kimbrough [1995], and van Reijswoud [1996]). The differences between a tagged-message system, an EDI system, and an FLBC system may be described as follows. Typically, in a tagged-message system, a message consists of two elements: the message header and the message body (cf. Chang and Leung [1987]). The message body may be processed only in the most rudimentary ways; it may be displayed, copied, and forwarded, but cannot be used for inferencing. The message header contains, in our terminology, a series of tags, normally including such information items as the message type, a unique message identifier, and various associated key words that serve as message descriptors. The elements in the header, the tags, are available for processing by inferencing procedures. We can think of the EDI approach as a tagged message in which most of the information has been moved out of the body and into the header. In an FLBC system, a message consists of a series of assertions, or declarations, each of which is, typically, a possible input to an inferencing procedure. We can think of such a system as an EDI system that replaces the header (expressed as a data structure) with a series of individually meaningful and arbitrarily orderable declarations, or statements.

The state of the art for FLBC systems is best described as being in the exploratory phase. This article represents an effort to explore the idea somewhat further and to do so by tying the effort to develop an FLBC to a solid theoretical base. We now turn to a short discussion of our theoretical outlook.

3. THEORY: SPEECH ACTS AND REPRESENTATIONS

Recent work in linguistics and philosophy of language, aimed at developing theories of how language understanding and communication work, has emphasized the role of inference and context (e.g., Bach and Harnish [1979], Cohen et al. [1990], Levinson [1983], Searle [1969; 1979], Searle and Vanderveken [1985], and Vanderveken [1990]). In concert with this work, and beginning, roughly, with the publication of Austin's [1962] *How to Do Things with Words*, a theory of—or theoretical approach to—linguistic communication has been under more or less continual development by linguists, philosophers, psychologists, and cognitive scientists generally. (Of course, there is precursor work, particularly Strawson [1950].) This theoretical approach is called *speech act theory* (*SAT*), in part because its adherents take as a starting point for their theorizing about linguistic communication the fact that to say something is, among other things, to take an action.[7] There is no generally accepted full description of the theory, since different authors tend to emphasize the details of their differences with other writers on speech acts (but see Levinson [1983] for a review of the literature). There are, however, certain core ideas broadly accepted by speech act theorists, and it is these core ideas that prove most useful for beginning to develop a formal language for business communication. They are

(1) the act decomposition of speech acts (see Section 3.1),
(2) the *F*(*P*) framework (see Section 3.2), and
(3) the *F* framework (see Section 3.3).

We shall now discuss them individually.

3.1 The Act Decomposition of Speech Acts

The first core idea of SAT is that every speech act may be understood as consisting of several distinct actions. The idea and most of the terminology originate with Austin [1962], although both have been developed in an extensive subsequent literature. Recognizing that different authors distinguish somewhat differently among the various constituent acts and even recognize different acts, for present purposes we may understand a speech act as representable by four distinct actions. Suppose that a speaker, *s*, succeeds in saying something to a hearer, *h*, in a given context, *c*. We may then distinguish the following acts:

—*utterance act*: the uttering of *u* by *s* to *h* in *c* of a particular expression from a given language;
—*locutionary act*: the actual saying of something by *s* to *h* in *c*;
—*illocutionary act*: the doing of something by *s* in *c*, in virtue of having performed the utterance act; and
—*perlocutionary act*: *h*'s being affected by *s* in *c*, in virtue of *s*'s utterance act.

The general picture of communication and understanding that emerges is that a linguistic communication—a successful *speech act*—between a speaker, *s*, and a hearer, *h*, may be viewed as a sequence of four steps, which (after Bach and Harnish [1979]) we shall call the *speech act scenario*. It begins with a *speech event* [Levinson 1983] or *utterance act* [Bach and Harnish 1979], consisting of an *utterance*, *u*, and a *context*, *c*. The utterance, *u*, may be many things, including a sentence from a given language

[7]The term *action* is, of course, being used in a technical and theory-laden, if not altogether clear, way. Briefly, *to act* is more than to do something; it is to do something with an appropriate attendant intention. Falling down is usually not an action, whereas pulling a lever in a polling booth normally is.

(e.g., English), a sentence fragment (e.g., "She's in the (pointing to the living room)"), or a sign designating a sentence (e.g., nodding assent, giving a "thumbs down" to reject an offer to sell stock). The context, *c*, may include

(1) certain conventions and assumptions (e.g., that English is the primary language presently in play; that this is the serious business of buying and selling equities and not, e.g., a game of charades);
(2) certain gestures and inflections of speech (e.g., pointing and emphasis);
(3) relevant history pertaining to a conversation (e.g., to fix the reference of a pronoun); and
(4) relevant ambient facts (e.g., "I'll see you in an hour" means the speaker will see the hearer at 3:00 PM, given that it is now, at the time of the utterance act, 2:00 PM).

Just what, in a given situation, should be included in the relevant context is a problem for which there is presently no broadly satisfactory answer. We have proceeded workman-like, putting into the context whatever we need to perform the job at hand. What we found we needed, for the application described below, was, occasionally, the history of the conversation as given by the IDs of the messages in the conversation.

The second stage of the speech act scenario is called the *locutionary act*. Our hearer, *h*, has heard the utterance act, that is, has heard *s* utter *u* in *c*. Now *h* has the problem of figuring out, inferring, what the utterance means. If, for example, *h* has just asked *s* if she will be home Tuesday night and *s* has responded with a nod (the utterance act in question), then *h* might infer that the content of *s*'s utterance is that she will be home on Tuesday night. Let *P* be this inferred (propositional) content of *s*'s utterance act. If *P* is what *s* intended her utterance to mean, then we say that the locutionary act aspect of *s*'s speech act (begun with *s*'s utterance act) has succeeded. (Notice that *P* is abstract. The utterance act is (a sentence) in a particular language, while (the proposition) *P* is what is said. For example, "Il pleut" and "Es regnet" are two different utterance acts having, as it were, a common locutionary act, that it is raining. Recall the exchange from the movie, "Shall We Dance." "What does that mean in English? The same thing it means in French.")

The third stage of the speech act scenario is called the *illocutionary act*. Our hearer has heard, or observed, the utterance act and has successfully interpreted it: *s*'s utterance means that she will be at home Tuesday night. But what is *s* really saying? Is *s* predicting that she will be home Tuesday night, or is she promising it? There is a difference, and the difference is important. If *h* succeeds in correctly inferring the attitude (promising, predicting, lamenting, etc.) toward *P* that *s* intended to communicate, then we say that the illocutionary act aspect of *s*'s speech act has succeeded. Following Searle [1979; 1985], let *F*, for *illocutionary force*, be this inferred attitude toward the content, *P*, and let what *s* has said be represented as *F*(*P*), an illocutionary force, *F*, applied to a content, to a true-or-false proposition, *P*.

Finally, the *perlocutionary act* aspect of the speech act includes the effects that *s*'s utterance act has on *h*. For example, if the illocutionary act is a promise (to be home on Tuesday night), then *h* might come to rely on the promise and consequently cancel a previous commitment in order to accommodate *s*'s visit.

In terms of the speech act scenario, our focus is on developing a formal language for utterance acts, one that is sufficiently rich and explicit that it can readily express the illocutionary acts that are needed in the conduct of commerce.[8]

3.2 The *F*(*P*) framework

The second core idea of SAT is the notion that every (or nearly every) illocutionary act involves an expression by the speaker of a propositional attitude toward some (possibly complex) proposition. For example, if the speaker says "It will rain," then typically the speaker is asserting (and predicting) that it will rain. Here, then, the proposition is that it will rain, and the propositional attitude is that of an assertion. On the other hand, if the speaker says, "Will it rain?" then typically the speaker is asking whether it will rain. In this case, the proposition is the same—that it will rain—and the propositional attitude expressed is that of a question. In both cases, the underlying proposition is the same, but the propositional attitude is different. In the first case, the attitude is an assertion, and in the second case, it is a question. Because propositional attitudes arise in other contexts (particularly in psychological explanation, e.g., believe, intend, desire), those associated with speech acts have been given a special name. They are called *illocutionary forces*. This second core idea is summarized by saying that every illocutionary act may be analyzed formally as having the structure, $F(P)$, where F is an illocutionary force applied to a proposition, P, called the *propositional content* of the act. Thus, this second core idea may be called the $F(P)$ *framework*.

There is something remarkable, and quite powerful, about the generality of the $F(P)$ framework. First, the $F(P)$ structure is amenable to iteration (recall Section 1.4). Thus, an assertion that a request has been made has the general form $F(F(P))$, with the outermost F standing for assertion, the inner F standing for request, and P standing for what (it is asserted) was requested. And there is no syntactic or conceptual limit on how deeply such nesting can be taken. The payoff, in terms of discernment (again, Section 1.4), is rich and elegant: proposing to request and requesting to propose, for example, are quite distinct and quite easily handled under the $F(P)$ framework.

---

[8]In the interests of brevity, much is being elided. We incline toward inferential theories of communication (e.g., Bach and Harnish [1979]), rather than decoding theories (e.g., Searle [1979; 1985]). For a discussion of the difference between decoding and inference, see Bhargava and Kimbrough [1995] and Sperber and Wilson [1988]. For its relevance to electronic commerce, see Kimbrough [1990b] and Moore [1993], especially in regard to the distinction between sentence meaning and speaker's meaning.

Second, $F(P)$ units may be combined, using a limited set of illocutionary connectives (cf. Searle and Vanderveken [1985, p. 3]). In this way, for example, a speaker's making an assertion *and* asking a question may be captured formally. Again, the payoff in terms of discernment, or expressive felicity, is substantial.

Third, the $F(P)$ framework is the principal vehicle for making the theoretical claims of SAT operational. The claim of universality for SAT mainly amounts to the claim that all that can be said (ever, by anyone—and certainly including routine business transactions) can be expressed within the $F(P)$ framework, that is, as iterations and combinations of $F(P)$ structures. Our research program does *not* assume that any such sweeping claim is true. Rather, we see it as a serious hypothesis, one that is very much alive in light of current evidence, and one that is ripe for testing with applied research. Note, moreover, that the universality hypothesis is exactly what makes SAT so interesting for electronic commerce and work support systems. If the hypothesis is correct (or nearly so), then we have in outline the underlying logical structure of everything that can be said, and certainly including messages for conducting business. This, if true, is exactly what one would hope for as a theoretical basis for robust applications. And, because the underlying logical structure is formal, translation can be made for process-to-process messaging.

Fourth, and finally, a point on composability (recall Section 1.4), which roughly amounts to analyzability by known rules. Process-to-process messages must be composable in this sense, and a main requirement of any formal language is that all of its expressions are composable. Iteration makes for composability, and so does combination using a limited set of connections (the first and second points above). All of this fosters discernment or expressive felicity.

3.3 The *F* Framework

With only a little—the $F(P)$ framework and SAT's claims of universality—we have gained a lot. We would like more. We would like for both F and P similarly to have a universal structure. This is not, and cannot be, the case for P. Although a regular grammar is possible (viz., first-order logic), different domains will have different vocabularies, and the total lexicon, across all applications, is likely to be huge. Nevertheless, for restricted applications (e.g., for many aspects of electronic commerce and work support systems) it should be possible to devise expressively felicitous systems using a limited vocabulary and various logical structures, including those of SAT. But this is not our main concern here (see Kimbrough and Moore [1993] for a treatment of these other issues).

According to SAT, the news is good with regard to F. The story is complicated (see Searle and Vanderveken [1985] and Vanderveken [1990] for formal treatments). In essence, however, there are an infinite number of illocutionary forces (the Fs), but these fall into a small number of types, which Searle calls the illocutionary points. In his view there are five:

(1) *the assertive point*: used to say how the world is; used to make statements;
(2) *the commissive point*: used to commit the speaker to an action; used to make promises;
(3) *the directive point*: used to commit the hearer to an action; used to give orders;
(4) *the declarative point*: used to make changes in virtue of speaking; here, "saying so makes it so," as in an umpire crying "You're out!" and
(5) *the expressive point*: used to express the speaker's attitude, as in "Oh, to be in England" and "Yea!" and "Boo!" and "We've got to work together to build a bridge to the twenty-first century."

An illocutionary force, an F, can be thought of as an illocutionary point plus various qualifications. For example, a prediction (an F) is an assertion about the future; a vow (another F) is a solemn promise.

Other frameworks for the basic illocutionary points are possible and have appeared in the literature (e.g., Bach and Harnish [1979]). Applied research on significant problems will be needed to resolve such differences. What is significant for present purposes is that the illocutionary points are

(1) small in number (this is from SAT);
(2) complete (also from SAT, although theorists differ as to the exact list); and
(3) useful as approximations.

By this last point, we mean that in many situations the complete articulation of the illocutionary force is unnecessary; the illocutionary point itself (perhaps slightly qualified) may often be used to express and interpret the intended message. SAT is silent on this point. It remains for applied research and practical applications to pass on its correctness. Our experience in this regard, some of which is reported below, is entirely sanguine.

3.4 Discussion

Our applying SAT as a foundation for designing information systems is, by itself, neither unique nor original. Flores and Winograd (e.g., Flores et al. [1988], Mantelman [1987], Winograd [1988], and Winograd and Flores [1987]) have built and described systems that employ speech act concepts. However, as noted by Blair [1992] in a favorable discussion of the use of SAT for information retrieval, the efforts of Flores and Winograd in this regard are not nearly as ambitious as what we are reporting in this article. The systems of Flores and Winograd do not use a full-fledged formal language for business communication. Instead, they have implemented a tagged messaging system, using simple elements of the F framework. Similar comments apply to other efforts to employ SAT in organizations (e.g., Campbell [1990], De [1986], Lehtinen and Lyytinen [1986], Lyytinen [1987; 1985], Smeltzer [1991], and Starling [1993]).

Our work focuses elsewhere. We are exploring the systematic use of a full-fledged FLBC (see McCarthy [1982] for an early vetting of this idea). For reasons indicated earlier, we believe that SAT and, in particular, its $F(P)$ framework are the appropriate starting points for the development of such an FLBC. In what follows, we shall focus on developing representations for utterance acts (for an FLBC), such that the inferences needed to produce the locutionary and illocutionary acts, and to reason with the results, are as correct and transparent as possible.

Having presented the pertinent essentials of SAT (see Kimbrough and Lee [1986] and Moore [1993] for additional information), we shall now discuss a specific application area, an Army office environment, for application of an FLBC.

4. ARMY OFFICE COMMUNICATION

4.1 Introduction to the Area

We chose an Army office environment to test the application of the FLBC approach because we are familiar with it and because the clear lines of authority in an Army office present opportunities for computerized inferencing on messages.

In an Army office, paths of command and responsibility can easily be delineated. Within such an office, each dialog carries with it information on its own implied force, based on the rank and relationship of the individuals involved. While rank may not be the sole guide of who works for whom, a combination of rank and job position reflects the lines of communication used within the office. Furthermore, the rigidity of the military chain of command clearly reinforces the comprehension of how illocutionary force is applied to various message types. For example, when a military commander issues a directive for an appointment with a subordinate, virtually all military personnel construe that request as an order, rather than as a suggestion, a polite request, or an invitation. While the perception of an analogous situation in the civilian world between a supervisor and subordinate may be similar, the exact underlying force of the message may not be as obvious and is likely more variegated.

4.2 Message Types

We have identified seven general message types for the Army office context.[9] The names we have selected for these seven are not formal names adhered to by the official military community. Instead, in the day-to-day functioning of many military staff officers, the names reflect what a staff officer might use as a subject heading on a written memorandum to a

[9]The main source for our information was Major Michael J. Thornburg, U.S. Army. We conducted several lengthy interviews with him. Between interviews, he consulted other Army officers with relevant experience. In the end, he endorsed the resulting list of message types (see Kimbrough and Thornburg [1989]). This list, however, should be seen as illustrative, rather than as definitive.

commander, coworker, or subordinate. The seven message types are, we believe, capable of facilitating a broad spectrum of communication between military personnel. They are as follows:

(1) read/review/comment,
(2) appointment,
(3) dissemination of information,
(4) staff action,
(5) query for information,
(6) absence, and
(7) statement.

We shall now briefly discuss each of these seven message types. We give further analysis, specific to our FLBC and our prototype implementation, in Section 5.

4.2.1 *Read/Review/Comment.* Much of an officer's day is taken up with reading documents or with writing critiques of, or comments on, documents. Read/review/comment (RRC) provides the speaker with the capability to distribute documents and messages to people, and to assign one or more people to read, to act on as appropriate, and possibly to critique a document. The message type conveys the force of a directive and the speaker may optionally require a response and set a date and time when some specified action is to be completed. Furthermore, recipients of an RRC message may be required to send an acknowledgment when the material is read. Records of each acknowledgment may be maintained by the sender (speaker), indicating the personnel who have read and complied with the message. This type of message is used extensively by military organizations and government agencies in distributing requirements set by military regulations, Federal guidelines, and Privacy Act requirements.

4.2.2 *Appointment.* In all professional environments, the ability to manage appointments is required to schedule events, ranging from major meetings to minor social gatherings. Just as in any face-to-face encounter, a request for an appointment requires the hearer to respond to the speaker's request. How elaborate the response is, especially a negative response, depends on the relationship between the speaker and the hearer. If the speaker is the commander, a simple "no" will not be sufficient. Instead, an explanation would probably be required. The explanation may also contain a question. If a colonel asks to see a major at 2 PM on Thursday, the major may reply negatively, explaining that he will be in a meeting with another officer at that time. Depending on the circumstances, the major may wish to include a question, for example, "Do you want me to change my meeting with Major Amos?"

4.2.3 *Dissemination of Information.* Every office has a bulletin board with notices whose posters may be of interest to various readers. Furthermore, every office circulates, for example, with routing slips, documents that may be of interest to, or were requested with a standing order by, their recipients. From the sender's point of view, this is a "send and forget" message. It is the responsibility of the hearer to read and act—or not—on the message.

4.2.4 *Staff Action.* One of the main workhorses of this system is the staff action message type. In a staff action, one person might be assigned to attend a meeting, or an entire office might be directed to work on a high-priority project. Normally, one or more responses by the hearer are required. Often, the required response comprehends the requested action. For example, if a report is to be written and delivered by a particular time to a particular officer, then the required response includes the report. When the speaker desires additional responses, such as message receipt confirmation, capability of meeting project due date, and acknowledgment of intermediate due dates (milestones), then these must be explicitly requested by the speaker.

4.2.5 *Query for Information.* A query for information is, from the point of view of SAT, closely related to a staff action. In the Army office context, the difference between a query and a staff action is genuine, but one of degree. The information requested in a query is expected to exist already, and the effort to collect the information is thought to be minor. A staff action would be used to produce, or substantially process, the information, whereas a query is intended to result in a relatively easy retrieval of information.

4.2.6 *Absence.* The absence message type allows speakers to give notification of planned and authorized future absences. When such an announcement is appropriately made, office procedures may be more or less automatically altered in order to maintain office functionality at a high level. Through checking announced absences, supervisors may know where their people are, messages can be rerouted to alternate personnel who are not absent, and scheduling meetings may be made simplified by looking ahead at the availability of various participants.

4.2.7 *Statement.* Similar to dissemination of information, the statement message type is used to convey information. While a dissemination of information carries with it only the implication that the speaker thinks the content might be of interest to the hearer, a statement message is an assertion by the speaker, to the hearer, that the content of the message is in fact true.

Given this general description of the seven message types and their uses in existing (not automated) Army office contexts, we proceed to an implementation-directed analysis, in light of the theory discussed in Section 3. In doing so, however, the reader should keep in mind that the context at hand is an Army office of intelligence analysts and that the aim of the study and implementation was to provide *some* off-loading of verbal and paper-based communication costs, rather than anything approaching a substantial elimination of managerial tasks.

5. LANGUAGE FOR OFFICE MESSAGES

We now consider how to represent the seven message types, discussed in Section 4.2, in an FLBC. Although we shall develop a particular language, we hypothesize that the family of languages to which it belongs, FLBC-2 (see below), is in fact quite general and can be applied in many contexts besides the particular application we are presently reporting on. In fact, we begin just such an application for X12 messages in Section 7. Our hypothesis, while not fully *tested* here, is *testable*, and its fortunes are significantly relevant to SAT. If our language, FLBC-2, or something much like it, can be made to work and work well in a variety of application domains, then SAT is corroborated. Conversely, if this language is radically inadequate, then SAT may be undermined. Ultimately, much is at stake.

5.1 Basic Structure of the FLBC

Our general strategy in representing a speech act is to identify the

—speaker,
—hearer,
—illocutionary force (or attitude),
—content, and
—context.

The form of an FLBC message can be summarized with the definition shown in Figure 1, which defines a family of languages. Figure 2 provides the syntax definitions needed to interpret Figure 1.

The following points arise with respect to Figure 1:

—As shown in item 1, messages sent between applications or within applications are either a single message (a ⟨msg-st⟩) or a list of messages (an ⟨oration⟩). Item 2 simply provides a means to send several messages at once; the interpretation of the message list ("[" [⟨msg-st⟩ {"," ⟨msg-st⟩}]" "]") is msg1 and msg2 and . . . .

—The basic message (⟨msg-st⟩) is defined in item 3. The ⟨msg-id⟩ uniquely identifies this message.

—Item 4 provides much of the power of this language. Notice that the second option is ⟨msg-st⟩. Combining items 1 and 4, it can be seen that one message can contain another message (which can contain another message . . .). The significance of this is discussed in a later section.

—The simpleUtterance in item 4 provides a means for specifying a message without specifying either its context or a message identifier. This is used for embedded messages such as A said B said X; in this message, B said X can be expressed in a simpleUtterance.

—A message can be sent from many speakers to many hearers, but one message token can only be sent to one hearer. The context predicate alsoSentTo in item 7 lists those people to whom message tokens, identical

```
 1. <message> ::== <msg-st> | <oration>
 2. <oration> ::== "oration(" <speaker> "," <hearer> ","
          "[" [ <msg-st> {"," <msg-st>}]" "] "," <oration-id> ")"
 3. <msg-st> ::== "msg(" <speaker> "," <hearer> ","
          <illoc-attitude> "," <content> ", [" [<context>
          {"," <context>}] "] ," <msg-id> ")"
 4. <content> ::== <pred-st> | <msg-st> |
          "and([" [ <msg-st> {"," <msg-st>}] "])" |
          "or([" [<content> {"," <content>}] "])" |
          "isNot(" <content> ")" |
          "iff(" <content> "," <content> ")" |
          "ifThen([" [ <msg-st> {"," <msg-st>}] ","
               [ <msg-st> "," <msg-st>] "])" |
          "simpleUtterance(" <speaker> "," <hearer> ","
                <illoc-attitude> "," <content> ")"
 5. <speaker> ::== "[" [ <person-id> {"," <person-id>}] "]"
 6. <hearer> ::== <person-id>
 7. <context> ::== "respondingTo(" <msg-id> ")" |
          "timeSent(" <time> ")" | "sendingMachine(" <mach-id> ")" |
          "alsoSentTo([" [ <person-id> {"," <person-id>}] "])"
 8. <pred-st> ::== <predicate> ["(" <arg> {"," <arg>} ")"]
 9. <arg> ::== <obj-id> | <time-pred> | <pred-st>
10. <time> ::== "time(" <Y> "," <Mo> "," <Day-of-Mo> ","
          <H> "," <Mi> "," <S> ")"
    where each argument is an integer in the appropriate range.
```

Fig. 1. Basic definition of FLBC-2.

1. A ::== B — Term A is defined as B
2. <C> — the non-terminal symbol C
3. A | B — A or B
4. [A] — zero or one instances of A
5. {A} — zero or more instances of A
6. "xyz" — the terminal symbol xyz

Fig. 2. Syntax definitions.

in attitude and content, were sent at the same time as the current message token.

—Descriptions of the other context predicates are as follows:

—respondingTo specifies the message to which the current message responds,
—timeSent specifies the time the current message was sent, and
—sendingMachine specifies the computer from which the message originated.

To obtain a specific language from this definition, a vocabulary must be defined: objects, illocutionary attitudes, and predicates. For an example,

```
1. <date> ::== "date(" <Y> "," <Mo> "," <Day-of-Mo> ")"

2. <time-or-date> ::== <time> | <date>

3. <time-pred> ::== <time> | <date> |
       "before(" <time-or-date> ")" |
       "after(" <time-or-date> ")" | "at(" <time>")" |
       "on(" <date>")" | "between(" <time> "," <time>")" |
       "between(" <date> "," <date> ")"

4. <obj-id> ::== "person(" <id> ")" | "message(" <id> ")" |
       "oration(" <id>")" | "document(" <id> ")"

5. <illoc-attitude> ::== assert | request | query

6. <predicate> ::== absent | appointment | available |
       commentOnItem | complete | doable | forwardedFrom |
       forwardToPerson | hasItem | hasRank | implement |
       inPosition | interesting | outrankedBy | readItem |
       reason | reportsTo | reviewItem | send | urgent
```

Fig. 3. A basic vocabulary for FLBC-2.

consider the definition shown in Figure 3, which is specialized for an Army office context.

There is much more to the story of this language. The meaning of the predicates and the arguments needed for each are not given, but supplying them is straightforward. Very few illocutionary attitudes are defined, but others were not needed for the prototype application. Extending the list of ⟨illoc-attitude⟩ to include such attitudes as *order*, *suggest*, *accept request*, and *deny* is, again, straightforward.

We now examine the seven Army office message types explicitly.

5.2 Statement

A statement, in terms of SAT, is an assertion. In making a statement, the speaker is asserting that what he is stating (i.e., the propositional content of the statement) is true. The FLBC representation of a statement message type is the following:

Msg (1) msg(From, To, assert, Ctnt, Ctxt, ID)

The only requirement for this message type is that the illocutionary attitude be assert. Permitted propositional content is implementation-specific. In a particular implementation, a lexicon of predicates and terms is developed. Any expression that is logically well formed and composed of predicates and terms from the lexicon is a valid propositional content, here and for all other message types.

5.3 Absence

A speaker's announcement of impending absence is an assertion whose associated content is a predication of the absent predicate, absent(Person, Begin, End), with the intended translation "Person Person is absent from time Begin to time End." Thus, this message is represented by

```
Msg (2) msg(From, To, assert,
            absent(From, Begin, End),
            Ctxt, ID)
```

We may also use the reason predicate to state the reason for the absence.

5.4 Dissemination of Information

In disseminating an item of information, the speaker is asserting that the information in question is interesting to the hearer. Let interesting(H, I) belong to our FLBC lexicon with the intended interpretation that the information item named by I is interesting to person H. Then one form of a dissemination of information message is as follows:

```
Msg (3) msg(S,H,assert, interesting(H, I), C, ID)
```

We note that I may be complex. For example, in our implementation it may be a logical (Boolean) combination of several predicates.

5.5 Appointment

An appointment message type is a directive to the effect that the hearer have a meeting with certain specified individuals at some time and place in order to discuss a certain topic. When rank matters, as it does here and almost always elsewhere, it is important to qualify the strength of the directive (cf. Sivori [1996]). Thus, in FLBC-2-1, we use request(N) to indicate illocutionary force in our message, where n ranges from -5 (pleading, beseeching) to $+5$ (commanding, giving an ultimatum), and where 0 represents a polite request. (A simple request is given without an accompanying integer to indicate strength. Only such simple requests are presently supported in our implementation.) The content for an appointment message has the following form:

```
Msg (4) appointment(Spkr, Hr, From, To, Place)
```

In the event that the speaker requires an explanation in case the request is denied (see Section 4.2.2), the content is expressed as a conjunction, using and and the predicate reason.

5.6 Query for Information

There are different ways in which questions might be handled. The method we use deviates somewhat from the taxonomy of Searle and of some (but not all) others, in which a question is a kind of directive. Our method is simply to treat a query as its own illocutionary force and to place the knowledge of what to do in response to a question in the programs that use and process the FLBC messages. Full analysis and defense of this approach must wait for future work. In short, then, a query looks like a statement message, with query replacing assert. Yes-no questions are represented by applying the query force to a declarative statement. Who-or-what questions have special question terms embedded in the statement expressions.

5.7 Staff Action

We model a staff action message as a directive, with request as the illocutionary force indicator in FLBC-2. The key to successful automation of this message type is to develop a useful (concise, yet powerful) lexicon for representing the content of such messages. Our initial investigations lead us to believe that this can be done. A great many message contents have to do with project status reporting, task assignment, and alteration of task priorities. Full discussion of this matter is beyond the scope of the present article, but briefly, we have pursued the following strategy: We have aimed, whenever possible, to use basic, rather than derived, illocutionary attitudes. For example, appointment and staff action messages are both requests. We could have added both appointment and staff-action to the list of illocutionary attitudes in FLBC-2-1, but we chose not to do so. Instead, when the message sender indicates that a staff action message should be sent, the system (in our implementation) infers that the appropriate attitude is a request. Furthermore, there is an implicit parse tree for permitted staff action message statements. The system uses this parse tree in order to prompt the user for the information needed for the message, and to validate the message. On the receiving side, the system is able to make inferences that classify messages in various ways, for example, as requests that are staff actions or as messages that require certain immediate actions. Thus, there is substantial inferencing performed during both the formation of a message and its interpretation by the system.

5.8 Read/Review/Comment (RRC)

In terms of SAT, we model an RRC message as a directive. The speaker is directing the hearer to read a particular document, to review it (to act appropriately, depending on the content of the document), and to reply with comments on the document as appropriate. We distinguish two types of RRC messages. RRC-1 is used when a speaker desires some sort of response but does not specify any additional actions. For example, if a project officer sends a document to an assistant, either via office hard-copy distribution or through electronic mail, the officer may transmit a request that the assistant acknowledge the receipt of the document.

RRC-2 is used by a speaker when he wants both a response and some specified actions by the hearer. The actions may be specified explicitly by the speaker or may be contained within the document in question. For example, a new administrative requirement could be sent to the appropriate department responsible for implementing such requirements. Within the document is contained what to implement, how to implement it, and when to do so. A commander who transmits this message as an RRC may merely ask the hearer to reply whether or not the required implementation date can be met. A similar message may involve sending a document that only contains what to implement and when implementation is required to be complete. In addition to inquiring whether the implementation date can be met, the speaker may include in the message information on how to implement the new procedures, a request to prepare an additional briefing or report, and so forth.

To illustrate, suppose that Colonel Wahl sends an RRC-1 message to Major Lane to the effect that Lane is to read a particular document, to implement its directives by a given date, and to respond a week earlier whether the implementation can be effected. Specifically, let

—*speaker*: Wahl (i.e., Colonel G. Wahl).
—*hearer*: Lane (i.e., Major M. Lane).
—*context*: nil; no relevant context.
—*content*: read(x, y) (i.e., x reads document y).
—*content*: implement(x, y) (i.e., x implements applicable directives in y).
—*content*: time(before(t), x) (i.e., x is a time on or before time t).
—*content*: doable(S, t) (i.e., situation S can be brought about at time t).

Given this, Colonel Wahl's message is

Msg (5) msg(Wahl, Lane, request, Φ, [ ], msg4)

where Φ is as follows:

```
Msg (6) and(read(Lane doc37),
            time(before(date(1994, 3, 30)),
                implement(Lane, doc37)),
            time(before(date(1994, 3, 23)),
              simpleUtterance(Lane, Wahl, inform,
                doable(implement(Lane, doc37)))))
```

We note that, although the complete message is complex, it is formulated by inferencing and under program control, with only a slight burden placed on the message sender.

Having presented these rudiments of our FLBC and the theory behind it, we shall now discuss our prototype implementation.

6. IMPLEMENTATION AND INFERENCING

The principal benefit of the syntactic articulation of messages in a business communications context is that the messages become semantically accessible. By expressing the messages in a theoretically sound language, inferencing can be facilitated. In order to illustrate this concept, we have developed a prototype FLBC system written in Prolog. Our main purpose in this section is to sketch a description of the prototype with enough detail that the feasibility and usefulness of (correct) inferencing on messages in a

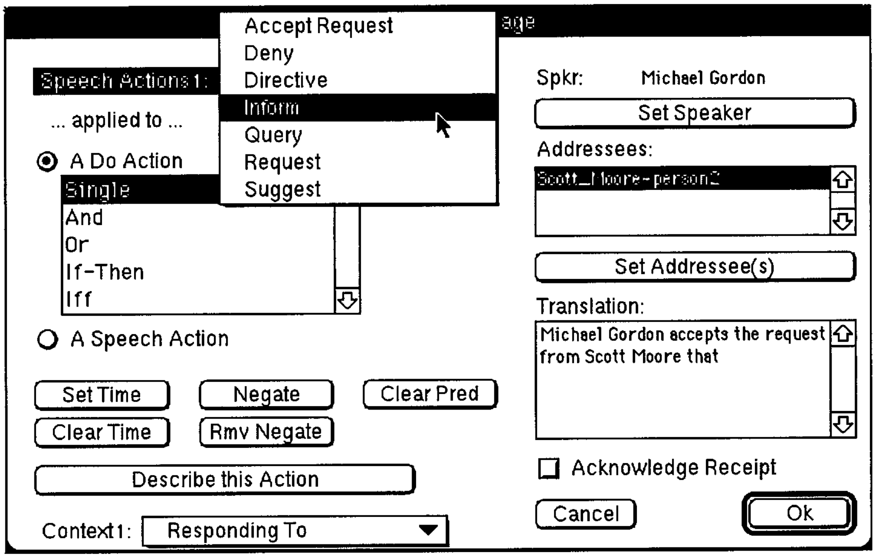

Fig. 4. General-purpose message construction dialog box.

business communication context are made plausible. (See Kimbrough [1990a] for a discussion of how the messages may be translated into first-order logic and how this translation can be used to prove the correctness of the various inferences that can be performed on a message.)

In our FLBC system concept, there are four main roles for inferencing related to messaging. First, during message initiation, inferencing is performed in order to validate the message before it is sent. We construe validation in a broad sense. It includes such matters as issuing a directive to a superior and issuing a directive to do something in the past. Second, on receipt, the message must be interpreted and handled appropriately. Unlike—or at least much more so than—in an EDI system, the message interpreter has, again, semantic access to the message; it can make inferences and initiate responses based on the manifest, composed syntax, which represents what the message means. (We elaborate on this point and illustrate it for X12 in Section 7.) The third sort of inferencing is what we call *system-level inferencing*. Using records of messages sent and received, various sorts of useful inferences may be drawn. For example, a user may inquire whether a directive he has issued has been responded to, or what directives addressed to him are outstanding. Finally, *application-level inferencing* may be performed by an application, treating messages sent and received as facts in a knowledge base.

We now discuss our implementation in terms of two specific scenarios.

6.1 Scenario 1

(1) *Mike*: Using the general-purpose message construction dialog box (see Figure 4), constructs a message informing Scott that Dave said that

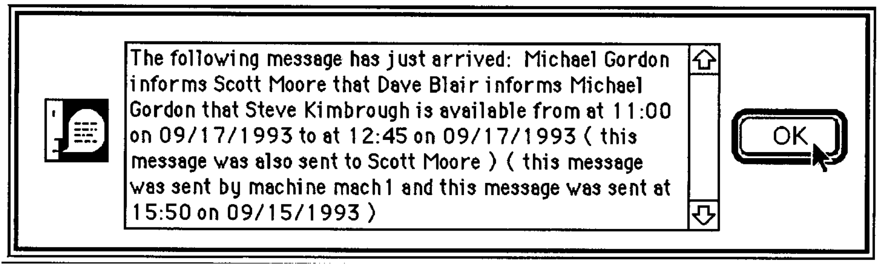

Fig. 5. Notification that a message has arrived.

Steve is available. The formal language representation of this message is as follows:

```
Msg (7) msg([person(p15)], person(p2), inform,
            simpleUtterance([person(p14)], person(p15),
              inform,
              available(person(p13),
                at(time(1993, 9, 17, 11, 0, 0)),
                at(time(1993, 9, 17, 12, 45, 0)))),
            [sendingMachine(mach1),
              timeSent(1993, 9, 15, 15, 50, 59)],
            msg627)
```

(2) *Scott*: Receives the message from Mike. This is immediately brought to Scott's attention (in a dialog box, as shown in Figure 5) since Scott has previously indicated that messages from Dave are important to him (e.g., Dave might be Scott's boss).

(3) *Scott*: From a dialog box, chooses to forward this message to Steve (see Figure 6).

(4) *Scott's Machine*: Formats the appropriate message in machine format, given the information from Scott; logs it locally; and forwards the message over the network to Steve. The message sent is as follows:

```
Msg (8) msg([person(p2)], person(p13), inform,
            forwardedFrom(msg627, person(person2)),
            [sendingMachine(mach1),
              timeSent(1993, 9, 15, 15, 56, 14)],
            msg628)
```

(5) *Steve*: Previously, has instructed his machine to be on the alert for statements from Dave (not Mike, not Scott, but *Dave*).

(6) *Steve's Machine*: Receives the message from Scott; logs it; recognizes that it implicitly contains a statement from Dave; and presents alerting dialog on screen next time Steve is logged on (see Figure 7).

The most important lesson of this scenario is that not only is expressive felicity handy but this power is useful only when a system is able to harness it effectively. The language allowed Scott to forward a message

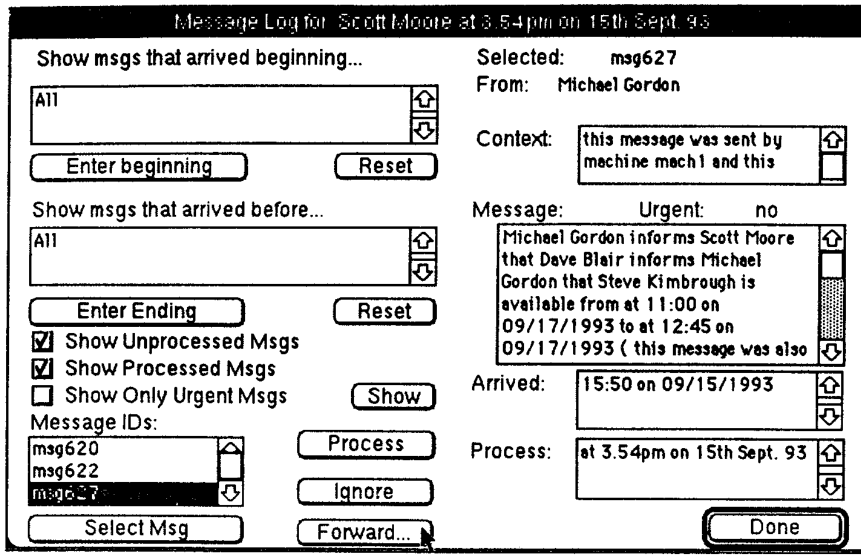

Fig. 6. Forwarding a message from the message log.

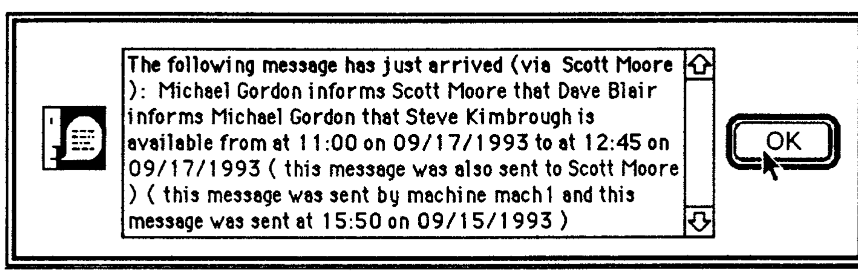

Fig. 7. Notification that a forwarded message implicitly contains a message from a person important to the recipient.

that contained a message that Mike said that Dave said something. In order to be able to use this message properly, the application receiving the message must be able to delve into the message to see what the content of the message actually is. The application must not simply look at the surface form that initially indicates that the message is from Scott. The language is more powerful than EDI, but this comes at a price: the applications must be sophisticated enough to use it.

6.2 Scenario 2

(1) *Scott*: Would like to make an appointment with Steve. Logs on to his machine and to the electronic messaging software.

(2) *Scott's Machine*: Presents Scott with a list of message type options (see Figure 8).

(3) *Scott*: Chooses the message type option *Request an Appointment*.

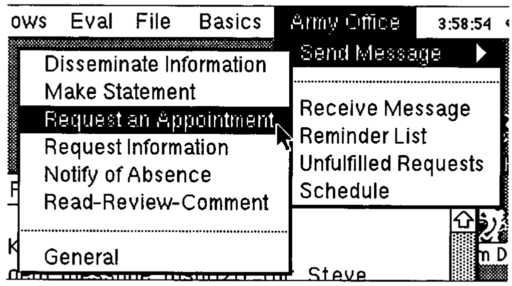

Fig. 8. Predefined message types in an office administration system.

(4) *Scott's Machine*: Prompts Scott for all required information, as well as for optional information for messages of type appointment (see Figure 9).

(5) *Scott*: Responds to prompts for (required and optional) information from his machine. Most responses to the prompts are given by scrolling menus, but additional text is typed in directly.

(6) *Scott's Machine*: Formats appropriate message in FLBC-2, given the information from Scott:

```
Msg (9) msg([person(p2)], person(p13), request,
             appointment(char240('the large conference room'),
               at(time(1993, 9, 21, 16, 0, 0)),
               at(time(1993, 9, 21, 17, 0, 0)),
               person(p2), [person(p13)]),
             [ ], msg630)
```

The message is logged locally, and the appointment is tentatively added to Scott's calendar (see Figure 10). The message is sent to Steve. The record of pending requests for Scott is updated (see Figure 11).

(7) *Steve's Machine*: Receives the request for an appointment message from Scott, and logs it locally. Categorizes it as a request from his boss, hence, an order. Recognizes that an acknowledgment is requested. Checks Steve's calendar, and finds it free for the date and time of the requested appointment. Adds the appointment to Steve's calendar, indicating the purpose and requester of the meeting, and the requested date and time. Formats and sends a message accepting the request for the meeting back to Scott:

```
Msg (10) msg([person(p13)], person(p2), inform,
              available(person(person13),
                at(time(1993, 9, 21, 16, 0, 0)),
                at(time(1993, 9, 21, 17, 0, 0))),
              [sendingMachine(mach1),
                timeSent(1993, 9, 15, 16, 6, 47),
                respondingTo(msg630)],
              msg635)
```

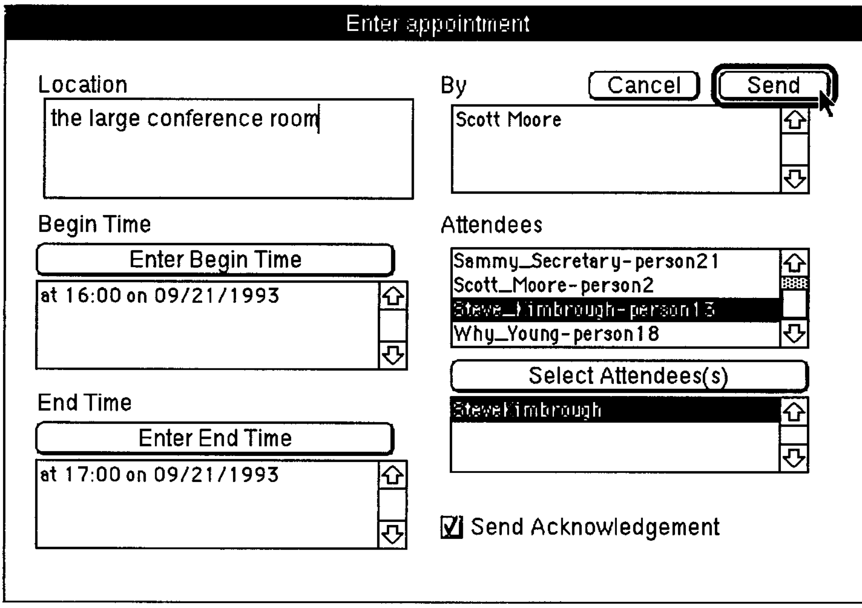

Fig. 9. Predefined dialog box for requesting an appointment.

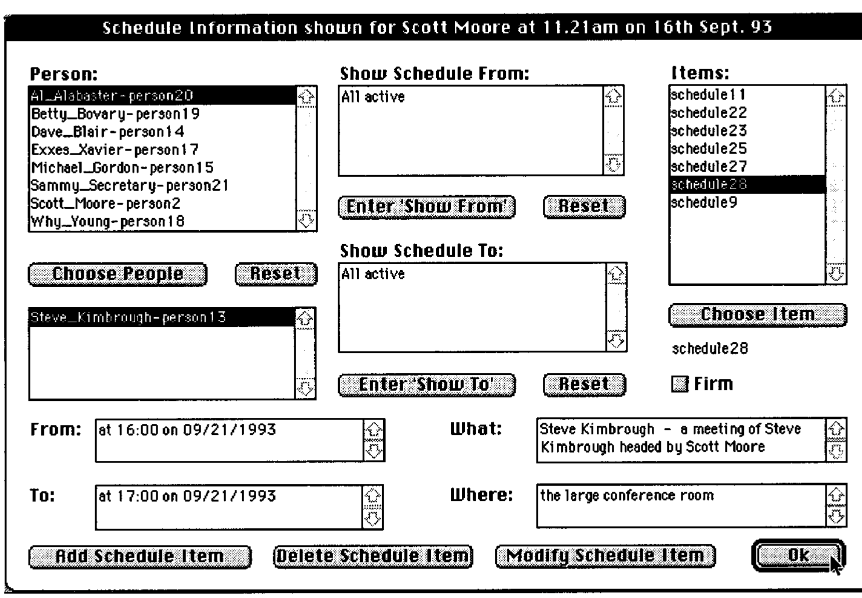

Fig. 10. Dialog box listing scheduled activities.

(8) *Scott's Machine*: Receives the acceptance message; updates Scott's calendar by changing the meeting to be a *firm* meeting.

(9) *Steve*: Logs on, checks his calendar for the day, and sees that he has a meeting with Scott scheduled.

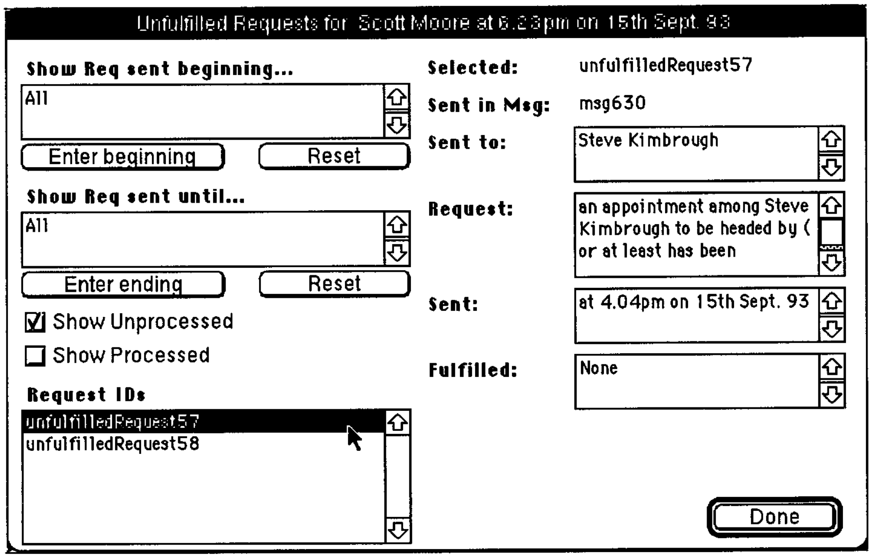

Fig. 11. Dialog box listing unfulfilled requests.

(10) *Scott*: Checks his calendar later in the day and sees that the meeting with Steve is now firm.

This scenario demonstrates that the application can do much work on behalf of the worker, freeing him to do other, more difficult tasks. Once Scott indicated he wanted a meeting with Steve, the system did all the coordination work for him. The result was a meeting between Scott and Steve, set up at a time at which they are both able to meet.

Obviously, scheduling meetings is not the only such task that can be performed by an application that receives a message. This scenario is a simple example pointing toward more complex and many other routine tasks a computer can do for us given languages that can more naturally support such activities.

7. EDI REVISITED

If an FLBC, such as FLBC-2, has been well designed, based on a solid theoretical foundation, then it ought to generalize. That is, we should be able to apply the language usefully in more than one context. We have seen that FLBC-2 works usefully (particularly for iterated speech act operators) in an Army office context. Can FLBC-2 be usefully applied to electronic commerce in general and EDI in particular? We think so, and it is the burden of this section to make a plausibility case for this.

We focus on a single, but entirely representative, EDI transaction set, X12's 840, "Request for Quotation" (RFQ). Figure 12 shows an example of a valid message for this transaction set. Figure 13 contains a rendering into English (by the X.12 standards committee) of this message.

```
[1] ST*840*159
[2] BQT*00*Q47391*820430
[3] N1*SE*X, Inc.
[4] N1*BY*Y Co.
[5] P01*1*30000*EA*0.42*PN*747355*PD*Circuit Network
[6] SCH*10000*EA****002*820604
[7] SCH*20000*EA****002*820709
[8] CCT*1*30000
[9] SE*9*159
```

Fig. 12. EDI X12 “Request for Quotation” (RFQ): Line numbers added.

```
[1] This is an RFQ Message * Message Number 159
[2] An Original Document * RFQ #Q47391 * Date: April 30, 1982
[3] Seller of item is X, Inc.
[4] Purchaser of item is Y Co.
[5] First Item: 30000 of part 747355 (a Circuit Network)
    at $0.42/item.
[6] Request that 10000 of the first item be delivered
    after June 4, 1982.
[7] Request that 20000 of the first item be delivered after
    July 9,1982.
[8] A total of 30000 items have been requested.
[9] There are 9 lines in this message.
    This is the end of message 159.
```

Fig. 13. Approximate English translation of X12’s 840 “Request for Quotation” (RFQ).

There is a lot to say, by way of comment, on this transaction set, which will generalize to the other transaction sets in X12 and to other EDI systems, such as EDIFACT and SWIFT. We are going to concentrate on just two such aspects of this message: the speech act structure and the date/time qualifiers. We devote a subsection to each, Sections 7.1 and 7.2, respectively.

7.1 Speech Act Structure

The RFQ message, Figure 12, is a request by Y Co. to X, Inc., for X, Inc., to provide Y Co. with a quote on the 3000 circuit networks specified. (Since Y lists the price, presumably Y is simply asking X to confirm that it will sell the networks at the price, dates, and quantities specified. But different trading partners can—and do—have different interpretation rules for this transaction set.) Now, a quote is a form of speech act, as is a request. We can plausibly interpret it either as a kind of assertion (“Yes, we sell things at that price.”) or, more likely, as a kind of promise (“We promise that if you agree to pay us 42¢ per network, we will deliver as indicated.”). Thus, this message’s structure is “Y requests that X promises to Y that Z” or, more plainly, request(promise(Z)). This is of the form $F_1(F_2(P))$ and is an example of what we are calling *iterated illocutionary forces* (cf. Section 1.4).

Not all EDI transaction sets involve iterated illocutionary forces, but many do and, as shown above, RFQ is one of them. But since X12 is alive

Table I. Cross Tabulation of Iterated Illocutionary Points in Electronic Commerce

	H			
S	Assert	Direct	Commit	Declare
Assert	1	2	3	4
Direct	5	6	7	8
Commit	9	10	11	12

and well, working with these iterated illocutionary forces, why bother with FLBC? Two of the most important reasons (there are others) are (1) economy of representation and (2) facilitation of inference.

By way of beginning to understand these reasons and why they apply, consider Table I. The table simply lays out certain combinations for two iterations of illocutionary points, as *S* utters (point in row) that *H* utters (point in column). Thus, cell 1 represents *S* asserting that *H* asserts ("She says that he says . . ."), while cell 2 stands for *S* asserting that *H* orders (directs) ("She says that he orders . . ."), and so on. Note that *S* and *H* are not necessarily different, and sometimes have to be identical. For example, you cannot promise for someone else, although you can promise that you will promise.

Recall that there are five fundamental illocutionary points: assertives, directives, commissives, declaratives, and emotives.[10] Table I leaves out emotives entirely, as they are not germane in this context. The table also lacks a row for *S* making a declaration (or "performative"). We have not found any examples of double-iteration declaratives of this sort, and we doubt they are important. If it is discovered otherwise, they can easily be added.

We are left with 12 cells in Table I. All are useful in commercial contexts.[11] To see this, consider a very simple commercial situation. We have three firms, Buyer, Seller, and Shipper, with the following departments:

—*Buyer*: Purchasing, Receiving, Accounts Payable, and Manufacturing;

—*Seller*: Sales, Shipping, and Billing; and

—*Shipper*.

Here are examples of common and useful utterance forms, mapped to the cells in Table I:

1, 4 Shipping asserts that Shipper [predicts | promises] to deliver before noon today.

[10]This is Searle's framework. Other frameworks differ, as we have previously noted, but these differences do not matter for the present purposes.

[11]This table only represents two iterations of illocutionary forces. Three and higher-order iterations do occur, and the points we make with respect to double iterations apply at least equally as well to the higher-order cases.

2, 3 Purchasing [asserts | directs] that Receiving declares the goods delivered in good order.

5, 8 Supervisor directs supervisee to [predict | promise] that $8 is the lowest available price for widgets.

6 Supervisor directs supervisee to query the Seller what the price of widgets is.

7 Supervisor directs supervisee to declare the delivery to be in good working order.

8 Supervisor directs supervisee to promise to keep the delivery dock open until midnight.

9 Manufacturing promises to predict requirements for widgets.

10 Management promises to direct Manufacturing to forward its predicted requirements to Seller in a timely fashion.

11 Receiving promises to reject (declare unfit) any shipment with any significant damage whatsoever.

12 Buyer promises to offer to buy (i.e., to promise contingent upon acceptance of the offer) from seller after the first of the fiscal year.

The X12 RFQ transaction set falls into cell 7 (or maybe 5) of Table I. That table contains 12 message types, for just double iterations of speech act operators, without regard to the content of the message (the P in terms of the $F(P)$ framework). Under the X12 (and generally for the EDI) way of representing messages, each of the 12 message types requires a separate, atomic representation ("840" in the case of RFQ). Under the FLBC approach, and in FLBC-2, we only need four atomic representations (assert, direct, commit, and declare), which we combine via a grammar to obtain the necessary distinctions. In this way, economy of representation is facilitated.

And economy of representation facilitates inferencing. In X12, given a log of EDI messages of various sorts it is a simple matter to answer the question "Which quotes did we request during December 1996?" It is quite another matter to answer the question "What are all requests we made during 1996 to X, Inc.?" The difficulty with the latter question, for X12, is that there are several hundred message types (transaction sets), scores of which are requests of some sort. Since the messages are only identified by transaction set number (e.g., 840), we cannot begin to answer this particular question unless we have executable rules at hand that identify, for each of the several hundred types of messages, which ones are requests by the speaker. This can certainly be done, but it costs and only applies for one category of management question: requests. The exercise would have to be repeated for other questions.

Actually, the situation is often more complicated than this. Some transaction sets (e.g., 832, 857, and 862 in X12) can be used to express more than one illocutionary force (in different message instances). This is also true in other EDI protocols. In SWIFT, for example, we find the following description of MT 304, "Advice/Instruction of a Third Party Deal" [SWIFT 1996, p. 35]:

> This message is sent by a fund manager to a custodian bank as an advice of/instruction to settle a third party foreign exchange deal.
>
> The definition of third party must be agreed up front between the fund manager and the custodian relative to deals executed by the custodian's treasury area on behalf of the fund manager.
>
> It is used to:
> - provide details about a new deal
> - provide a settlement notification
> - amend a previously sent message
> - cancel a previously sent message.

Again, the problem of locating, for example, requests or cancellations, could be handled by assembling executable rules (meaning postulates), as described below. These rules, however, would be complicated to write and difficult to maintain.

This kind of problem has been faced before by the information systems community and has been decisively resolved in favor of general, principled representations that facilitate very flexible, open-ended queries. We speak, of course, of relational database systems. Similarly, FLBCs generally, and FLBC-2 in particular, offer exactly these virtues when compared to standard EDI formats. Taking the present example, to answer the question "What have we requested?" find all messages we sent having the form request(X). To answer the question "What quotes have we requested?" find all messages we have sent having the form request(quote(X)).[12]

Let us now turn to date/time qualifiers in X12, where we will see these points brought home in spades.

7.2 Date/Time Qualifiers

Lines 6 and 7 of the RFQ message in Figure 12 each contain the token 002 preceding a date. The 002 is a date/time qualifier in X12, with the interpretation "Delivery Requested." We shall see that there are quite a few such qualifiers, but first let us see, fundamentally, why this has to be so and why an FLBC (and, in particular, FLBC-2) produces representational economy and facilitates inferencing.

Consider a simple example of a sentence operator, the necessity operator from modal logic, $\Box$. To represent "Necessarily, P and Q and R," we write

$$\Box(P \wedge Q \wedge R) \tag{1}$$

[12]If quote is taken as an illocutionary force, or request(promise(X)) where promise(X) has the form of a quote. Recall the constructibility thesis for SAT: the many illocutionary forces can all be defined in terms of constructions and qualifications of the five basic illocutionary points. Thus, for example, a quote would be an offer to sell, and an offer to sell is a promise to deliver certain goods contingent on a promise to transfer payment for the goods. Thus, it becomes practicable to use "higher-level" forces, e.g., quote, which then can be unpacked, and stored if necessary, in terms of their formal definitions.

Notice it happens here that the operator has as its scope the entire subsequent expression. In this case, necessity is like the illocutionary forces, where *every* expression falls under the scope of some illocutionary operator (recall the $F(P)$ thesis). Suppose, however, that we want to express the meaning of Expression (1), but without introducing a new operator. What could we do? We might introduce a symbol, S, that stands for $\Box(P \wedge Q \wedge R)$, along with rules that unpack the inferential relationships, for example, $S \rightarrow P$, $S \rightarrow Q$, and so on.

A somewhat better approach, short of introducing a necessity operator and its logic, would be to recognize that Expression (1) can be simplified to

$$(\Box P \wedge \Box Q \wedge \Box R) \tag{2}$$

Now, instead of introducing a symbol that represents the whole expression, we introduce new symbols and rules at a more atomic level. We might, for example, represent $\Box P$ by S and add the rule $S \rightarrow P$. Similarly, we might have T stand for $\Box Q$ and U stand for $\Box R$ (with added rules, $T \rightarrow Q$, etc.). Now, we can use $(S \wedge T \wedge U)$ to represent Expression (2).

This kind of move, which is called the introduction of *meaning postulates* in the logic literature, where it is usually disparaged, can be made to work in principle, but it is terribly clumsy and invites implementation difficulties.[13] It is a kludge if there ever was one. It is exactly what has been done in the X12 protocols and what must be done, given that the illocutionary operators, whose meanings are being used, are not introduced explicitly. In short, if, for example, we want to express a request to quote and if our syntax for saying so does not reflect the semantics of the underlying logical operators, then we get driven to the sort of move just described with respect to the necessity operator.

If this account is right in the main, then we would expect to see, for example, in X12, this sort of problem; that is, we would expect to see a large number of atomic symbols whose underlying logical meanings significantly overlap. This is exactly what we find. The case is especially egregious for date/time modifiers in X12. The (growing) list of more than 600 X12 date/time modifiers, as of December 1996, is given in Table II.

FLBC-2 offers a better way. We retain FLBC-2, Figure 1, and modify its basic vocabulary, Figure 3, slightly. Our RFQ is now represented by

```
Msg (11) msg('X Co.', 'Y, Inc.', request,
                msg('Y Inc.', 'X Co.', quote,
                Φ, reply('Q47391')),
                'Q47391')
```

[13]Actually, it is not even clear, either for modal logic or for illocutionary forces, that meaning postulates can really be made to work in a practical situation. We note, e.g., that in most modal logics the schema $\phi \rightarrow \Diamond\phi$ is a theorem, when instantiated by any well-formed formula, ϕ. But then from ϕ we can derive $\Diamond\Diamond\phi$ and so on infinitely. It is hard to see how a finite list of meaning postulates could accommodate this.

where 'Q47391' is the unique message ID, just as in the original RFQ, and reply(|) is a function, added to the vocabulary, that returns a unique name given its argument. Φ is a placeholder, which we now need to unpack.

'Q47391' is a unique ID, identifying the message. We need two further IDs, one for each delivery. Call them 'd1' and 'd2'. We need to add a few predicates to our basic vocabulary. Here they are, with their arguments filled in for the sake of providing examples:

(1) delivery('d1'): "'d1' is a delivery."
(2) to('d1', 'Y Co.'): "'d1' (a delivery) is to 'Y Co.'"
(3) itemID('d1', 747355): "The subject of the delivery 'd1' is items of ID 747355."
(4) itemDescription(747355, 'Circuit Network'): "Item 747355 is a 'Circuit Network.'"
(5) from('d1', 'X, Inc.'): "'d1' (a delivery) is from 'X, Inc."
(6) numberOfUnits('d1', 747355, 10000): "The number of units of 747355 in 'd1' (a delivery) is 10000."
(7) unitPrice('d1', 747355, 0.42): "The price of 747355 in 'd1' (a delivery) is 0.42."
(8) after('d1', 820604): "'d1' (a delivery) occurs after June 4, 1982."

Given these additions to our lexicon, Φ becomes the following:

```
Msg (12) and(delivery('d1'), to('d1', 'Y, Inc.'),
              itemID('d1', 747355), itemDescription(747355, 'Circuit
                Network'),
              from('d1', 'X, Inc.'), numberOfUnits('d1', 747355, 10000),
              unitPrice('d1', 747355, 0.42), after('d1', 820604),
              delivery('d2'), to('d2', 'Y, Inc.'),
              itemID('d2', 747355),
              from('d2', 'X, Inc.'), numberOfUnits('d1', 747355, 20000),
              unitPrice('d2', 747355, 0.42), after('d2', 820709))
```

This, we submit, accurately represents the RFQ message in FLBC-2, with the lexicon augmented by the above eight predicates. Having looked at many other EDI transaction sets, we are convinced that the findings of this one case generalize very nicely. Furthermore, we note that FLBC-2, Figure 1, allows for Boolean combinations of message contents, something not countenanced in any EDI transaction set we are aware of.

The fact that we have had to add the predicates for representing this first, RFQ, message should not be surprising. EDI and Army offices are different application areas. Also, perusing the above list of X12 date/time qualifiers should make it clear that all of our eight new predicates will be useful in many other places and will permit significant representational economies. Here, for example, is a sampling of illocutionary forces used in X12 EDI messages:

(1) assert

Table II. Code Values Specifying Type of Date or Time, or Both Date and Time

Code	Values	Code	Values
001	Cancel After	052	Cumulative Quantity End
002	Delivery Requested	053	Buyers Local
003	Invoice	054	Sellers Local
004	Purchase Order	055	Confirmed
005	Sailing	056	Estimated Port of Entry
006	Sold	057	Actual Port of Entry
007	Effective	058	Customs Clearance
008	Purchase Order Received	059	Inland Ship
009	Process	060	Engineering Change Level
010	Requested Ship	061	Cancel if Not Delivered by
011	Shipped	062	Blueprint
012	Terms Discount Due	063	Do Not Deliver After
013	Terms Net Due	064	Do Not Deliver Before
014	Deferred Payment	065	1st Schedule Delivery
015	Promotion Start	066	1st Schedule Ship
016	Promotion End	067	Current Schedule Delivery
017	Estimated Delivery	068	Current Schedule Ship
018	Available/Constructive Placement	069	Promised for Delivery
019	Unloaded	070	Scheduled for Delivery (After and Including)
020	Check	071	Requested for Delivery (After and Including)
021	Charge Back	072	Promised for Delivery (After and Including)
022	Freight Bill	073	Scheduled for Delivery (Prior to and Including)
023	Promotion Order—Start	074	Requested for Delivery (Prior to and Including)
024	Promotion Order—End	075	Promised for Delivery (Prior to and Including)
025	Promotion Ship—Start	076	Scheduled for Delivery (Week of)
026	Promotion Ship—End	077	Requested for Delivery (Week of)
027	Promotion Requested Delivery—Start	078	Promised for Delivery (Week of)
028	Promotion Requested Delivery—End	079	Promised for Shipment
029	Promotion Performance—Start	080	Scheduled for Shipment (After and Including)
030	Promotion Performance—End	081	Requested for Shipment (After and Including)
031	Promotion Invoice Performance—Start	082	Promised for Shipment (After and Including)
032	Promotion Invoice Performance—End	083	Scheduled for Shipment (Prior to and Including)
033	Promotion Floor Stock Protect—Start	084	Requested for Shipment (Prior to and Including)
034	Promotion Floor Stock Protect—End	085	Promised for Shipment (Prior to and Including)
035	Delivered	086	Scheduled for Shipment (Week of)
036	Expiration	087	Requested for Shipment (Week of)
037	Ship Not Before	088	Promised for Shipment (Week of)
038	Ship No Later	089	Inquiry
039	Ship Week of	090	Report Start
040	Status (After and Including)	091	Report End
041	Status (Prior and Including)		
042	Superseded		
043	Publication		
044	Settlement Date as Specified by the Originator		
045	Endorsement Date		
046	Field Failure		
047	Functional Test		
048	System Test		
049	Prototype Test		
050	Received		
051	Cumulative Quantity Start		

Table II. *Continued*

Code	Values	Code	Values
092	Contract Effective	143	Due Date of First Payment to Principal and Interest
093	Contract Expiration	144	Estimated Acceptance
094	Manufacture	145	Opening Date
095	Bill of Lading	146	Closing Date
096	Discharge	147	Due Date Last Complete Installment Paid
097	Transaction Creation	148	Date of Local Office Approval of Conveyance of & Damaged Real Estate Property
098	Bid (Effective)	149	Date Deed Filed for Record
099	Bid Open (Date Bids Will Be Opened)	150	Service Period Start
100	No Shipping Schedule Established as of	151	Service Period End
101	No Production Schedule Established as of	152	Effective Date of Change
102	Issue	153	Service Interruption
103	Award	154	Adjustment Period Start
104	System Survey	155	Adjustment Period End
105	Quality Rating	156	Allotment Period Start
106	Required By	157	Test Period Start
107	Deposit	158	Test Period Ending
108	Postmark	159	Bid Price Exception
109	Received at Lockbox	160	Samples to be Returned By
110	Originally Scheduled Ship	161	Loaded on Vessel
111	Manifest/Ship Notice	162	Pending Archive
112	Buyers Dock	163	Actual Archive
113	Sample Required	164	First Issue
114	Tooling Required	165	Final Issue
115	Sample Available	166	Message
116	Scheduled Interchange Delivery	167	Most Recent Revision (or Initial Version)
118	Requested Pick-up	168	Release
119	Test Performed	169	Product Availability Date
120	Control Plan	170	Supplemental Issue
121	Feasibility Sign Off	171	Revision
122	Failure Mode Effective	172	Correction
124	Group Contract Effective	173	Week Ending
125	Group Contract Expiration	174	Month Ending
126	Wholesale Contract Effective	175	Cancel if not shipped by
127	Wholesale Contract Expiration	176	Expedited on
128	Replacement Effective	177	Cancellation
129	Customer Contract Effective	178	Hold (as of)
130	Customer Contract Expiration	179	Hold as Stock (as of)
131	Item Contract Effective	180	No Promise (as of)
132	Item Contract Expiration	181	Stop Work (as of)
133	Accounts Receivable—Statement Date	182	Will Advise (as of)
134	Ready for Inspection	183	Connection
135	Booking	184	Inventory
136	Technical Rating	185	Vessel Registry
137	Delivery Rating	186	Invoice Period Start
138	Commercial Rating	187	Invoice Period End
139	Estimated	188	Credit Advice
140	Actual	189	Debit Advice
141	Assigned		
142	Loss		

Table II. *Continued*

Code	Values
190	Released to Vessel
191	Material Specification
192	Delivery Ticket
193	Period Start
194	Period End
195	Contract Re-Open
196	Start
197	End
198	Completion
199	Seal
200	Assembly Start
201	Acceptance
202	Master Lease Agreement
203	First Produced
204	Official Rail Car Interchange & (Either Actual or Agreed Upon)
206	Status (Outside Processor)
207	Status (Commercial)
208	Lot Number Expiration
209	Contract Performance Start
210	Contract Performance Delivery
211	Service Requested
212	Returned to Customer
213	Adjustment to Bill Dated
214	Date of Repair/Service
215	Interruption Start
216	Interruption End
217	Spud
218	Initial Completion
219	Plugged and Abandoned
220	Penalty
221	Penalty Begin
222	Birth
223	Birth Certificate
224	Adoption
225	Christening
226	Lease Commencement
227	Lease Term Start
228	Lease Term End
229	Rent Start
230	Installation
231	Progress Payment
232	Claim Statement Period Start
233	Claim Statement Period End
234	Settlement Date
235	Delayed Billing (Not Delayed Payment)
236	Lender Credit Check
237	Student Signed
238	Schedule Release
239	Baseline
240	Baseline Start

Code	Values
241	Baseline Complete
242	Actual Start
243	Actual Complete
244	Estimated Start
245	Estimated Completion
246	Start no earlier than
247	Start no later than
248	Finish no later than
249	Finish no earlier than
250	Mandatory (or Target) Start
251	Mandatory (or Target) Finish
252	Early Start
253	Early Finish
254	Late Start
255	Late Finish
256	Scheduled Start
257	Scheduled Finish
258	Original Early Start
259	Original Early Finish
260	Rest Day
261	Rest Start
262	Rest Finish
263	Holiday
264	Holiday Start
265	Holiday Finish
266	Base
267	Timenow
268	End Date of Support
269	Date Account Matures
270	Date Filed
271	Penalty End
272	Exit Plant Date
273	Latest On Board Carrier Date
274	Requested Departure Date
275	Approved
276	Contract Start
277	Contract Definition
278	Last Item Delivery
279	Contract Completion
280	Date Course of Orthodontics Treatment & Began or is Expected to Begin
281	Over Target Baseline Month
282	Previous Report
283	Funds Appropriation—Start
284	Funds Appropriation—End
285	Employment or Hire
286	Retirement
287	Medicare
288	Consolidated Omnibus Budget Reconciliation Act (COBRA)
289	Premium Paid to Date

Table II. *Continued*

Code	Values
290	Coordination of Benefits
291	Plan
292	Benefit
293	Education
294	Earnings Effective Date
295	Primary Care Provider
296	Return to Work
297	Date Last Worked
298	Latest Absence
299	Illness
300	Enrollment Signature Date
301	Consolidated Omnibus Budget Reconciliation Act & (COBRA) Qualifying Event
302	Maintenance
303	Maintenance Effective
304	Latest Visit or Consultation
305	Net Credit Service Date
306	Adjustment Effective Date
307	Eligibility
309	Plan Termination
310	Date of Closing
311	Latest Receiving Date/Cutoff Date
312	Salary Deferral
313	Cycle
314	Disability
315	Offset
316	Prior Incorrect Date of Birth
317	Corrected Date of Birth
319	Failed
320	Date Foreclosure Proceedings Instituted
321	Purchased
322	Put into Service
323	Replaced
324	Returned
327	Quarter Ending
328	Changed
329	Terminated
330	Referral Date
331	Evaluation Date
332	Placement Date
333	Individual Education Plan (IEP)
334	Re-evaluation Date
335	Dismissal Date
336	Employment Begin
337	Employment End
338	Medicare Begin
339	Medicare End
340	Consolidated Omnibus Budget Reconciliation Act (COBRA) Begin
341	Consolidated Omnibus Budget Reconciliation Act (COBRA) End
342	Premium Paid to Date Begin
343	Premium Paid to Date End
344	Coordination of Benefits Begin
345	Coordination of Benefits End
346	Plan Begin
347	Plan End
348	Benefit Begin
349	Benefit End
350	Education Begin
351	Education End
352	Primary Care Provider Begin
353	Primary Care Provider End
354	Illness Begin
355	Illness End
356	Eligibility Begin
357	Eligibility End
358	Cycle Begin
359	Cycle End
360	Disability Begin
361	Disability End
362	Offset Begin
363	Offset End
364	Plan Period Election Begin
365	Plan Period Election End
366	Plan Period Election
367	Due to Customer
368	Submittal
369	Estimated Departure Date
370	Actual Departure Date
371	Estimated Arrival Date
372	Actual Arrival Date
373	Order Start
374	Order End
375	Delivery Start
376	Delivery End
377	Contract Costs Through
378	Financial Information Submission
379	Business Termination
380	Applicant Signed
381	Cosigner Signed
382	Enrollment
383	Adjusted Hire
384	Credited Service
385	Credited Service Begin
386	Credited Service End
387	Deferred Distribution
388	Payment Commencement
389	Payroll Period
390	Payroll Period Begin
392	Plan Entry
393	Plan Participation Suspension

Table II. *Continued*

Code	Values	Code	Values
394	Rehire	445	Initial Placement
395	Retermination	446	Replacement
396	Termination	447	Occurrence
397	Valuation	448	Occurrence Span
398	Vesting Service	449	Occurrence Span From
399	Vesting Service Begin	450	Occurrence Span To
400	Vesting Service End	451	Initial Fee Due
401	Duplicate Bill	452	Appliance Placement
402	Adjustment Promised	453	Acute Manifestation of a Chronic Condition
403	Adjustment Processed	454	Initial Treatment
404	Year Ending	455	Last X-Ray
405	Production	456	Surgery
406	Material Classification	457	Continuous Passive Motion (CPM))
408	Weighed	458	Certification
409	Date of Deed in Lieu	459	Nursing Home From
410	Date of Firm Commitment	460	Nursing Home To
411	Expiration Date of Extension to Foreclose	461	Last Certification
412	Date of Notice to Convey	462	Date of Local Office Approval Conveyance of Occupied
413	Date of Release of Bankruptcy	463	Begin Therapy
414	Optimistic Early Start	464	Oxygen Therapy From
415	Optimistic Early Finish	465	Oxygen Therapy To
416	Optimistic Late Start	466	Oxygen Therapy
417	Optimistic Late Finish	467	Signature
418	Most Likely Early Start	468	Prescription Fill
419	Most Likely Early Finish	469	Provider Signature
420	Most Likely Late Start	470	Date of Local Office Certification of Conveyance of & Damaged Real Estate
421	Most Likely Late Finish	471	Prescription
422	Pessimistic Early Start	472	Service
423	Pessimistic Early Finish	473	Medicaid Begin
424	Pessimistic Late Start	474	Medicaid End
425	Pessimistic Late Finish	475	Medicaid
426	First Payment Due	476	Peer Review Organization (PRO) Approved Stay From
427	First Interest Payment Due	477	Peer Review Organization (PRO) Approved Stay To
428	Subsequent Interest Payment Due	478	Prescription From
429	Irregular Interest Payment Due	479	Prescription To
430	Guarantor Received	480	Arterial Blood Gas Test
431	Onset of Current Symptoms or Illness	481	Oxygen Saturation Test
432	Submission	482	Pregnancy Begin
434	Statement	483	Pregnancy End
435	Admission	484	Last Menstrual Period
436	Insurance Card	485	Injury Begin
437	Spouse Retirement	486	Injury End
438	Onset of Similar Symptoms or Illness	487	Nursing Home
439	Accident	488	Collateral Dependent
440	Release of Information	489	Collateral Dependent Begin
441	Prior Placement	490	Collateral Dependent End
442	Date of Death		
443	Peer Review Organization (PRO) Approved Stay		
444	First Visit or Consultation		

Table II. *Continued*

Code	Values
491	Sponsored Dependent
492	Sponsored Dependent Begin
493	Sponsored Dependent End
494	Deductible
495	Out-of-Pocket
496	Contract Audit Date
497	Latest Delivery Date at Pier
498	Mortgagee Reported Curtailment Date
499	Mortgagee Official Signature Date
500	Resubmission
501	Expected Reply
502	Dropped to Less than Half Time
503	Repayment Begin
504	Loan Servicing Transfer
505	Loan Purchase
506	Last Notification
507	Extract
508	Extended
509	Servicer Signature Date
510	Date Packed
511	Shelf Life Expiration
512	Warranty Expiration
513	Overhauled
514	Transferred
515	Notified
516	Discovered
517	Inspected
518	Voucher (Date of)
519	Date Bankruptcy Filed
520	Date of Damage
521	Date Hazard Insurance Policy Cancelled
522	Expiration Date to Submit Title Evidence
523	Date of Claim
524	Date of Notice of Referral for Assignment
525	Date of Notice of Probable Ineligibility of Assignment
526	Date of Foreclosure Notice
527	Expiration of Foreclosure Timeframe
528	Date Possessory Action Initiated
529	Date of Possession
531	Date of Acquisition of Title
532	Expiration of Extension to Convey
533	Date of Assignment Approval
534	Date of Assignment Rejection
535	Curtailment Date from Advice of Payment
536	Expiration of Extension to Submit Fiscal Data
537	Date Documentation
538	Makegood Commercial Date
539	Policy Effective
540	Policy Expiration
541	Employee Effective Date of Coverage
542	Date of Representation
543	Last Premium Paid Date
544	Date Reported to Employer
545	Date Reported to Claim Administrator
546	Date of Maximum Medical Improvement
547	Date of Loan
548	Date of Advance
549	Beginning Lay Date
550	Certificate Effective
551	Benefit Application Date
552	Actual Return to Work
553	Released Return to Work
554	Ending Lay Date
555	Employee Wages Ceased
556	Last Salary Increase
557	Employee Laid Off
558	Injury or Illness
559	Oldest Unpaid Installment
560	Preforeclosure Acceptance Date
561	Preforeclosure Sale Closing Date
562	Date of First Uncured Default
563	Date Default Was Cured
564	Date of First Mortgage Payment
565	Date of Property Inspection
566	Date Total Amount of Delinquency Reported
567	Date Outstanding Loan Balance Reported
568	Date Foreclosure Sale Scheduled
569	Date Foreclosure Held
570	Date Redemption Period Ends
571	Date Voluntary Conveyance Accepted
572	Date Property Sold
573	Date Claim Paid
574	Action Begin Date
575	Projected Action End Date
576	Action End Date
577	Original Maturity Date
578	Date Referred to Attorney for Foreclosure
579	Planned Release
580	Actual Release
581	Contract Period
582	Report Period
583	Suspension
584	Reinstatement

Table II. *Continued*

Code	Values	Code	Values
585	Report	600	As Of
586	First Contact	601	First Submission
587	Projected Foreclosure Sale Date	602	Subsequent Submission
589	Date Assignment Filed for Record	700	Override Date for Settlement
590	Date of Appraisal	701	Interline Settlement System Assigned
591	Expiration Date of Extension to Assign	702	Sending Road Time Stamp
592	Date of Extension to Convey	703	Original Transaction
593	Date Hazard Insurance Policy Refused	704	Delivery Appointment Date and Time
594	High Fabrication Release Authorization	706	Date Material Usage Suspended
595	High Raw Material Authorization	993	Request for Quotation
596	Material Change Notice	994	Quote
597	Latest Delivery Date at Rail Ramp	996	Required Delivery
598	Rejected	997	Quote to be Received By
		ZZZ	Mutually Defined

(2) cancel (illocutionary point: a declarative, ≈ declare void)
(3) clear (illocutionary point: declarative)
(4) confirm (illocutionary point: assertive)
(5) declare
(6) defer (illocutionary point: declarative)
(7) endorse (illocutionary point: declarative)
(8) estimate (illocutionary point: assertive)
(9) order (illocutionary point: commissive or directive)
(10) promise
(11) request
(12) schedule (illocutionary point: assertive or directive)

Incorporating them into FLBC-2 is straightforward. Furthermore, here is a sampling of ordinary verbs (not indicating speech acts) used in X12 EDI messages:

(1) arrive
(2) change
(3) charge back
(4) check
(5) deliver
(6) end
(7) expire
(8) fail
(9) perform
(10) process
(11) promote
(12) protect
(13) publish
(14) receive
(15) sell
(16) settle
(17) ship
(18) start
(19) supersede
(20) test
(21) unload

Again, incorporating them into FLBC-2 is straightforward.

The fact that FLBC-2 worked in both application areas (we have examined and confirmed many cases beyond those reported here) lends support to the hypothesis that the speech act framework, with iterated illocutionary operators, may have a very broad range of valid applications. Only much more extensive studies and evaluations of fielded systems will determine whether this is in fact the case, but our available evidence is quite favorable.

8. CONCLUSION

We have described results from a much more extensive project on formal languages for business communications. The basic findings and ideas may be summarized as follows:

(1) The value, in a business communications context, of syntactically articulating messages, for the purpose of supporting processing and inferencing on the messages (i.e., for semantic access), has been amply demonstrated by experience with EDI and by prototype tagged-message electronic messaging systems. The full value of this idea, however, is far from being realized.

(2) The idea of an FLBC is a generalization of EDI and tagged-message electronic mail systems, and promises to provide the basis for expressively and inferentially rich computerized messaging systems.

(3) Any FLBC implementation ought to be theoretically motivated. Recent theoretical work in philosophy of language and linguistics—in pragmatics generally and SAT particularly—holds great promise of providing an adequate theoretical basis for FLBCs.

(4) The family of languages, FLBC-2, presented and discussed here, and upon which the prototype system was built, can be rigorously specified and conform to the principles of SAT.

(5) The architecture of an FLBC system may be thought of as a generalization of the EDI architecture. In the prototype, discussed here, of such an FLBC system, four main roles were found for inferencing on messages: (1) validation during message generation, (2) message interpretation, (3) system-level inferences, and (4) application-level inferences, where (3) and (4) may integrate knowledge about messages with application-specific knowledge.

This said, much remains to be done. Theory needs to be broadened and deepened. FLBCs need to be defined and studied more systematically. Prototypes need to be used and experimented with. But these are topics for other articles.

ACKNOWLEDGMENTS

Thanks go to Major Michael J. Thornburg (U.S. Army) and Ronald M. Lee for useful suggestions and comments, to Michael Covington for stimulating discussions, and to the anonymous referees for a number of helpful suggestions.

REFERENCES

ADAM, N. R. AND YESHA, Y., EDS. 1996. *Electronic Commerce: Current Research Issues and Applications*. Lecture Notes in Computer Science, vol. 1028. Springer-Verlag, Berlin, Germany.

AUSTIN, J. L. 1962. *How to Do Things with Words*. Oxford at the Clarendon Press, Oxford, U.K.

BACH, K. AND HARNISH, R. M. 1979. *Linguistic Communication and Speech Acts*. The MIT Press, Cambridge, Mass.

BHARGAVA, H. K. AND KIMBROUGH, S. O. 1995. On embedded languages, meta-level reasoning and computer-aided modeling. In *The Impact of Emerging Technologies on Computer Science and Operations Research*, S. G. Nash and A. Sofer, Eds. Kluwer Academic Publishers, Boston, Mass., 27–44.

BLAIR, D. C. 1992. Information retrieval and the philosophy of language. *Comput. J. 35*, 3 (June), 200–207.

BLEAKLEY, F. R. 1994. Fast money: Electronic payments now supplant checks at more large firms. *Wall St. J*. (Apr. 13), A1+.

CAMPBELL, K. S. 1990. Explanations in negative messages: More insights from speech act theory. *J. Bus. Commun. 27*, 4 (Fall), 357–375.

CASE, G. R. 1982. Feedback on electronic junk. *Commun. ACM 25*, 6 (June), 378–398.

CHANG, S. K. AND LEUNG, L. 1987. A knowledge-based message management system. *ACM Trans. Off. Inf. Syst. 5*, 3 (July), 213–236.

COHEN, P. R., MORGAN, J., AND POLLACK, M. E., Eds. 1990. *Intentions in Communication*. System Development Foundation Benchmark. The MIT Press, Cambridge, Mass.

COMER, D. E. AND PETERSON, L. L. 1985. Conversations—An alternative to memos and conferences. *Byte 10*, 13 (Dec.), 263–272.

COMER, D. E. AND PETERSON, L. L. 1986. Conversation based mail. *ACM Trans. Comput. Syst. 4*, 4 (Oct), 299–319.

COVINGTON, M. 1996. Toward a new type of language for electronic commerce. In *Proceedings of the 29th Annual Hawaii International Conference on System Sciences* Vol. 4, *Information Systems—Organizational Systems and Technology*, J. F. Nunamaker, Jr. and R. H. Sprague, Jr., Eds. IEEE Computer Society Press, Los Alamitos, Calif., 329–336.

COVINGTON, M. A. 1997. Speech acts in electronic communication, with special reference to KQML and ANSI X12. In *Proceedings of the 30th Annual Hawaii International Conference on System Sciences*, J. F. Nunamaker, Jr. and R. H. Sprague, Jr., Eds. IEEE Computer Society Press, Los Alamitos, Calif.

DE, S. 1986. Providing effective decision support: Modeling users and their requirements. *Decis. Support Syst. 2*, 4 (Dec.), 309–319.

DESANCTIS, G. AND GALLUPE, R. B. 1987. A foundation for the study of group decision support systems. *Manage. Sci. 33*, 5 (May), 589–609.

EDWARDS, D. W. 1987. Electronic data interchange: A senior management overview. Tech. Rep., International Center for Information Technologies, Washington, D.C.

EMMELHAINZ, M. A. 1993. *EDI: A Total Management Guide*. 2nd ed. Van Nostrand Reinhold, New York.

FININ, T., FRITZSON, R., MCKAY, D., AND MCENTIRE, R. 1994. KQML as an agent communication language. In *Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM 94)* (Nov.). ACM Press, New York.

FININ, T., WEBER, J., WIEDERHOLD, G., GENESERETH, M., FRITZON, R., MCKAY, D., MCGUIRE, J., PELAVIN, R., SHAPIRO, S., AND BECK, C. 1993. Draft specification of the KQML agent-communication language plus example agent policies and architectures. Dept. of Computer Science and Engineering, Univ. of Maryland Baltimore County, Baltimore, Md. Available via http://www.cs.umbc.edu/kqml/kqmlspec/spec.html.

FLORES, F., GRAVES, M., HARTFIELD, B., AND WINOGRAD, T. 1988. Computer systems and the design of organizational interaction. *ACM Trans. Off. Inf. Syst. 6*, 2 (Apr.), 153–172.

FOSTER, D. F. 1994. Electronic commerce: A business perspective. *EDI World 4*, 6 (June), 28–31.

FROST AND SULLIVAN. 1988. The electronic data interchange (EDI) market in the U.S. Tech. Rep., Frost and Sullivan, New York, Apr.

GERLACH, M. L. 1992. *Alliance Capitalism*. Oxford University Press, New York.

HARRISON, B. 1994. *Lean and Mean: The Changing Landscape of Corporate Power in the Age of Flexibility*. Basic Books, New York.

HILTZ, S. R. AND TUROFF, M. 1985. Structuring computer-mediated communications systems to avoid information overload. *Commun. ACM 28*, 7 (July), 680–689.

KAYE, A. R. AND KARAM, G. M. 1987. Cooperating knowledge-based assistants for the office. *ACM Trans. Off. Inf. Syst. 5*, 4 (Oct.), 297–326.

KIMBERLEY, P. 1991. *Electronic Data Interchange*. McGraw-Hill, New York.

KIMBROUGH, S. O. 1990a. On representation schemes for promising electronically. *Decis. Support Syst. 6*, 2 (May), 99–122.

KIMBROUGH, S. O. 1990b. On two foundational assumptions for computer-based semi-autonomous inter-organizational communication. Tech. Rep., The Wharton School, Dept. of Operations and Information Management, Univ. of Pennsylvania, Philadelphia, Pa., Nov.

KIMBROUGH, S. O. AND LEE, R. M. 1986. On illocutionary logic as a telecommunications language. In *Proceedings of the 7th International Conference on Information Systems* (San Diego, Calif., Dec. 15–17). The Society for Management Information Systems, Athens, Ga., 15–26.

KIMBROUGH, S. O. AND MOORE, S. A. 1992. Message management systems: Concepts, motivations and strategic effects. *J. Manage. Inf. Syst. 9*, 2 (Fall), 29–52.

KIMBROUGH, S. O. AND MOORE, S. A. 1993. On obligation, time, and defeasibility in systems for electronic commerce. In *Proceedings of the 26th Annual Hawaii International Conference on System Sciences.* Vol. 3, *Information Systems: DSS/Knowledge-Based Systems*, J. F. Nunamaker, Jr. and R. H. Sprague, Jr., Eds. IEEE Computer Society Press, Los Alamitos, Calif., 493–502.

KIMBROUGH, S. O. AND THORNBURG, M. 1989. On messaging with semantic access in an army office environment. In *Proceedings of the 22nd Annual Hawaii International Conference on System Sciences* Vol. 3. IEEE Computer Society Press, Los Alamitos, Calif., 566–574.

LABROU, Y. AND FININ, T. 1994. A semantics approach for KQML—A general purpose communication language for software agents. In *Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM 94)* (Gaithersburg, Md., Nov 29–Dec. 2). ACM Press, New York.

LEE, R. M. 1980. CANDID—A Logical calculus for describing financial contracts. Ph.D. thesis. Work. pap. 80-06-2, Dept. of Decision Sciences (now Dept. of Operations and Information Management), Univ. of Pennsylvania, Philadelphia, Pa.

LEE, R. M. 1984. Automating red tape: The performative vs informative roles of bureaucratic documents. *Off.: Technol. People 2*, 187–194.

LEE, R. M. 1988a. International contracting—A formal language approach. In *Proceedings of the 21st Annual Hawaii International Conference on System Sciences, Volume I, Applications*, R. H. Sprague, Ed. IEEE Computer Society Press, Los Alamitos, Calif., 69–78.

LEE, R. M. 1988b. A logic model for electronic contracting. *Decis. Support Syst. 4*, 27–44.

LEE, R. M. AND BOSE, R. 1988. Deontic reasoning in bureaucratic systems. In *Proceedings of the 21st Annual Hawaii International Conference on System Sciences, Volume III, Decision Support and Knowledge Based Systems*, B. R. Konsynski, Ed. IEEE Computer Society Press, Los Alamitos, Calif., 477–485.

LEE, R. M. AND RYU, Y. 1989. Event grammars: A modeling representation for business procedures. In *Proceedings of the 22nd Annual Hawaii International Conference on System Sciences, Volume III, Decision Support and Knowledge Based Systems*, R. Blanning and D. King, Eds. IEEE Computer Society Press, Los Alamitos, Calif., 553–559.

LEE, R. M. AND WIDMEYER, G. 1986. Shopping in the electronic marketplace. *J. Manage. Inf. Syst. 2*, 4 (Spring), 21–35.

LEHMANN, F. 1996. Machine-negotiated, ontology-based EDI (electronic data interchange). In *Electronic Commerce: Current Research Issues and Applications*, N. R. Adam and Y. Yesha, Eds. Lecture Notes in Computer Science, vol. 1028. Springer-Verlag, Berlin, Germany, 27–46.

LEHTINEN, E. AND LYYTINEN, K. 1986. Action based model of information systems. *Inf. Syst. 11*, 4, 299–317.

LEVINSON, S. C. 1983. *Pragmatics*. Cambridge University Press, Cambridge, U.K.

LYYTINEN, K. 1987. Two views of information modeling. *Inf. Manage. 12*, 1 (Jan.), 9–19.

LYYTINEN, K. J. 1985. Implications of theories of language for information systems. *MIS Q. 9*, 1 (Mar.), 61–74.

MALONE, T. W., GRANT, K. R., LAI, K.-Y., RAO, R., AND ROSENBLITT, D. 1987a. Semistructured messages are surprisingly useful for computer-supported coordination. *ACM Trans. Off. Inf. Syst. 5*, 2, (Apr.), 115–131.

MALONE, T. W., GRANT, K. R., TURBAK, F. A., BROBST, S. A., AND COHEN, M. D. 1987b. Intelligent information-sharing systems. *Comm. ACM 30*, 5 (May), 390–402.

MANTELMAN, L. 1987. Orchestrating people and computers in their networks. *Data Commun. 16*, 10 (Sept.), 144–162.

MAYFIELD, J., LABROU, Y., AND FININ, T. 1995. Desiderata for agent communication languages. In *Proceedings of the AAAI Symposium on Information Gathering from Heterogeneous, Distributed Environments, AAAI-95 Spring Symposium* (Mar. 27–29). American Association for Artificial Intelligence, Menlo Park, Calif.

MAYFIELD, J., LABROU, Y., AND FININ, T. 1996. Evaluation of KQML as an agent communication language. In *Intelligent Agents.* Vol. 2, *Proceedings of the 1995 Workshop on Agent Theories, Architectures, and Languages*, M. Wooldridge, J. P. Muller, and M. Tambe, Eds. Springer-Verlag, Berlin, Germany.

MCCARTHY, J. 1982. The common business communication language. In *Textverarbeitung und Bürosysteme*, A. Endres and J. Reetz, Eds. R. Oldenbourg Verlag, Munich, Germany.

MOORE, S. A. 1993. Saying and doing: Uses of formal languages in the conduct of business. Ph.D. thesis, The Wharton School, Dept. of Operations and Information Management, Univ. of Pennsylvania, Philadelphia, Pa., Dec.

MOORE, S. A. AND KIMBROUGH, S. O. 1995. Message management systems at work: A system for office communications. *J. Organ. Comput. 5*, 2, 83–100.

POLLACK, A. 1989. Computers that read and analyze. *N.Y. Times* (June 7), D1.

POWELL, W. W. 1990. Neither markets nor hierarchy: Network forms of organization. *Res. Organ. Behav. 12*, 295–336.

RAYMOND, L. AND BERGERON, F. 1996. EDI success in small and medium-sized enterprises: A field study. *J. Organ. Comput. Electron. Commer. 6*, 2, 161–172.

REICH, R. B. 1991. *The Work of Nations: Preparing Ourselves for 21st Century Capitalism*. Knopf, New York.

SALMINEN, A. 1995. EDIFACT for business computers: Has it succeeded? *StandardView 3*, 1 (Mar.), 33–42.

SEARLE, J. R. 1969. *Speech Acts*. Cambridge University Press, Cambridge, U.K.

SEARLE, J. R. 1979. *Expression and Meaning*. Cambridge University Press, Cambridge, U.K.

SEARLE, J. R. AND VANDERVEKEN, D. 1985. *Foundations of Illocutionary Logic*. Cambridge University Press, Cambridge, U.K.

SIVORI, J. R. 1996. Evaluated receipts and settlement at Bell Atlantic. *Commun. ACM 39*, 6 (June), 24–28.

SMELTZER, L. R. 1991. An analysis of strategies for announcing organization-wide change. *Group Organ. Stud. 16*, 1 (Mar.), 5–24.

SWIFT. 1996. S.W.I.F.T. in the treasury markets: Foreign exchange, money markets, derivatives, gold & precious metals. Handbook 1885, Society for Worldwide Interbank Financial Telecommunications, New York, Oct.

SOKOL, P. K. 1995. *From EDI to Electronic Commerce: A Business Initiative*. McGraw-Hill, New York.

SPERBER, D. AND WILSON, D. 1988. *Relevance: Communication and Cognition*. Harvard University Press, Cambridge, Mass.

STARLING, G. 1993. Project management as a language game. *Ind. Manage. Data Syst. 93*, 9, 10–18.

STEEL, K. 1994. Another approach to standardising EDI. *Electron. Mark. 12* (Sept.), 9–10.

STRAUB, D. W., JR. AND WETHERBE, J. C. 1989. Information technologies for the 1990s: An organizational impact perspective. *Commun. ACM 32*, 11 (Nov.), 1328–1339.

STRAWSON, P. F. 1950. On referring. *Mind LIX*, 235, 320–44.

VANDERVEKEN, D. 1990. *Meaning and Speech Acts, Volume I, Principles of Language Use*. Cambridge University Press, Cambridge, U.K.

VAN REIJSWOUD, V. E. 1996. The structure of business communication: Theory, model and application. Ph.D. thesis, Technische Univ. Delft, Delft, The Netherlands, June.

WINOGRAD, T. 1988. Where the action is. *Byte* (Dec.), 256A–258.

WINOGRAD, T. AND FLORES, F. 1987. *Understanding Computers and Cognition: A New Foundation for Design*. Addison-Wesley, Reading, Mass.

YOUNG, S. R. AND HAYES, P. J. 1985. Automatic classification and summarization of banking telexes. In *Proceedings of the 2nd Conference on Artificial Intelligence Applications* (Washington, D.C., Dec. 11–13). IEEE Computer Society Press, Los Alamitos, Calif., 402–408.

Received October 1994; revised December 1996 and January 1997; accepted January 1997