

On Lean Messaging with Unfolding and Unwrapping for Electronic Commerce*

Steven O. Kimbrough Yao-Hua Tan

Abstract

EDI (electronic data interchange) messages are notoriously lean and difficult (or impossible) to interpret without additional information. While recognizing and sympathizing with the many criticisms that have been made of the design of EDI protocols, we argue in this paper that there is something basically correct, even inevitable, in the noted leanness of EDI messages. By way of making this case, we present a framework that describes how interpretation of EDI messages works, and indeed must work. Central elements of this framework are what we call unwrapping and unfolding of messages. We present and discuss these concepts in some detail. Further, we demonstrate how to exploit these concepts in formalizations for electronic commerce. In particular, we show how Kimbrough's lean event semantics for speech acts, and Tan and Thoen's theory of directed obligation can be fit naturally and fruitfully into this framework, and to each other. Much work remains to be done, but the progress in formalization in evidence here should be generalizable.

Keywords: EDI, electronic data interchange, electronic commerce, FLBC, formal language for business communication, deontic logic, logic, business message

*File: lean-sok-tan-si.tex; 1999-10-09, 1999-10-10, 1999-10-11, 1999-10-15, - 16, -17; 1999-11-11, 2000-02-12, 2000-04-12. From: lean-sok-tan-19980324.TEX.

Contents

1	Context	4
2	Lean Messaging with Wrapping	6
2.1	The Basic Schema	6
2.2	Leanness	8
2.3	The Meaning Problem	11
2.4	The Inferencing Problem	13
2.4.1	Interchange Agreement	14
2.4.2	Reference to other Contracts or Procedures	15
2.5	Background Law	16
2.6	Summary So Far	17
3	Meaning Unfolding	19
3.1	Unfolding a simple promise	22
3.2	Unfolding a Simple Purchase Order	27
3.3	An aside on mapping to the world	29
4	Unwrapping	32
5	Conclusion	35
	References	36

List of Figures

1	EDI messaging schema: Message u from speaker s to addressee r	6
2	Generic Quotation Form. After: http://www.entrepreneurmag.com/formnet/	7
3	EDI (X.12) Request for Quotation (line numbers added). From [15].	8
4	Approximate English Translation of Request for Quotation. From [15].	9
5	Generic Purchase Order Form. After: http://www.entrepreneurmag.com/formnet/	10
6	Simple Purchase Order Example	25
7	EDI-like representation of the simple purchase order in Figure 6.	26
8	Meaning unfolding, via lean event semantics, of a simple purchase order, Figures 6 and 7	27

1 Context

Most observers of, and indeed most participants in, EDI would agree that present technology and standards for computer-to-computer exchange of structured information are importantly, even fundamentally, deficient.¹ Costs are too high, the first-trade problem is daunting and discourages SMEs (small- and medium-sized enterprises) from participating in EDI, too much manual intervention is needed even for well-established systems, established EDI protocols are not suited to the Web, there is no semantics for existing protocols (or only incomplete and confused ones), and so on (cf., [15, 19, 33, 2] among many others). All this and more is part of the received wisdom today, and we largely agree.

What to do? The usual move in Information Systems when faced with a puzzling application problem is to structure the application domain so that it can more effectively be computerized. A formal theory, including a formal semantics, would seem to be in order. Why?

1. Computerization requires that the application domain be structured and formalized. This may occur explicitly, in the form of a theory, or implicitly and only in the minds of the programmers. Clearly, the explicit route is preferable.
2. Explicit structuring and formalization allows us in an important sense to know what we are doing. A formal semantics for a domain gives us insight and understanding, which we may use to revise and apply our theory. It allows us to specify and test requirements independently of a particular implementation. For example, a logic model of a purchase order can be examined and tested for the inferences it supports, whether or not there is any implementation. This provides ways of validating specific implementations.
3. Formalization and structuring also facilitate extending our representations and designing rules for generating and processing messages. We shall illustrate this in the sequel. For now, we simply note that structuring and formalization ought to help with interpreting EDI messages but also with the rules and policies that do the interpreting.

¹Regarding *computer-to-computer exchange of structured information*, this can serve as a rough definition of EDI (electronic data interchange).

Again, these are widely-agreed upon thoughts with which we agree. But how to proceed from here? Different people will have—do have—different approaches to structuring and formalizing the EDI domain. Some favor a logic-based approach (e.g., Kimbrough, Moore, Covington, e.g., [3, 4, 5, 14, 15, 16]). Others, one based on XML (e.g., [20, 33, 32]), Petri nets (Lee and his group at EURIDIS [1, 17]), or some other formalism. Even given a formal strategy, there will no doubt be different substantive approaches.

Besides the call for formalization and rigor, are there any other principles that can be invoked to guide these investigations? We think so and our main purpose in writing this paper is to present and argue for certain such principles, which we call *lean messaging with wrapping, unfolding, and unwrapping*. In brief, we believe that existing EDI systems embody these principles already and for good reasons. Even with improved messaging standards they should remain in force. In fact, a main reason for such improvements is to avail EDI systems of the benefits of these principles.

The remainder of the paper is organized as follows. §2 presents a series of observations regarding the EDI scene today. We abstract these observations, focusing throughout on specific examples, and we develop a general framework for them. According to this framework,

1. EDI (and in general business) messages are lean, in a sense we make clear in the sequel.
2. EDI messaging occurs in the context of extensive background information, which is used by senders and receivers to interpret messages and take actions based on them. We call this background information the *wrapper*.
3. Interpreting and acting upon EDI messages is a two-step process:
 - (a) By a process we call *meaning unfolding*, the basic meanings of the lean messages are extracted.
 - (b) By a process we call *unwrapping*, consequences of the unfolded messages are extracted, using contextual information present in the wrapper.

After we develop the general picture, in §2, we discuss unfolding in some specificity and detail in §3, and unwrapping in §4. §5 offers concluding remarks.

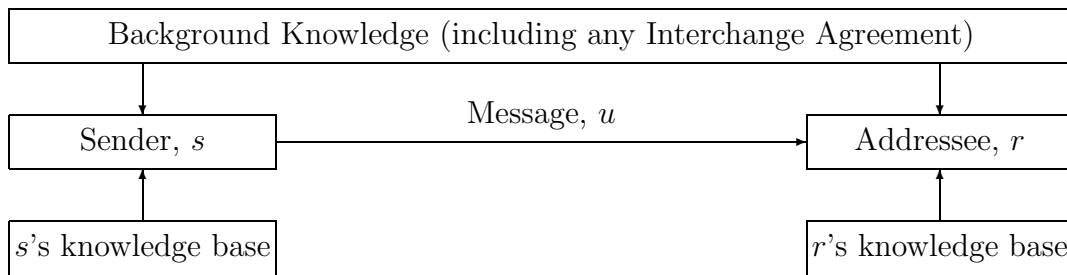


Figure 1: EDI messaging schema: Message u from speaker s to addressee r

2 Lean Messaging with Wrapping

2.1 The Basic Schema

Consider in the abstract how business messaging—and EDI messaging in particular—is effected. See Figure 1, the basic EDI messaging schema. When two organizations, s and r , wish to do business with EDI they typically begin by negotiating an Interchange Agreement. This is a contract which, among other things, specifies rules of trade between s and r , including which EDI protocols and transaction sets are to be used, and how EDI messages are to be interpreted. Once this contract is in place the sender of a message can draw upon its own knowledge bases, Background Knowledge (e.g., relevant laws and administrative rules), and the Interchange Agreement to formulate a specific message, u (utterance) in the Figure. Similarly, the receiver of a message draws upon its own knowledge bases and the Background Knowledge (including the Interchange Agreement) in order to interpret and act upon the incoming message, u .

A simple example—requesting a price quote and responding to one—will serve to focus the discussion, and (without loss of generality) will serve to bear the points we wish to make.

Figure 2, page 7 “Generic Quotation Form,” presents a simple, pre-EDI, paper-oriented quotation form. It can be, and is being, used for conduct of routine business-to-business transactions. It is the sort of thing that EDI protocols represent, with the aim of replacing paper forms with computer-to-computer automated messaging. Actual EDI protocols, fully specified (e.g., in X12 and EDIFACT), are considerably more complex and fraught with options than might be suggested after examination of Figure 2.

TO:		QUOTE DATE			
		YOUR REQUEST NO.			
		OUR QUOTATION NO.			
SHIP TO:		THIS QUOTATION VALID ONLY			
		FROM TO			
DELIVER DATE PROMISED	DELIVERY VIA	FOB	TERMS		
ITEM	QUANTITY ORDERED	DESCRIPTION	UNIT COUNT	UNIT PRICE	TOTAL AMOUNT
SPECIAL NOTES:		DATE	APPROVAL SIGNATURE		
		<p>THANK-YOU FOR REQUESTING A QUOTATION FROM US. WE ARE PLEASED TO PROVIDE YOU WITH THIS INFORMATION.</p> <p>WE THANK YOU FOR YOUR PAST BUSINESS, AND WE LOOK FORWARD TO SERVING YOU IN THE FUTURE.</p>			
*** QUOTATION ***					

Figure 2: Generic Quotation Form.

After: <http://www.entrepreneurmag.com/formnet/>.

[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

Figure 3: EDI (X.12) Request for Quotation (line numbers added). From [15].

Figure 3 shows a valid request for quotation EDI message in the X12 standards. Figure 4 shows its approximate translation into plain English. Finally, Figure 5 shows the inevitable purchase order form. Together, the figures so far, Figures 1–5, will serve as the basic reference examples for the discussion that follows. The simplifications, introduced in these examples for the sake of the discussion, are without loss of generality with regard to the issues to hand.

We now offer some reflections on this basic schema.

2.2 Leanness

By virtue of the messaging schema, §2.1, a certain leanness attends EDI messaging, and indeed business messaging in general. In fact, one of the characteristic features of business messages (including Electronic Data Interchange (EDI)) is their unintelligibility without substantial reference to context. EDI messages present data structures that are basically stripped down, with little or nothing in them explaining the relations among the data elements constituting the message. For example, in the Request for Quotation EDI message, Figure 3, only a code in the header indicates that it is a data structure for representing an RFQ. This is the case generally in both the X12 and EDIFACT standards, as well as the S.W.I.F.T. protocols. It is also true of standard paper forms for doing business; see Figures 2 and 5, pages 2 and 10. The types of information given in these message data structures are, for example, names and addresses of buyer, seller, involved

- [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.

Figure 4: Approximate English Translation of Request for Quotation. From [15].

banks for payment of the goods, payment method, delivery terms, and so forth. But neither in the EDI messages, nor in the standard business forms, is there an explanation how these different rôles relate to each other. For example, the information that the seller is the one who delivers goods to the buyer, and that the buyer is the one who pays for the goods is not (explicitly) represented in the message data structures.

There are several reasons for this minimalist representation of information in EDI (and generally in business-to-business) messages. One is of course efficiency of data transfer. The fewer bits an EDI message contains the less it needs resources. A second reason is that EDI messages are international standards. By only providing the bare data and omitting textual explanation about the relations between the data, the problem of translation from one language, or legal system, to another is reduced. And there is a governing (heuristic) principle here: only send news. What is reasonably constant is treated as background, as context, and belongs to the wrapper; messages are for new information.

EDI (and generally business-to-business messaging) is about much more than mere electronic *transfer* of messages. Computer mediated, automated *processing* of these messages, for sending or upon receipt, is at the heart of the matter. Without automated creation of, and response to, electronic

TO:		CUSTOMER NO.			
		TERMS			
		SALES			
SHIP TO:		SHIP WEEK OF			
		FOB			
		ORDER NO.	DELIVERY VIA	ROUTING	
PLEASE SHIP THE FOLLOWING AS SPECIFIED					
ITEM	QUANTITY ORDERED	DESCRIPTION	UNIT COUNT	UNIT PRICE	TOTAL AMOUNT
SPECIAL INSTRUCTIONS:		TOTAL AMOUNT			
		DATE	APPROVAL SIGNATURE		
		PURCHASER SIGNATURE			
		TITLE			
*** PURCHASE ORDER ***					

Figure 5: Generic Purchase Order Form.
 After: <http://www.entrepreneurmag.com/formnet/>.

messages, there is little point to EDI; one might as well use plain e-mail. By operating leanly, however, by omitting so much information from EDI messages, the understanding and processing of these messages becomes a nontrivial exercise. Humans reading paper forms (cf., Figures 2 and 5) have background knowledge and training that allows them to interpret a message, and know, e.g., that the sender of a purchase order (presumably) expects to receive certain goods and to pay for them in a certain way. Further, a trained person could respond, e.g., to a request for quotation, by ascertaining that the sender should be responded to, by checking with the proper source in house, and by forwarding a reply to the sender (say by filling out a form as in Figure 2).

Automated processing of an EDI message requires that the computer system is able to store in its own database such important information as the name of the sender, the ID of the message, what the sender wants and so on. Automated processing also requires that the computer system can initiate further action. A very simple example is that (in responding to a purchase order) the computer system automatically sends information to the production department that is relevant for the production of the requested goods, sends advanced shipment notification to the carrier for shipment of the goods, and sends an invoice to the buyer after the goods are shipped. Depending on the terms of delivery the computer might even have to arrange insurance of the goods for the transport.

All of this, whether done by people or by computer systems, requires (among other things) that:

1. The meaning of the message to be processed be unpacked and understood by the processor; and
2. The consequences of the message (once its meaning is understood) be inferred by the processor and acted upon.

We call (1) the *meaning problem* and (2) the *inference problem*. And we now discuss each in its own section.

2.3 The Meaning Problem

The meaning problem in its basic forms is also known as the *mapping* problem. At the most elementary level, a message processor must recognize that a certain field in the message corresponds to, *is to be mapped to*, a particular

field in the recipient’s information system. For example, QUOTE DATE in a Quotation form (Figure 2) might need to be mapped to the QTEDATE field in the QUOTES table of the ACCOUNTS database. Also, the *format* of the content of the field, a date, will often have to be mapped as well. In general, an organization’s representation of its business in its data processing system will seldom if ever closely match the representation in a received (or sent) business message. The meaning problem is the problem of translating (or mapping) from one representation to another.

We can think of meaning unfolding as a process (or piece of software) that translates an EDI message into a sentence, or formula that can be read or processed by a human or computer, respectively. The following expression represents this process.

$$\text{MET} : u \mapsto u_{\text{unfolded}} \quad (1)$$

In the expression, the operator (or process) MET embodies a Meaning Extraction Theory. It transforms the input u (normally, an EDI message) to the output u_{unfolded} (typically, an organization’s representation of the meaning in the received message, u). We say that u_{unfolded} is the unfolded message corresponding to u . Either or both, we note, may be semantically and syntactically complex.

With EDI, meaning unfolding articulates the contents of messages. The messages by themselves are hardly intelligible for the untrained human reader, since only the most basic data are represented, and not the (meaning) relations among them. In meaning unfolding, relations are added to the bare data of an EDI message. For example, in an EDI X12 RFQ (cf., Figure 3), data are given about identities of prospective buyer and seller, and about quantities and types of goods purchased, and so on, but it is not explicitly stated that the buyer is responsible for payment for the goods, or that the seller is responsible for shipment. Much information of this sort comes out in the English translation, Figure 4, which we may think of as the output of a MET operator, applied by humans. The character and detailed outputs of meaning unfolding will vary depending on whether the subsequent processing is by humans or by machines, but it must occur in some form no matter what. When people process messages received on paper forms, they must be trained so as to embody an appropriate Meaning Extraction Theory (MET). When the unfolding is done automatically, the MET must be explicitly programmed. In both cases, we emphasize, the MET operator will

rely on: (a) Background Knowledge, and (b) the processing organization's knowledge base (see Figure 1).

An aside to conclude this section.

Existing software that performs the meaning unfolding process—so-called EDI translator software—is widely noted for being inflexible, limited in scope, expensive, and inefficient [26, 27]. Usually, how the mapping is done—how the MET function works—is idiosyncratic, varying from company to company. Specialized message forms from a particular sender are mapped and translated to the particular data processing regime of the receiver. For each EDI supplier a company has, a particular mapping and translator configuration has to be done.

Carefully attending to meaning unfolding of the EDI message can help to reduce the magnitude of this translation problem. If a reasonably general theory of what messages can say (a semantic theory) were to hand, then certain efficiencies in implementation would be possible. Having a description of the meaning of an EDI message would afford valuable help for programmers who have to develop the translator software. Instead of aiming to map a given message form to a particular firm's data processing structure, the message could be mapped to an intermediate representation of high generality and public availability, one with well-known and understood semantics. This will in general be easier than mapping to the particular data processing system of a given firm.² Individual firms would have to map from the general intermediate representation to their specific data processing systems, but this is something that need be done only once (and maintained).

So, representing the meaning of an EDI message explicitly is not only helpful for humans in understanding the message. It also supports the development of the translator software, and in general the EDI application software that is built to process these messages automatically. Full development of these points is, however, beyond the scope of the present paper.

2.4 The Inferencing Problem

In addition to the meaning problem of EDI messages there is second problem, namely the inferencing problem, i.e., the problem of inferring the implications

²To our knowledge, Ken Steel is the first to have made this point forcefully and in print [28]. The point has also been seized upon by the XML community. We note that Steel has not offered a comprehensive semantics for the intermediate form, nor has the XML community demonstrated how a DTD can perform such a function.

of an EDI message. To illustrate what we mean by the inferencing problem, we discuss some examples.

2.4.1 Interchange Agreement

EDI messages are only legally binding between partners who have signed a so-called interchange agreement [27]. Without such an agreement the EDI messages have at best murky legal implications, and hence might not create obligations for agents mentioned in the message. Especially for international trade, the interchange agreement has to stipulate under which country’s jurisdiction the agreement resorts. This greatly affects which legal implications obtain for an EDI message. The legal implications of an EDI message can easily vary from country to country. If a specific EDI message is legally binding in country A, but not in country B, i.e., is non-enforceable in B, then only in country A does the promise stated in the message imply a (legal) obligation.

A simple example will serve to illustrate our point. Suppose that message u is a purchase order, that r honors the purchase order and delivers the goods as requested. Something like the following expression should be inferable:

ϕ :
Buyer B is obligated to pay seller S price P by payment method M for delivery of goods G under terms D.

In order to account for the inference that the message u leads to a condition that is legally binding and obligatory, ϕ , we introduce the notion of a wrapper. The wrapper, which is a process or piece of software that can make such inferences, is typically independent from the EDI message. It contains information, typically of a general sort, that is not stated explicitly in the message. In other words, the wrapper is the theory that enables the right type of inferences from the data in an EDI message.

The following expression—both analogous to and fitting to expression (1)—represents this process.

$$\text{CET} : u_{\text{unfolded}} \mapsto \phi \tag{2}$$

In the expression, the operator (or process) CET embodies a Consequence Extraction Theory. It transforms the input u_{unfolded} (normally, an unfolded EDI message) to the output ϕ (typically, an organization’s representation of

a consequence or action resulting from the received message, u , and other information). We say that ϕ is the unwrapping of the unfolded message u_{unfolded} .

2.4.2 Reference to other Contracts or Procedures

EDI messages often include references to other contracts or procedures. For a proper understanding of the messages these references must also be understood, recognized and acted upon. For example, the Incoterms is a set of clauses for the delivery specifications of purchased goods. An Incoterm determines whether the buyer or seller is responsible for organizing and paying for transport of the goods from the seller's place to the buyer's place (see [7, 9]). According to the Incoterm Free-on-Board (FOB) the seller is responsible for everything concerning the transport of the goods until the goods are shipped on the vessel in the port of origin. This Incoterm becomes effective in practice by simply including the term FOB in the purchase order, often under the slot for delivery terms. An inference process is required to determine which obligations are implied by the term FOB in an EDI purchase order. Let us assume that the meaning of FOB is expressed as follows.

ψ :

The seller is obliged to arrange for the shipped goods export documents, inland transport to the railing of the vessel, and cover all the expenses for this.

Assume that α denotes *Seller S is obliged to buyer B to deliver goods G under delivery terms FOB for price P by payment method M*, then the wrapper should support the following inference with respect to the obligations that are implied by the FOB term in the message.

$$\text{CET} : \alpha \mapsto \psi \tag{3}$$

The Letter of Credit (LC) procedure offers another illustration. For example, in a purchase order message the payment procedure can be specified, e.g., post-payment (payment after delivery of the goods), or pre-payment (payment before delivery of the goods). The letter of credit procedure is a more complicated payment procedure, in which it is stipulated that a local bank is taking over the risk for the seller of payment by the buyer. The LC procedure requires specific documents, such as a bill of lading. In this case

the payment method indicated in the purchase order makes an implicit reference to these other trade documents, which also have to be processed in the right way (see also [1, 17]). Again, some form of CET, based on the wrapper (Background Knowledge plus the organization’s knowledge base; Figure 1) is needed in order to discern consequences of messages and to act upon them.

2.5 Background Law

EDI messages should not contradict national or international contract law. For example, under the United Nations Convention on the International Sale of Goods (CISG), if in a purchase contract the trading partners agree that the goods will be delivered under the Free-on-Board (FOB) Incoterm, the seller is responsible for arranging the proper export documents for the shipment of the goods. However, if his government issued a ban on the transport of this type of goods to the country of the buyer (as used to be issued by the American Government for the export of certain type of computers to the former Soviet Union), then the CISG stipulates that in this case the seller is exempted from this FOB obligation to arrange export documents. In other words, the FOB clause in the purchase contract is overruled by the CISG in special situations (see [7, 9]). Let us assume that the expression *Exportban* represents this information about the export ban issued by the government on the export of the type of goods such as the goods G mentioned in a message, u_{unfolded} . The blocking of the inference can be stated as follows.

$$\text{CET} : u_{\text{unfolded}} \wedge \text{Exportban} \not\rightarrow \psi \quad (4)$$

Here $\not\rightarrow \psi$ indicates that the inference to ψ no longer holds due to the extra information about the export ban. This type of reasoning, where conclusions are lost by adding extra information is called defeasible reasoning (see [15] for early remarks on the use of defeasible reasoning in electronic commerce). One could argue that inferring the legal implications of an EDI message are always defeasible even within a country where the contract is valid, because there might be exceptional circumstances that override these implications.

All these examples clearly show that it is far from obvious which implications can be inferred from a particular EDI message, and hence which obligations hold for the parties mentioned in an EDI message. The CET inference process, using the wrapper, is needed to extract this information from the EDI message. The CET and the wrapper are typically pieces of software that

are independent from the EDI message. They contain background information that is not stated explicitly in the message. This background information could be available via the web. Some prototypes have already been developed for providing this on-line background information. For example, the International Chamber of Commerce (ICC) launched the pilot project Eterms (<http://www.verisign.com/Eterms/index.html>). The Eterms Repository is an on-line database of legal and information security terms, clauses, model agreements, practice statements and related materials that serve the electronic commerce community. The purpose of the project is to develop an accessible body of terms, conditions, and related materials that trading partners will be able to use or refer to in establishing an electronic commerce relationship. Another pilot project is the on-line available Interprocs models of trade procedures (see: <http://abduction.euridis.fbk.eur.nl/projects/weboutline/Web.Outline.html>). Graphical models of trade procedures can be downloaded from this site, and can help merchants to understand which trade procedures apply for their trade. A detailed description of this project can be found in [17].

We note that practically speaking not all aspects of wrapping can be made explicit. Clearly, many of the wrapper examples mentioned above also exist in the traditional paper-based document environment. In paper contracts these issues are not addressed explicitly either. However, in the new electronic environment of electronic commerce, there is incentive to make as much as possible of this implicit information explicit, and thereby available for automated processing. Probably the complete wrapper theory that answers all our questions is infeasible. However, developing the wrapper theory is a kind of incremental process. The broader this theory is, the more likely automated processing can use it.

2.6 Summary So Far

Message unfolding is the generating of a semantic model of an EDI message. Inevitably this requires going to a more detailed level of explanation than is normally the case. It is the wrapping that requires (and rewards) going to a more abstract level, allowing a message to get different interpretations depending on the legal system and other relevant background information by which it is processed. The difference between meaning unfolding and unwrapping can be illustrated as follows. Meaning unfolding of an EDI purchase order is adding, for example, the information that the buyer is responsible for

payment of the goods. However, this responsibility only implies a legal obligation for the buyer if the purchase order is legally binding, which depends on the interchange agreement. And this legal inferencing has its basis in the wrapping of the EDI message. The basic idea is that the obligations do not automatically follow from the meaning of an EDI message; they follow from the message and the wrapper. The need for meaning unfolding and unwrapping of EDI messages also explains the observation that electronic commerce is much more than just exchanging EDI messages.

To sum up, EDI messages are lean in the sense that only a minimal, or core meaning, tends to be expressed. The larger, or extended, meaning of the message can only be determined in light of the governing wrapper theory, which includes the interchange agreement and relevant background laws. Thus, to interpret a message, a recipient will first unfold its core meaning, and subsequently use the wrapper theory to make action-relevant inferences. The following points summarize the argument so far.

1. The EDI (and generally business-to-business) messaging schema, Figure 1, describes the essential context of EDI messaging (for present purposes).

Here, we focus on the problem faced by r , the recipient of a message, u , in interpreting u and taking appropriate action. What we have to say will apply straightforwardly to the problems faced by s , the sender of the message, who must figure out what the message should say.

2. Figures 2–5 present representative business messages, either as EDI messages or as paper-based forms.
3. Recipient r has available contextual information that it can use in interpreting and responding to the message u . We call this information the *wrapper* (for r). In terms of the basic messaging schema, Figure 1, r 's wrapper consists of the Background Knowledge (including any Interchange Agreement) and r 's knowledge base.
4. Whether r is a person or a computer system, r 's processing the message requires that:
 - (a) The meaning of the message to be processed be unpacked and understood by the processor.

We call this the *meaning* problem. It is handled by applying a Meaning Extraction Theory (MET) to the message, thereby yielding a (meaning) unfolded expression. In symbols:

$$\text{MET}:u \mapsto u_{\text{unfolded}} \quad (5)$$

We call this process *unfolding*.

- (b) The consequences of the message (once its meaning is made explicit and understood) be inferred by the processor and acted upon.

We call this the *inference problem*. It is handled by applying a Consequence Extraction Theory (CET) to the unfolded message, thereby yielding a collection of direct consequences, u_{cons} (think of it as a conjunction of possibly many expressions), which r 's processing system can record and directly act upon. In symbols:

$$\text{CET}:u_{\text{unfolded}} \mapsto u_{\text{cons}} \quad (6)$$

We call this process *unwrapping*.

5. Both the MET and the CET inevitably draw upon the wrapper. This is true whether processing of u (the message) is automated or done by humans. Context must be added in order to interpret and act upon a received message.
6. The architecture described in the previous points allows—and actual business practice (including EDI messaging systems) realizes—a certain leanness in the design of messages. The message (cf., Figures 2–5) is abstracted from its underlying meaning (cf., Figure 8). This permits message formats to remain constant, while the meanings they convey are changed by context.

This completes our overview of the points we wish to emphasize. We now turn to the task of providing more concrete illustrations.

3 Meaning Unfolding

We begin with two distinctions. The logical theory we offer will be sensitive to, and will make use of, both.

First, we distinguish two sorts of meaning: core meaning and extended meaning. This distinction—alluded to above—is similar to the distinction made in speech act theory between the locutionary act and the illocutionary act. There, a distinction has been accepted between the sentence meaning (locutionary act) and the speaker’s meaning (illocutionary act), although the two meanings may coincide. For example, the sentence meaning of “Do you know what time it is?” (an asking of the addressee whether the addressee is aware of the hour) is different from the usual speaker meaning (a request to be told the time). We recognize such a distinction in ordinary language, when we say things like “What she said was X but what she meant was Y.”

As we have noted, EDI messages have certain core meanings (in virtue of the standards to which they conform)³ as well as additional meanings that are unpacked (unwrapped and unfolded) in light of the the wrapper (governing interchange agreement, etc.).

Our second distinction is a three-fold one. For statements, or messages, we distinguish three levels of representation:

³Of course, we are idealizing a bit. In fact one of the purposes of the interchange agreements is to specify what has been left vague by the standards.

1. Informal: Typically in natural language.

Example:

“All politicians are liars.” (7)

2. Formal: Syntactically formal, but not fully logical (e.g., without formal rules of inference, or a formal semantics).

Example (for representing “All politicians are liars,” i.e., (7) \mapsto (8)):

(politician and liar)(x) (8)

3. Logical: Syntactically and semantically formal, i.e., in a logic and with a formal semantics. Example (for representing “All politicians are liars,” i.e., (7) \mapsto (9) and (8) \mapsto (9)):

$\forall x(P(x) \rightarrow L(x))$ (9)

Now, if machines are to process messages effectively, the messages cannot be expressed informally, as in Expression (7). Instead, formal expressions, using an appropriate and convenient syntax, as for example in Expression (8), are required. EDI messages, when they do not include free text, are formal, but not logical. Important for the success of a formal messaging syntax, as in Expression (8), is a theory of—semantics for—what the messaging syntax means. This is provided (in outline) by Expression (9) for Expression (8).

Notice further that Expression (9) is really an *unfolding* of the meaning of Expression (8). It articulates, more precisely and in more detail, what Expression (8) means. Why? Because Expression (8) is (we assume) expressed in a formal syntax without accompanying logic and/or semantics, while Expression (9) *is* represented in logic and has (we assume) an appropriate accompanying semantics (i.e., the standard one). Of course, it would be possible to unfold and articulate the meaning of expressions like (8) without advertent to formal logic, but there is little reason to do so. Formal logic is well developed theoretically and presents something of a gold standard for formal clarity. That is fundamentally why we use it for present purposes.

Our point is *not* that formal logic expressions, such as Expression (9), should be used instead of syntactically convenient expressions, such as Expression (8), for business-to-business communication. Instead, our point is

that a theory of unfolding—with a semantics, as in (9)—is valuable for understanding how to design and validate effective automation systems for electronic commerce, and that formal logic is an appropriate vehicle for expressing such a theory. For example, something a proper logic gives us is an automatable way of exploring the inferences that can or cannot be made from a collection of statements. Especially for the sort of complex and subtle reasoning we need to support in electronic commerce, e.g., reasoning about commitments and obligations, this should be valuable both for specifying and validating software systems (working, e.g., with syntactically formal EDI expressions) and for facilitating the construction (dare we hope automatically?) of interchange agreements and other wrapping elements.

3.1 Unfolding a simple promise

Meaning unfolding is the articulation of the basic semantics for an expression. Using the example of promising, we may think of meaning unfolding as follows. Let us say that at the informal level of representation we have:

At time t, s promises r that s will deliver to r goods g within 30 days.

(10)

Using an EDI-like message notation, this might be expressed formally (but not logically):

1. promise : 12345
2. date-time : 1999-09-23
3. from : s
4. to : r
5. deliver
 - (a) goods : g
 - (b) to : r
 - (c) by : s
 - (d) date-time : 1999-09-23 + day(1999-09-23 + 30)

(11)

So, we should think of expression (11) as an instance of u in Figure 1.

At our level 3, logical representation, we can represent this promise as follows using Kimbrough’s lean event semantics [14]:

$$\begin{aligned} & \text{promise}(12345) \wedge \text{Speaker}(12345, s) \wedge \text{Addressee}(12345, r) \wedge \\ & \text{Cul}(12345, 1999-09-23) \wedge \Box(\top \rightarrow (K(12345) \leftrightarrow (\text{deliver}(e) \wedge \\ & \text{Agent}(e, s) \wedge \text{Benefactive}(e, r) \wedge \text{Sake}(e, 12345) \wedge \text{Theme}(e, g) \\ & \wedge \text{Cul}(e, t) \wedge \leq(t, +(1999-09-23, \text{day}(1999-09-23 + 30)))))) \end{aligned}$$

(12)

Comments and points arising:

1. Expression (12) reads roughly as “12345 is a promise by s to r , occurring on September 23, 1999. The promise 12345 concerns a delivery event e . Necessarily, this promise is kept if and only s effects a delivery to r of g , and this is for the sake of the promise 12345. In addition, the delivery, e must occur (for the promise 12345 to be kept) within thirty days of September 23, 1999.”
2. This reading is also (we assume) a correct interpretation of the EDI-like message, Expression (11) and of the original ordinary language-like utterance, Expression (10).
3. \Box in Expression 12 is a modal logic necessity operator. For any sentence, P , read $\Box P$ as “Necessarily, P ” or “It is necessary that P .” This necessity operator is required for technical reasons, which for present purposes we elide. It is generally safe simply to drop the operator, \Box . Doing this yields the *extensional approximation*, and for practical purposes the approximation suffices.
4. Unlike Expression (11), Expression (12) is logical and fully formal. It is also the unfolded version of Expression (11). Note in this regard how Expression (12) makes explicit so much that would otherwise have to be read into or assumed regarding Expressions (10) and (11).
5. The predicates in Expression (12) all happen to be from a (here unspecified) controlled vocabulary, and would be used in unfolding very many types of primary messages.

6. There is nothing special about promising that limits the scope of this approach to representing messages. The move in evidence here generalizes to the other illocutionary forces, such as asserting, requesting, and declaring. Although further details are required, the analysis is remarkably simple. Even so, it is able to capture the core logical behavior of various illocutionary forces. See [10, 11, 12, 13, 14].
7. \top , appearing in Expression (12), is shorthand for any logical tautology. It always evaluates to true. Anything conditioned on \top is thus only vacuously conditioned. If I say I will go to the store whether or not it rains, my going is but vacuously conditioned on the weather.
8. Notice that the meaning, as analysed, implies nothing about whether certain actions are obliged by a promise. Promising is one thing, keeping a promise is another. We believe that this in fact accords well with the leanness of EDI messages, noted and discussed above. For example, it standardly belongs to the meaning of a purchase order that the seller delivers goods to the buyer and the buyer pays the seller, but this is only *obligatory* for these parties if additional assumptions are made. Making these assumptions is normally part of what the interchange agreement (the wrapper) is about. In this respect our analysis deviates from the standard view, or *central dogma*, of speech acts, exemplified in the work of Searle and Vanderveken [23, 24, 25, 31] among others. Searle and Vanderveken assume that a speech act of promising automatically implies the obligation to fulfill the promise. However, in our view this implication critically depends on the context. For some contexts it might hold and not for others. This inference is performed by using the wrapper, which we discuss in §4. See [14] for a more detailed discussion of this point.

TO: r		CUSTOMER NO. s			
		TERMS 30-days-net			
SHIP TO: an-address		SALES			
		SHIP WEEK OF 1999-12-03			
		FOB customer			
ORDER NO. 54321	DELIVERY VIA		ROUTING		
PLEASE SHIP THE FOLLOWING AS SPECIFIED					
ITEM	QUANTITY ORDERED	DESCRIPTION	UNIT COUNT	UNIT PRICE	TOTAL AMOUNT
catid-32-9	box	copier paper	6	\$45.01	\$270.06
catid-35-9	dozen	no. 2 pencil	3	\$1.99	\$5.97
SPECIAL INSTRUCTIONS:		TOTAL AMOUNT			\$276.03
		DATE	APPROVAL SIGNATURE		
		1999-11-19			
		PURCHASER SIGNATURE			
		TITLE			
*** PURCHASE ORDER ***					

Figure 6: Simple Purchase Order Example

1. purchase-order : 54321
2. date : 1999-11-19
3. from : s
4. to : r
5. ship-to: an-address
6. terms: 30-days-net
7. ship-week-of: 1999-12-03
8. FOB: customer
9. grand-total: 276.03
10. deliver
 - (a) goods : catid-32-9
 - (b) unit: box
 - (c) description: "copier paper"
 - (d) quantity: 6
 - (e) unit-price: 45.01
 - (f) line-total: 270.06
11. deliver
 - (a) goods : catid-35-9
 - (b) unit: dozen
 - (c) description: "no. 2 pencil"
 - (d) quantity: 3
 - (e) unit-price: 1.99
 - (f) line-total: 5.97

Figure 7: EDI-like representation of the simple purchase order in Figure 6.

3.2 Unfolding a Simple Purchase Order

Figures 6 and 7 show, respectively, a simple example of a specific purchase order and a representation of it in an EDI-like (structured, not logical) syntax. We have kept the names of data items short, and the number of data items few, for the sake of convenience. The example is rich enough to bear the points we wish to make, and sparse enough to avoid burdening the reader with unnecessary detail. Actual purchase orders, and the EDI protocols for them, are of course much more complex. Full treatment of such examples is beyond the scope and purposes of this paper.

$$\text{purchase-order}(54321) \wedge \text{Speaker}(54321, s) \wedge \text{Addressee}(54321, r) \wedge \text{Cul}(54321, 1999-11-19) \wedge$$

$$\square((\top \rightarrow (H(54321) \leftrightarrow$$

$$\begin{aligned} & ((\text{ship}(e1) \wedge \text{Agent}(e1, r) \wedge \text{Benefactive}(e1, s) \wedge \text{to}(e1, \text{an-} \\ & \text{address}) \wedge \text{Theme}(e1, \text{catid-32-9}) \wedge \text{Sake}(e1, 54321) \wedge \text{unit}(e1, \\ & \text{catid-32-9, box}) \wedge \text{description}(\text{catid-32-9, "copier paper"}) \wedge \\ & \text{quantity}(e1, \text{catid-32-9, 6}) \wedge \text{Comc}(e1, t1) \wedge \text{FOB}(e1, \text{customer}) \\ & \wedge \text{during}(t1, \text{week-of}(1999-12-03))) \wedge \end{aligned}$$

$$\begin{aligned} & (\text{ship}(e2) \wedge \text{Agent}(e2, r) \wedge \text{Benefactive}(e2, s) \wedge \text{to}(e2, \text{an-} \\ & \text{address}) \wedge \text{Theme}(e2, \text{catid-35-9}) \wedge \text{Sake}(e2, 54321) \wedge \text{unit}(e2, \\ & \text{catid-35-9, dozen}) \wedge \text{description}(\text{catid-35-9, "no. 2 pencil"}) \wedge \\ & \text{quantity}(e2, \text{catid-35-9, 3}) \wedge \text{Comc}(e2, t2) \wedge \text{FOB}(e2, \text{customer}) \\ & \wedge \text{during}(t2, \text{week-of}(1999-12-03))) \wedge \end{aligned}$$

$$(H(54321) \rightarrow (K(54321) \leftrightarrow$$

$$\begin{aligned} & (\text{pay}(e3) \wedge \text{Agent}(e3, s) \wedge \text{Benefactive}(e3, r) \wedge \text{Sake}(e3, [e1, \\ & e2]) \wedge \text{Theme}(e3, p3) \wedge \text{unit}(e3, p3, \$) \wedge \text{quantity}(e3, p3, q) \\ & \wedge \text{unitprice}(e1, \text{catid-32-9, p1}) \wedge \text{unitprice}(e2, \text{catid-35-9, p2}) \wedge \\ & \text{unit}(e1, p1, \$) \wedge \text{unit}(e2, p2, \$) \wedge \text{quantity}(e1, p1, 45.01) \wedge \\ & \text{quantity}(e2, p3, 1.99) \wedge =(p3, +(*(45.01, 6), *(1.99, 3))) \wedge \\ & \text{Cul}(e3, t3) \wedge \text{TERMS}(e3, 30\text{-days-net})) \wedge \end{aligned}$$

Figure 8: Meaning unfolding, via lean event semantics, of a simple purchase order, Figures 6 and 7

Figure 8 displays the results (the u_{unfolded}) of using lean event semantics to apply a reasonable MET (meaning extraction theory) to the u of Figures 6 and 7. Although there is very much to say about this example, we limit ourselves to a few remarks, aimed at making the example accessible and its connection with our points clear.

1. The expression in Figure 8 is a fully formal, logical *model* (and hence, approximation) of what we think would typically be meant by the purchase order shown in Figure 6.
2. As before:
 - (a) Predicates beginning with a capital letter (*Speaker*, *Theme*, *Comc*, etc.) are generic reserved words in lean event semantics;
 - (b) Predicates in lower case (e.g., *deliver*, *pay*) are (here) verbs whose meaning is specified independently (more on this below); and
 - (c) Predicates in all upper case (e.g., *FOB*, *TERMS*) are domain-specific.
3. The expression in Figure 8 is a single logical conjunction of the three main blocks of code.
4. The gist of the first block is that this is a purchase order from s to r happening (“culminating,” *Cul*) on 1999-11-19, regarding (having *Theme* of) three events: $e1$, $e2$, $e3$.
5. The gist of the second block is to indicate that this purchase order is making a request (of r by s). The request is honored (H) if and only if two shipping events occur, $e1$ and $e2$. Descriptions of these events follow and match to the original purchase order. Note that in both cases, the shipping events are (requested to) begin (“commencing”, *Comc*) during the period indicated. Nothing is said about when delivery should occur.
6. The gist of the third block is that if the afore-described request is in fact honored, then this purchase order is also making a promise. The promise is kept (K) if and only if $e3$ is a paying event with the properties described.

7. Note that the unfolded expression is fully logical and supports inferencing. For example, suppose the purchase order is issued and in fact r honors it fully (ships the goods as, when, and where described). Suppose further that s fails pay r as described. It follows, logically, that the promise associated with the purchase order, made by s , has not been kept.
8. Suppose, indeed, that the promise has been broken. In order to extract those consequences we need a CET and a look at the wrapper. We turn to that in the next full section.

3.3 An aside on mapping to the world

No semantics can ever be entirely formal. At some point we need to map between the formal symbols used in the formal language, and the system that language is to model. The trick is to give this mapping at a very basic level, and then rely as much as possible on composition of symbols to provide additional meaning. With our logical semantics (e.g., Figure 8) we need to provide a basic mapping for: terms (s , r , etc.), functions (e.g., *week-of*), and predicates (e.g., *ship*, *pay*). Terms are rather straightforward: one makes a catalog, a sort of table, in which the referring expression is mapped to what it refers to. For example, s might be the customer number of the firm Nadir, Inc. Functions are also straightforward: one either maps them, much as terms are mapped, or one defines them by composition from more primitive functions.

Predicates, we have noted, are of three kinds:

1. Specific for lean event semantics, e.g., *Theme*, *Comc*, *Unit*
2. Generic, e.g., *ship*, *pay*
3. Domain specific, e.g., *FOB*, *TERMS*

The specific predicates should be few in number and can be defined (mapped) explicitly. See the previous discussion on Incoterms. The generic predicates can and should be mapped to a clear, broadly-accepted, and public specification. Language having the importance it does, such specifications are indeed available. WordNet, an electronic lexicon produced over a number of years at Princeton University, is an excellent example [6]. Let us see, briefly, what

WordNet has to say about the three generic (verb) predicates in Figure 8: *purchase-order*, *ship*, *pay*.

WordNet recognizes *purchase order* as a noun, but not as verb. Nonetheless, what it says is useful. The following passage, as well as all the subsequently quoted passages, is from WordNet 1.6.

```
The noun purchase order has 1 sense (no senses from tagged texts)
1. {04902219} <noun.communication> order1#7, purchase order#1 --
(a commercial document used to request someone to supply
something in return for payment; "IBM received an order for
a hundred computers")
```

Notice that a purchase order here—and in Figure 8—involves both a request for something and a (promised) payment for it.

WordNet 1.6 recognizes *ship* as a verb for which there is exactly one sense: to transport commercially.

1 sense of ship

Sense 1

```
{01328437} <verb.motion> transport1#4, send#4, ship#1 --
(transport commercially)
=> {01328337} <verb.motion> barge1#2 --
    (transport by barge on a body of water)
=> {01331285} <verb.motion> dispatch#1, despatch#1,
    send off#1 -- (send off promptly)
=> {01331167} <verb.motion> bundle off#1 --
    (send off unceremoniously)
=> {01331450} <verb.motion> route2#1 --
    (send documents or materials to appropriate
    destinations)
=> {01331576} <verb.motion> forward#1, send on#1 --
    (send or ship onward from an intermediate post or
    station in transit; "forward my mail")
```

More refined meanings are available with different verbs. We may thus tentatively identify the *ship* of Figure 8 with *ship#1* of WordNet.

Finally, WordNet 1.6 recognizes 11 distinct meanings for *pay* as a verb.

The verb *pay* has 11 senses (first 11 from tagged texts)

```
1. {01540968} <verb.possession> pay#1 --
```

- (give money in exchange for goods or services;
 "I paid four dollars for this sandwich";
 "Pay the waitress, please")
2. {00718708} <verb.communication> give1#5, pay#2 --
 (convey, as of a compliment, regards, attention, etc.;
 bestow; "Don't pay him any mind"; "give the orders";
 "Give him my best regards"; "pay attention")
 3. {01541614} <verb.possession> pay up#1, ante up#1, pay4#3 --
 (cancel or discharge a debt; "pay up, please!")
 4. {01542031} <verb.possession> pay2#4, pay off4#4, make up#3,
 compensate2#4 -- (do or give something to somebody in
 return; "Does she pay you for the work you are doing?")
 5. {01695538} <verb.social> pay#5 --
 (render; "pay a visit"; "pay a call")
 6. {00500874} <verb.cognition> pay3#6 --
 (bear (a cost or penalty), in recompense for some action;
 "You'll pay for this!"; "She had to pay the penalty for
 speaking out rashly"; "You'll pay for this opinion later")
 7. {01566906} <verb.possession> yield#10, pay1#7, bear1#8 --
 (bring in; as of investments; "interest-bearing accounts";
 "How much does this savings certificate pay annually?")
 8. {01869530} <verb.stative> pay#8 --
 (be worth it; "It pays to go through the trouble")
 9. {00496485} <verb.cognition> give2#10, pay#9, devote#2 --
 (as in the expressions "give thought to";
 "give priority to", etc.)
 10. {01541783} <verb.possession> pay14#10 --
 (discharge or settle; "pay a debt"; "pay an obligation")
 11. {01600647} <verb.possession> pay13#11 --
 (make a compensation for; "a favor that cannot
 be paid back")

Pretty straightforwardly, we can tentatively identify *pay* with *pay#1*.

These verb mappings to WordNet happen to be unproblematic. Even so, we note that some degree of clarification has been gained. More importantly, nothing like this is available without some form of explicit meaning unfolding.

We now turn to how the wrapper may be used to extract further consequences from business messages.

4 Unwrapping

For reasons mentioned above, illocutionary, deontic, and other consequences of an EDI message typically have to be inferred. These inferences might also be defeasible [18, 21, 22]. Think of modeling the legal implications of a contract. Basically these are not represented in the EDI message, but can be inferred from the EDI message(s) and a background theory (which belongs to the wrapper). A typical example is to infer whether a promise really implies an obligation, or in more legal terms to infer what the legal status is of an EDI message. All of this might vary from country to country. The wrapper is country dependent. These are large and important issues, and they cannot be resolved here in a short paper. But we can make a start.

Our suggestion is that the necessary inferences attending to EDI messages are presently achieved through a system of lean messaging with subsequent unfolding and unwrapping, relying on the wrapper. Further, we suggest there are good reasons to mirror this strategy in new protocols and in the development of a formal semantics for EDI, and we have offered (a sketch of) an account of how to represent the lean semantics (the unfolding) of an EDI message. Our aim now is to indicate how to formalize part of the wrapping so that required additional inferences—the unwrapping and extraction of consequences—can be made automatically. We focus on directed obligation by way of example.

Modeling directed obligation is particularly interesting and challenging. Standard deontic logics are about obligations *tout court*. For business purposes, among others, we need a concept of, and hence a formalization of, directed obligation in which one agent has an obligation *to* another agent to do something. In the case of our purchase order, if r delivers, then s not only has to (is obliged to) pay for the goods, but has to pay r in particular. How to model this has only recently begun to be investigated. Initial results were not fully satisfactory (see [8]). Recently, however, Tan’s group has developed an analysis and a logic for directed obligation (see [30, 29]). The central formula

$${}_i\mathcal{O}_j^S({}_i\mathcal{E}\alpha) \stackrel{\text{def}}{=} (\neg_i\mathcal{E}\alpha \Rightarrow_S {}_j\mathcal{P}({}_j\mathcal{E}la_i)) \quad (13)$$

represents their analysis of directed obligation and should be read as “Under legal system S , i is obligated to j that i brings about $[\mathcal{E}]$ state α in the world *means* if i fails to bring about α then under system S it would be that j is permitted $[\mathcal{P}]$ to bring legal action $[la]$ against i .” By way of explanation,

the operators ${}_i\mathcal{O}_j^S$, ${}_i\mathcal{E}$ and ${}_i\mathcal{P}$ are modal operators. The operator ${}_i\mathcal{O}_j^S$ is a deontic operator that expresses that agent i is obliged towards j under legal system S to bring something about. The operator ${}_i\mathcal{E}$ is a so-called *action operator* that expresses that agent i brings something about. The operator ${}_i\mathcal{P}$ is again a deontic operator for which we assume the axioms of standard deontic logic (for technical details see [30, 29]). The operator ${}_i\mathcal{O}_j^S$ is defined in terms of the two other operators ${}_i\mathcal{E}$ and ${}_i\mathcal{P}$. There are two fundamental differences between ${}_i\mathcal{O}_j^S$ and a standard deontic logic operator \mathcal{O} . First, ${}_i\mathcal{O}_j^S$ represents a directed obligation between two specific agents i and j , whereas \mathcal{O} is completely anonymous. Secondly, ${}_i\mathcal{O}_j^S$ defines in purely objective and operational terms what will happen if the obligation is violated. For example, if a buyer has the obligation towards the seller to bring about payment of the delivered goods, then the seller has the permission to bring about legal action against the buyer if he does not pay. Symbolically, we have:

$${}_b\mathcal{O}_s^S({}_b\mathcal{E}paying(e)) \stackrel{\text{def}}{=} (\neg{}_b\mathcal{E}paying(e) \Rightarrow_S {}_s\mathcal{P}({}_s\mathcal{E}la_b)) \quad (14)$$

where b is the buyer and s the seller.

The operational character of this definition gives not only a detailed explanation of what it means to have a contractual obligation between two trading partners, but it makes the notion of directed obligation also applicable to artificial agents, because it is only the operational behaviour of the agent that matters and not his internal mental states, which artificial agents probably lack.

With this definition (and attendant logic) we can write *bridge laws* or *meaning postulates* or *Protokolsätze* that connect the language of messaging with the language of directed obligation. Here are two examples:

$$\begin{aligned} &\forall e, e', s, r && (15) \\ &(\text{purchase-order}(e) \wedge \text{Speaker}(e, s) \wedge \text{Addressee}(e, r) \wedge \\ &\text{accept}(e') \wedge \text{Agent}(e', r) \wedge \text{Theme}(e', e)) \\ &\rightarrow (\neg{}_r\mathcal{E}H(e) \Rightarrow_S {}_s\mathcal{P}({}_s\mathcal{E}la_r)) \end{aligned}$$

If you get a PO and you accept it (i.e., you send a message in which you say $\text{accept}(e') \wedge \text{Agent}(e', r) \wedge \text{Theme}(e', e)$), if you don't honor it (deliver the goods), then the sender of the PO can initiate legal action against you.

$$\begin{aligned}
& \forall e, s, r & (16) \\
& (\textit{purchase-order}(e) \wedge \textit{Speaker}(e, s) \wedge \textit{Addressee}(e, r) \wedge \\
& H(e)) \rightarrow (\neg_s \mathcal{E}K(e) \Rightarrow_s {}_r \mathcal{P}({}_r \mathcal{E}la_s))
\end{aligned}$$

If you send a PO and it is honored (the goods are delivered) and you don't pay, then the receiver of the PO can bring legal action against you.

Expression (16) shows clearly the technical difference between meaning unfolding and wrapping. The predicate $K(e)$ is an essential part of the unfolding of the purchase order (Figure 8), but in this unfolding nothing is said yet about obligations. Nothing in Figure 8 says that $K(e)$ is obliged! To express this deontic aspect we have to add the definition of directed obligation ($\neg_s \mathcal{E}K(e) \Rightarrow_s {}_r \mathcal{P}({}_r \mathcal{E}la_s)$). Note that this adding of rules regarding obligations, of course, critically depends on whether the purchase order is legally binding according to an interchange agreement.

Comments:

1. Lean events semantics theory has paid off handsomely here. Notice that the bridge laws ((15) and (16)) are completely general and use only the controlled vocabulary (verb predicates, thematic rôle predicates (from the controlled vocabulary of lean event semantics), and predicates from the logic of directed obligation).
2. The illocutionary auxiliary predicates (here H for “is honored” and K for “is kept”) of Kimbrough’s lean events theory of speech acts have a natural and essential place in the example bridge laws, allowing them to be general.
3. The previous point leads one to wonder whether lean events semantics theory might be used to analyze further the logic of directed obligation. We suspect so, but that is a subject for another paper.
4. Much remains to be done if wrappers for electronic commerce are to be substantially formalized and given a full semantics. That must be a subject for many subsequent papers, but we see no fundamental impediments to completing such a programme of research.

5 Conclusion

What we have presented here is an argument for a way of thinking about how transactions work in electronic commerce (lean messaging in the context of wrappers, meaning unfolding, representation with lean events semantics, unwrapping to extract inferences, directed obligation theory, and so on). We have observed that under existing arrangements EDI messaging is lean, messages are unfolded (in the industry lingo, “mapped” to the receiver’s data processing system), and then unwrapped (in the industry lingo, “processed”). Indeed, this arrangement, so far as it goes, seems right, even inevitable: messages carry news, wrappers hold context.

What we have presented in addition to this basic framework is a demonstration of how two particular theories—Kimbrough’s lean events semantics, and Tan and Thoen’s theory of directed obligation—can be used to account for unfolding and unwrapping. These two theories bring considerable virtues to the table. They are both logical theories and come with proof theories and meaningful semantics. Both, in virtue of being broadly logical, can be mapped to ordinary language in a foundational way, per the discussion in §3.3. And they can be made to work together, as we have shown with example bridge laws.

It is significant, however, that both theories—lean events and directed obligation—are of recent origin and must be said to be still under development. Neither theory may turn out to be the best in its class for modeling messages or wrappers. Only future research can settle the issue. That there is an issue and that it is clearly articulated is a main contribution of this paper.

References

- [1] BONS, R., LEE, R., AND WAGENAAR, R. Computer-aided auditing of inter-organizational trade procedures. *Intelligent Systems in Accounting, Finance and Management* 6, 2 (1997), 29–46.
- [2] COMMERCE.NET. Projects & pilots.
<http://www.commerce.net/projects/currentprojects/eco/>. Accessed 1999-07-12.
- [3] COVINGTON, M. A. Toward a new type of language for electronic commerce. In *Proceedings of the Twenty-Ninth Annual Hawaii International Conference on System Sciences, Vol. IV, Information Systems—Organizational Systems and Technology* (Los Alamitos, CA, 1996), J. Nunamaker, Jr. and R. Sprague, Jr., Eds., IEEE Press, pp. 329–336.
- [4] COVINGTON, M. A. Speech acts in electronic communication, with special reference to KQML and ANSI X12. In *Proceedings of the Thirtieth Annual Hawaii International Conference on System Sciences* (Los Alamitos, CA, 1997), J. F. Nunamaker, Jr. and R. H. Sprague, Jr., Eds., IEEE Press, pp. 410–419.
- [5] COVINGTON, M. A. Speech acts, electronic commerce, and KQML. *Decision Support Systems* 22 (1998), 203–211.
- [6] FELLBAUM, C., Ed. *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, MA, 1998. ISBN: 0-262-06197-X.
- [7] FOEKENS, A., MITRAKAS, A., AND TAN, Y.-H. Facilitating international electronic commerce by formalizing the Incoterms. In *Proceedings of the Thirtieth Annual Hawaii International Conference on System Sciences* (January 1997), J. F. Nunamaker, Jr. and R. H. Sprague, Jr., Eds., HICSS'97, IEEE Computer Society Press, pp. 459–467.
- [8] HERRESTAD, H., AND KROGH, C. Obligations directed from bearers to counterparties. In *Proceedings of the Fifth International Conference on Artificial Intelligence and Law* (New York, NY, 1995), ICAIL'95, ACM Press, pp. 210–218.
- [9] INTERNATIONAL CHAMBER OF COMMERCE (ICC). *Guide to the Incoterms 2000*. Publication N620E, ICC Publishing, Paris, 1999.

- [10] KIMBROUGH, S. O. On representation schemes for promising electronically. *Decision Support Systems* 6, 2 (1990), 99–122.
- [11] KIMBROUGH, S. O. On electronic commerce, subatomic semantics and Cæsar’s stabbing. In *Proceedings of the Thirtieth Hawaii International Conference on System Sciences* (Los Alamitos. CA, 1997), R. H. Sprague, Jr., Ed., IEEE Press, pp. 361–370.
- [12] KIMBROUGH, S. O. On $ES\Theta$ theory and the logic of the X12 date/time qualifiers. In *Proceedings of the Thirty-First Hawai’i International Conference on System Sciences* (Los Alamitos. CA, 1998), R. H. Sprague, Jr., Ed., IEEE Press, pp. 330–339.
- [13] KIMBROUGH, S. O. Sketch of a basic theory for a formal language for business communication. In *Proceedings of the Thirty-First Hawai’i International Conference on System Sciences* (Los Alamitos. CA, 1998), R. H. Sprague, Jr., Ed., IEEE Press, pp. 717–725.
- [14] KIMBROUGH, S. O. Formal language for business communication: Sketch of a basic theory. *International Journal of Electronic Commerce* 3, 2 (Winter 1998–99), 23–44.
- [15] KIMBROUGH, S. O., AND MOORE, S. A. On obligation, time, and defeasibility in systems for electronic commerce. In *Proceedings of the Twenty-Sixth Annual Hawaii International Conference on System Sciences, Volume III, Information Systems: DSS/Knowledge-Based Systems* (Los Alamitos, CA, 1993), J. F. Nunamaker, Jr. and R. H. Sprague, Jr., Eds., IEEE Computer Society Press, pp. 493–502.
- [16] KIMBROUGH, S. O., AND MOORE, S. A. On automated message processing in electronic commerce and work support systems: Speech act theory and expressive felicity. *ACM Transactions on Information Systems* 15, 4 (October 1997), 321–367.
- [17] LEE, R. M. Distributed electronic trade scenarios: Representation, design, prototyping. *International Journal of Electronic Commerce* 3, 2 (1999), 105–136.
- [18] LEE, R. M., AND RYU, Y. U. DX: A deontic expert system. *Journal of Management Information Systems* 12, 1 (1995), 145–169.

- [19] LEHMANN, F. Machine-negotiated, ontology-based EDI (electronic data interchange). In *Electronic Commerce: Current Research Issues and Applications*, N. R. Adam and Y. Yesha, Eds., vol. 1028 of *Lecture Notes in Computer Science*. Springer, Berlin, Germany, 1996, pp. 27–46.
- [20] LEVENTHAL, M., LEWIS, D., AND FUCHS, M. *Designing XML Internet Applications*. Prentice Hall PTR, Upper Saddle River, NJ 07458, 1998.
- [21] NUTE, D., Ed. *Defeasible Deontic Logic: Essays in Nonmonotonic Normative Reasoning*. Kluwer Academic Publishers, Dordrecht, Holland, 1997.
- [22] RYU, Y. U., AND LEE, R. M. Defeasible deontic reasoning and its applications to normative systems. *Decision Support Systems* 14 (1995), 59–73.
- [23] SEARLE, J. R. *Speech Acts*. Cambridge University Press, Cambridge, England, 1969.
- [24] SEARLE, J. R. *Expression and Meaning*. Cambridge University Press, Cambridge, England, 1979.
- [25] SEARLE, J. R., AND VANDERVEKEN, D. *Foundations of Illocutionary Logic*. Cambridge University Press, Cambridge, England, 1985.
- [26] SIVORI, J. R. Evaluated receipts and settlement at Bell Atlantic. *Communications of the ACM* 39, 6 (June 1996), 24–28.
- [27] SOKOL, P. K. *From EDI to Electronic Commerce: A Business Initiative*. McGraw-Hill, Inc., New York, NY, 1995.
- [28] STEEL, K. Another approach to standardising EDI. *Electronic Markets* 12 (1994), 34–47.
- [29] TAN, Y.-H., AND THOEN, W. A logical model of transfer of obligations in trade contracts. *Journal of Accounting, Management and Information Systems* 8 (1998), 23–38.
- [30] TAN, Y.-H., AND THOEN, W. Modeling the dynamics of transferable obligations in business procedures. In *Proceedings of the Thirtieth*

Annual Hawai'i International Conference on System Sciences (January 1997), J. F. Nunamaker, Jr. and R. H. Sprague, Jr., Eds., IEEE Press, pp. 494–501.

- [31] VANDERVEKEN, D. *Meaning and Speech Acts, Volume I, Principles of Language Use*. Cambridge University Press, Cambridge, England, 1990.
- [32] VEO SYSTEMS. Conducting business on the internet: Using technology to cross organizational boundaries. http://www.veosys.com/xml/white_papers/whitepapers.html. Accessed 1999-07-12.
- [33] XML/EDI GROUP, T. Home page for the XML/EDI Group. Web page, January 1998. <http://www.xmledi.net>.