CERIAS Tech Report 2000-29

COMPLEX EVENTS IN AN ONTOLOGICAL-SEMANTIC NATURAL LANGUAGE PROCESSING SYSTEM

by Craig McDonough

Center for Education and Research in
Information Assurance and Security,
Purdue University, West Lafayette, IN 47907-2086

COMPLEX EVENTS IN

AN ONTOLOGICAL-SEMANTIC

NATURAL LANGUAGE PROCESSING SYSTEM

A Thesis

Submitted to the Faculty

of

Purdue University

by

Craig J. McDonough

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2000

To Meghan—

for being here from the beginning;

to Maddy—

for opening the study door, happy-faced, at all the right times; and,

very much mostest of all,

to Wendy—

for your never-bending belief in me.

ACKNOWLEDGMENTS

First, I would like to thank my thesis advisor, Dr. Victor Raskin for so many things—for encouraging me to research in the field of natural language processing, for suggesting a dissertation on complex events in the first place, for helping me see the important things, for eliminating so much red tape. For these, and other things, I owe Dr. Raskin a debt of gratitude I can only hope to repay.

I would also like to thank Dr. Sergei Nirenburg for serving on my committee and for providing me access to the Mikrokosmos Knowledge Acquisition Editor, without which this dissertation could never have been completed. I also appreciate the opportunity, granted by Dr. Nirenburg, to spend time at New Mexico State University's Computing Research Laboratory, which confirmed for me that he is a man of great vision.

My thanks go also to Dr. Mikhail Atallah and Dr. Myrdene Anderson, for serving on my committee and for graciously allowing this dissertation to take on a much different shape than my proposal could have predicted.

Finally, I would like to thank my father-in-law, Dr. Joseph Waldron, for helping me navigate the path that he himself has taken.

TABLE OF CONTENTS

Page

Page

ABSTRACT

McDonough, Craig J. Ph.D., Purdue University, May, 2000. Complex Events in an Ontological-Semantic Natural Language Processing System. Major Professor: Dr. Victor Raskin.

The goal of this dissertation is to elucidate principles for representing complex-event knowledge (or “scripts”) for use in an ontological-semantic natural language processing system, specifically the Mikrokosmos system. Complex events are, simply, events that comprise other events. A shorthand example is: the event of buying may comprise picking out merchandise, bringing it to the cash register, offering money to the cashier, receiving change, and leaving the store with the merchandise. As previous research has shown, texts make widespread use of such “world knowledge” by leaving many events implicit, relying on the hearer/reader to infer this information within the discourse context. The challenge for natural language processing programs is to construct a model of this world knowledge to fill in these gaps. Previous programs armed with such knowledge, e.g., SAM (Cullingford 1978) and Ms. Malaprop (Charniak 1977), have made some advances, but several design problems prevented real progress. It is argued in this dissertation that the ontological-semantic paradigm generates semantic descriptions of texts rich enough to make use of complex-event knowledge, thereby eliminating one barrier to implementation of such knowledge. Furthermore, this dissertation develops specific modifications to the ontological-semantic system that enable representations of complex-event knowledge to: (i) represent, in principle, any sequence of "real-world" events, (ii) achieve broad conceptual coverage, (iii) discern "fine-grained" differences in conceptual information, and (iv) significantly reduce redundancy in knowledge representation. Several constructed complex-event descriptions are adduced as evidence of feasibility of the formalism. Though this knowledge has not yet been implemented, the suggestions made in this dissertation should prove to be a significant step in representing the immense body of complex-event knowledge for natural language processing systems.

CHAPTER ONE:
THE ONTOLOGICAL-SEMANTIC APPROACH TO NATURAL LANGUAGE PROCESSING

The goal of this dissertation is to elucidate principles for representing 'complex-event' knowledge ("scripts," in Schankian terms; cf. Schank and Abelson 1977) for use in an ontological-semantic natural language processing system, specifically the Mikrokosmos system, currently being developed jointly at New Mexico State University's Computing Research Laboratory and Purdue University's Natural Language Processing Laboratory. Complex events are, in this dissertation and in ontological semantics, events that comprise other events. A shorthand example is: the event of buying may be made up of picking out merchandise, bringing it to the counter/cash register, offering the money to the cashier, receiving change, and leaving the store with the merchandise.

Unlike other dissertations addressing complex events/scripts (e.g., Cullingford 1978, Wilensky 1980, 1981, DeJong 1979, Miikkulainen 1993), this dissertation does not attempt the two-fold task of theorizing a solution and then building a program to implement the theory. Rather, since a system is already in place, this dissertation focuses on developing a theory of complex events for this system. Furthermore, also unlike previous research on the topic, this dissertationt builds complex-event knowledge specific enough and sensitive enough to be used in general-purpose language understanding; that is, the principles and guidelines developed herein are meant to be applicable to the world, not just to a "micro-world," (cf. Minsky 1975, a sublanguage (cf. Raskin 1971, Kittredge 1987), or domain (cf. Winograd 1976 [1972]). The bulk of early work on complex events/scripts was done in the mid-1970s through the 1980s at the Yale University Artificial Intelligence lab. While research there focused demonstrating the feasibility of the knowledge-engineering approach, this dissertation takes for granted that such an approach is viable, in the context of an ontological-semantic paradigm.

1.1 Introduction

This dissertation first describes the architecture of the ontology and the advantages this model affords for natural language processing (NLP) tasks, specifically the representation of complex-event knowledge. Then, after a review of the previous literature on the representation of complex events (Chapter Two), this dissertation then discusses theoretical issues in the representation of complex events for an ontological-semantic NLP system (Chapter Three) and data in support for the theory (Chapter Four). Chapter Five provides a summary of the findings of this thesis and makes specific recommendations for the vast amount of work still ahead.

1.2 World knowledge in natural language processing systems

This section is concerned, broadly, with the boundary between word knowledge and world knowledge, the extent to which world knowledge is required, and how world knowledge should be represented in a natural language processing system. The motivation for this section is two-fold. First, knowledge of complex events (or, “scripts”) is generally considered to fall into the class of knowledge called “world knowledge.” It is therefore helpful to define and delimit the scope of world knowledge. Second, ontological semantics takes a novel approach to the boundary between word and world knowledge—one that is different from both the semantics/pragmatics distinction in linguistics and the (usual) lexicon/world-knowledge-base distinction in NLP.

1.2.1 Motivating the need for world knowledge

Since Bar-Hillel’s (1960) scathing, but accurate, critique of extant machine translation (MT) systems, the theory that *any* natural language understanding system concerned with “the determination of the specific meaning in context of a word” (*Ibid*. 159) will require knowledge about the world in order to understand a natural language text has gained wide acceptance. Minsky (1968), Schank and Abelson (1977), Hayes (1985); Carlson and Nirenburg (1990), and Allen (1995) make similar claims in computational linguistics; Raskin (1985) and Nirenburg (1986) address the same issue on the interface of computational linguistics with semantic theory.

Bar-Hillel’s (1960: 158) example is elegant in its simplicity: assigning the correct meaning to (1), for instance, will be impossible unless the understanding system is equipped with knowledge about the world, e.g., the relative sizes of objects.

(1) The box was in the pen

Determining “prepositional-phrase-attachment” also often requires world knowledge, as in the following (adapted from Allen 1995: 392):

(2)
(a) I read a story about evolution within the last ten minutes.
(b) I read a story about evolution within the last million years.

Here, knowledge about the relative default durations of evolution and reading events is needed to assign a meaning to the prepositional phrase.

Other phenomena generally considered to fall within the scope of “world knowledge” include: conceptual information (e.g., Mahesh 1996, Lenat and Guha 1990), facts and instances of concepts (e.g., Mahesh 1996), scripts/complex events (e.g., Schank and Abelson 1977, Cullingford 1978, Carlson and Nirenburg 1990), discourse models (e.g., , Charniak 1979, 1986, Golden and Rumelhart 1993, Rumelhart 1994), speaker goals and plans (Schank and Abelson 1977, Wilensky 1980), presupposition presuppositions (cf. Levinson 1983), speech acts (Austin 1962 [1975], Searle 1972), conversational implicature (Grice 1975), time (Allen 1984), and visual cues (e.g., Minsky 1975, Winograd 1976 [1972]).

1.2.2 Defining ‘world knowledge’

Before proceeding any further, however, a definition of “world knowledge” is called for. One might, at first, naively suggest that world knowledge is that which must augment the linguistic interpretation of a text to arrive at a full semantic interpretation of that text. World knowledge, on this view, becomes “implicit” knowledge, while textual information becomes “explicit” knowledge. This definition will, in fact, be adequate for a linguistic theory (e.g., a theory of the semantics-pragmatics interface), which is concerned with sentences in isolation and which is not concerned with formalizing the body of world knowledge. From a natural language processing perspective, however, such a definition of world knowledge suffers from two serious shortcomings (a third shortcoming—that world knowledge can only be discovered by analyzing extant texts—is rather trivial): (i) it cannot account for the fact that "implicit" information in one text (or one word sense) may be "explicit" in another; and, (ii) it presupposes an understander (a human) of word meaning that is not available for an NLP system.

This first problem is the difficult question of determining the boundary between word and world: as Boguraev (1989: 8) notes, while speaking for the field of NLP, "it is hard to pinpoint a boundary between the semantic knowledge that the use of a partiulcar word (sense) implies and the expert background which prompts its use in a specific domain." Such a difficulty requires an NLP system either to duplicate information both in a lexicon and in a world-knowledge base or to force as much information as possible into the lexicon or world-knowledge base. The second problem points out that even if a word/world boundary could be drawn, this would most likely take advantage of the perceptual and cognitive abilities of a human understander.

Ontological semantics, as practiced in the Mikrokosmos system, makes a theoretical distinction between conceptual entities that exist in the world and the natural language markers (lexemes) that are used to make reference to those conceptual entities. The former composes the ontology (i.e., world-knowledge base), while the latter composes the lexical knowledge base (LKB). (See Onyshkevych and Nirenburg 1995, Mahesh 1996, Viegas *et al*. 1999, Viegas 1999). The LKB expresses syntactic/morphological information for lexical items, as well as mappings to the ontology. Such a theoretical choice for the division of labor simplifies greatly the determination of a word/world boundary and is clearly superior in multilingual situations, since language-specific lexicons may be mapped to the same ontology (cf. Nirenburg and Raskin, to appear). However, at the same time, this choice engenders a significant burden on the ontology, the repository of world knowledge. This burden is the subject of the next subsection.

1.2.3 The extent to which world knowledge is required

The answer to, 'To what extent is world knowledge required?', changes depending on one's goals for an NLP system. Faced with the immensity of the task of constructing a world-knowledge base, many NLP systems have focused on a more manageable subset of this knowledge and have thus taken one of two paths: (i) attempt a very coarse-grained analysis of the input text (cf. Schank 1972, Wilks 1986); or, (ii) focus on constructing world knowledge for only a limited domain. The "blocks world" of Winograd's (1976 [1972]) SHRDLU program and Cullingford's (1978) SAM, which covers three stereotypical situations, are examples of the latter. Though outside the usual scope of NLP, it should be noted that expert systems, designed to analyze/impart specific "expert knowledge" in limited domains, also face the difficulty of representing world knowledge. Again, the focus in such systems is on representing only a small subset of what might be included as world knowledge: e.g., medical diagnoses, weather reports, and computer software troubleshooting.

For a general-purpose language understanding system, like the one being developed at the CRL for a large variety of NLP systems, however, the above-mentioned successes in knowledge representation turn out to be in large part unusable, as the restricted-domain world-knowledge created for these systems is not easily extendable to address wider domains. The problem, as Dreyfus (1985: 76) notes, is that representing the totality of world knowledge does not lend itself to "chunking" into sub-worlds, since even in these sub-worlds, in order to be useful and extendable, "the everyday world would have to be included already." (Similar views had been expressed much earlier in semantic theory: cf. Katz and Fodor 1963). As an example, Dreyfus cites Simon's (1977) criticism of SHRDLU (Winograd 1976 [1972]): "SHRDLU's test of whether something is owned is simply whether it is tagged 'owned.' There is no intensional test of ownership, hence SHRDLU knows what it owns, but doesn't understand what it is to own something" (Simon 1977: 1064, qtd. in Dreyfus 1985: 76).

1.2.4 Representing world knowledge in ontological semantics

The world knowledge constructed for the Mikrokosmos system fulfills two necessary criteria for general-purpose language understanding: (i) conceptual entities are defined and distinguished in terms of the world *in toto*; and, (ii) the framework is compatible with the representation of all types of world knowledge. Representation of one type of world knowledge, complex events, is the subject of this dissertation.

Hayes (1979) has argued that other knowledge representation formalisms, such as semantic networks and frame systems, are no more powerful than a first-order predicate calculus (FOPC). Though this may be so, it is intuitively appealing both to link conceptual entities together based on certain properites (via a semantic network) and to chunk knowledge about a particular conceptual entity (via a frame). Furthermore, representation using frames or semantic networks may be more *computationally effective* (cf. Shastri 1981), i.e., allow quicker access to knowledge, than representation in FOPC.

World knowledge, in the Mikrokosmos system, is represented using a semantic network/frame system hybrid. Sections 1.2 through 1.4 detail the representation of world knowledge in this system and how this knowledge contributes to the other modules of the system.

1.3 Motivating an ontology for natural language processing applications

1.3.1 Ontology defined

An ontology is a "constructed world model" (Nirenburg and Raskin, to appear), "domain conceptualization" (Fikes *et al*. 1997), or "set of different classes of objects[1] by which we classify the world" (Allen 1995: 231). That is, an ontology is a repository of conceptual entities held (importantly, by other humans) to be in force for some domain (e.g., all, medical diagnosis, weather reports, etc.). These conceptual entities serve as a grounding for lexical items in a given natural language (cf. Mahesh and Nirenburg 1995).

A natural next question might be, What grounds the conceptual entities in the ontology? In contrast to other approaches—most notably Conceptual Dependency (Schank 1972, 1975) and Preference Semantics (Wilks 1986)— that attempted to isolate a handful of conceptual primitives, the Mikrokosmos ontology provides a richly interconnected hierarchy of concepts defined in terms of other concepts. (There are, of

[1] I.e., "entities." The Mikrokosmos ontology uses "object" in a specialized sense. (See section 1.4.3.)

course, several meta-ontological predicates that help define relationships between concepts; see Mahesh 1996.) Nirenburg *et al*. (1995) has argued that no such handful of conceptual primitives is available; furthermore, they argue that this is not damaging to the theory of ontology, since one might even question the grounding of so-called conceptual primitives.

1.3.2 Philosophical issues

There are bound to be philosophical objections to an enterprise whose chief objectives are specifying the conceptual entities available to a human perceiver and representing commonsense knowledge. Nirenburg *et al*. (1995) has argued against several of these objections (e.g., that ontologies are not language-independent and are not reproducible). In this section, counterarguments to two other possible objections relevant to this dissertation are put forth: (i) that knowledge of complex events is too variable and too speaker-dependent to be generalized for an ontology, and (ii) that development of complex event knowledge is too expensive, in terms of both engineering and computation.

1.3.2.1 Generalizing about knowledge of complex events

One problem that has plagued NLP throughout its history is that words resist simple definitions. Instead, a lack of clear boundaries between what does and does not fall within the scope of a word's referent renders a simplistic set theory account nearly useless (cf. Viegas *et al*. 1999). NLP systems concerned with discourse phenomena, as well as the entire field of pragmatics (the study of language use in context), have also had to address the difficulties arising from the variability of knowledge shared (or unshared) between speakers and hearers in a particular discourse context (cf. Kempson 1997). The same might also be said of attempts to formalize knowledge of complex events: there is too much variability, from situation to situation and from speaker to speaker, for such an enterprise to succeed.

This is an empirical question: if it can be demonstrated that at least a significant subset of this knowledge can be represented in complex events built for an NLP system, then the argument against generalizability has been effectively countered. Schank and Abelson (1977) were the first to argue that such knowledge could be built. (Subsequent experimental research in cognitive psychology confirmed that script-type knowledge does generalize over a wide range of speakers: cf. McKoon *et al.* 1989 and Hudson *et al*. 1992). Chapter Four of this dissertation provides further evidence that complex-event knowledge representation is tractable for a computing system.

1.3.2.2 The expensiveness of building complex events

Even those researching within the ontological semantics paradigm agree that the expense of building complex event knowledge is not worth the payoff for some types of NLP systems, e.g., machine translation (Nirenburg and Raskin, to appear; Mahesh 1996). Indeed the cost of representing all of the stereotypical sequences known to a person seems staggering at first blush—so much so that no one has offered more than a tentative suggestion as to how much knowledge might need to be represented. Hayes (1985: 1), for example, in discussing the "formalization of our knowledge of the everyday *physical* world" [emph. mine] states that "In 1978, I predicted that the overall task was an order of magnitude (but not ten orders of magnitude) more difficult than any that had been undertaken so far. I now think that two or three orders of magnitude is a better estimate." One might argue that representing stereotypical event sequences is at least as complex.

However, it is undeniable that the payoff of representing complex event knowledge would be considerable. This is a first counterargument, however naïve it may be. Second, and again perhaps naïve, there appears to be no other way to achieve this desideratum without encoding this vast amount of knowledge. (Non-representational approaches, such as connectionist models are still woefully unable to cope with such high-level rule abstractions.) A third, and more theoretically sound, counterargument in favor of constructing complex event knowledge is that the ontological-semantic framework allows this knowledge to be implemented incrementally. Each complex event that is constructed is situated in a comprehensive ontology and thus conforms to expectations in the *world* (not simply an isolated "micro-world"; cf. Dreyfus' 1985 criticisms of SAM (Cullingford 1978) and SHRDLU (Winograd 1976 [1972])). Therefore, with the exception of minor revisions that are typical of the knowledge representation enterprise, the addition of new complex events does not affect the status of previously constructed complex events.

1.4 The structure of the Mikrokosmos ontology

In this section we turn to a description of the Mikrokosmos ontology, developed and maintained at the Computing Research Laboratory at New Mexico State University, as well as Purdue University. Because this thesis describes the implementation of complex events ("scripts," in Schank and Abelson 1977) specifically for the Mikrokosmos system, we devote particular attention here to its structure.

1.4.1 The ontology and its relationships with other modules

The Mikrokosmos ontology is a discrete module containing knowledge about conceptual (semantic) "things" existing in the world, as well as remembered instances of these things. The ontology is non-domain-specific and includes a model of the physical world, represented as a tangled subsumption hierarchy. Included as remembered instances are an *onomasticon* of names of things (e.g., people, country names) and a fact database that holds propositions such as the GNP of South Africa and the 1998 winner of the World Cup. The ontology also contains knowledge about discourse participants (attitudes, goals, etc.) and discourse situations (speech acts, scripts, etc.); see Nirenburg and Raskin, to appear). The information contained in the ontology is input to the other modules in an ontological-semantic NLP system.

Ontological concepts are the atoms used for *text meaning representations* (TMRs), which are discussed in detail in section 1.5. Because the Mikrokosmos system was designed to be usable for knowledge-based machine translation, the ontology contains language-independent information. The sharing of knowledge between tasks, whether translation or understanding, is therefore facilitated. The lexicon is a repository for language-specific information concerning the relationship between lexemes and ontological information; there is therefore no one-to-one mapping from lexemes to concepts. Thus, the German word "Taube" will point to both the PIGEON and DOVE concepts, while the English "pigeon" will point to PIGEON, and "dove" will point to

DOVE. The lexicon, then, notates how a particular natural language demarcates meanings among conceptual entities. It should be mentioned that although ontological concepts are given English-word names, this fact has no consequences for the ontology whatsoever, since concept names are not interpretable by the computer system. English-word names, instead of random numbers, are used for ease of ontology maintenance and development.

In, for example, an analysis (i.e., understanding or translation) of an input natural language text, morphological and syntactic analysis of the text yield an output to the semantic modules (the lexicon and ontology). The ontology may then provide feedback to the morphological/syntactic analysis for disambiguation. For example, if syntactic analysis yields equally viable parses based on the part-of-speech ambiguity of one word, and if the semantic module contains information that can identify the correct part-of-speech, then the semantic module can return this output to the syntactic analyzer for further disambiguation.

1.4.2 Knowledge representation

The vast majority of current NLP systems employ some form of symbolic representation. Although a few connectionist (i.e., subsymbolic) systems are in development, e.g., Miikkulainen's (1993; 1995) DISCERN, none have exhibited near the capacity for the high-level rule abstraction that symbolic systems have. DISCERN, for example, understands stories no more complex than Cullingford's (1978) SAM system.

Conceptual knowledge in ontological semantics is represented in terms of frames. Though a detailed description of frame representation, as contrasted with Schankian scripts, is presented in Chapter Two, we present here a short introduction to the definition of the term as it is used in this dissertation (and, indeed, how this definition is implemented in the Mikrokosmos system). Two points, in particular, deserve mention here: (i) the term "frame" has been extended to have any number of implications for knowledge representation (only some of which are in force in this dissertation); and, (ii) frame representation allows certain possibilities that, say, knowledge representation in FOPC does not.

Minsky (1975) introduced the term *frame* as a way of organizing knowledge, in the context of a theory for computer vision. As will be noted in Chapter Two, the terms frame and *script* have, in the intervening time, become to a large extent conflated (for good reason, however). Moreover, just four years after Minsky's paper, Hayes (1979: 46) found it necessary to assert that what is meant by a frame already "has become even looser" and that "It is not at all clear now what frames are, or were ever intended to be." In the original conception, a frame is knowledge structure under which other knowledge structures could be organized. So, for example, the frame for ROOM might include such knowledge structures as furniture, walls, ceilings, decorations, and even function. These slots could take a specified range of fillers, where the semantics of these will be specified by the system. Thus, in for example:

(3) room

purpose	*sleep*
furniture	*bed*, *dresser*
walls	*left, right, far, near*
ceiling	*yes*
decoration	*painting, mirror*

the "furniture" slot may be defined as taking a "contains X" slot, whereas the "walls" slot may be defined as a "part-of" relation and the ceiling slot may take a simple Boolean yes/no filler. These are the essential characteristics of the frame representation in Mikrokosmos. Specifically, the choice of frames for knowledge representation in this system has not been influenced by an particular theory of conceptual entities. (The structure of the ontology, however, has.)

The second point we make here is that frames and formal logic differ less in their expressive power than they do in the computational effectiveness of their architectures. Both are merely formalisms and thus require a semantic theory, in order to understand natural language. For instance, the predicates and variables of formal logic and the frames, slots, and fillers of a frame system require that these be mapped to lexemes in actual natural languages; without a semantic theory, both remain uninterpretable symbol systems. In principle, any inheritance hierarchy (one of the usual features of a frame representation) built into a frame system is expressible in a formal language, for example by defining the "is-a" predicate and constructing lists.

1.4.3 The architecture of the ontology

All ontological concepts, represented as frames, are classified into the categories of OBJECT, EVENT, and/or PROPERTY[2]; these concepts are descended directly from ALL in the subsumption hierarchy. Each of these three second-tier conceptual classifications are further subdivided: OBJECT into PHYSICAL-OBJECT, MENTAL-OBJECT, and SOCIAL-OBJECT; EVENT into PHYSICAL-EVENT, MENTAL-EVENT, and SOCIAL-EVENT; and, PROPERTY into ATTRIBUTE and RELATION. The eight third-tier concepts are further subdivided, and so on. The maximum depth of the hierarchy is currently 14 levels and is

[2] Throughout this dissertation, actual ontological concept names will be formatted in small caps in order to distinguish them from English words and real-world entities referred to by these concepts.

a function of two ontological-development heuristics: (i) broad-coverage over all possible concepts is desired, and (ii) grain-size should be sufficient for the task that the ontology is used for.

1.4.3.1 OBJECTs, EVENTs, and PROPERTYs

In general, OBJECTs correspond to "nouny" things, EVENTs to "verby" things, and PROPERTYs to "adjectivy" things. However, given both the flexibility of human language and the architecture of the ontology, two issues concerning redundancies arise. First, while natural languages regularly allow events to be nominalized (e.g., "They *destroyed* the city" vs. "Their *destruction* of the city"), both verbs and nominalized verbs are mapped to ontological EVENTs, thereby reinforcing the equivalence of such expressions. Second, redundancy between OBJECTs and PROPERTYs and between EVENTs and PROPERTYs is built into the ontology. In a case where conceptual relationships among two OBJECTs need to be expressed, e.g., "the headquarters of a corporation" (Mahesh 1996: 42), one OBJECT can be represented as a PROPERTY of the other OBJECT, as in:[3]

(4) Frame: CORPORATION

Slot	Facet	Filler
HEADQUARTERS	Sem	PLACE

Here, HEADQUARTERS is listed in the ontology only as a PROPERTY, not as an OBJECT. Cases where a concept exists both as a OBJECT and a PROPERTY do exist, e.g., the OBJECT concept EARTH-RESOURCE and the PROPERTY concept HAS-EARTH-RESOURCE. This contrasts with the representation of *partonomies* (part-whole relationships), e.g., "ceiling of a room," which are represented using the HAS-PARTS slot, as in the following example:

[3] Facets are explained in section 1.4.3.3.2.

(5) Frame: ROOM

Slot	Facet	Filler
HAS-PARTS	Sem	CEILING, WALL, [etc.]

Duality between EVENTs and PROPERTYs is also maintained by the ontology; for example:

(6) Frame: AUTOMOBILE

Slot	Facet	Filler
OPERATED-BY	Sem	HUMAN

where OPERATED-BY, although referring to an event, instead is specified as a PROPERTY of an OBJECT frame.

1.4.3.2 Inheritance

The ontology maintains a strict system of "is-a" inheritance; that is, when a concept A descended from a concept B, regardless of whether there are intervening levels between the two, A will exhibit the following characteristics with respect to B: (i) all properties ("slots" in Mikrokosmos terminology) of B will be properties of A; and, (ii) a range-value of any property of A that is also expressed as a property of B will be either as restrictive or more restrictive than the corresponding range-value for the property in A. Thus, as an example of (i), the concept MAMMAL, descended from the concept VERTEBRATE, will inherit the properties of VERTEBRATE. And, as an example of (ii), the concept MAMMAL takes the filler PLACE for the LOCATION property (i.e., a mammal can be located at any place), while the concept SEA-MAMMAL, which is descended of MAMMAL, restricts the value of the LOCATION property to BAY, OCEAN, or SEA.

A special filler, *nothing*, may, however, be used to block inheritance. If a child-concept inherits from a parent-concept a slot that is irrelevant to the child-concept, the slot in the child-concept may be filled with *nothing*, thereby, in effect, removing the slot from the child-concept (and any children of the child-concept). Although widespread use of this feature would defeat the purpose of having an inheritance hierarchy, in practice the *nothing* filler is rarely used (see Mahesh 1996).

The ontology, while predominantly a linear hierarchy, maintains the possibility of multiple inheritance. Thus, a concept a may be descended from both concepts X and Y. When this is the case, the concept a will be a conjunction all properties of both X and Y. Given this strong constraint, multiple inheritance are rare in the ontology.

1.4.3.3 Frame notation

1.4.3.3.1 Frames

Each ontological concept is represented as a *frame* (Minsky 1975). As knowledge within the frame is linked, through certain relationships, with other frames, the ontology resembles a semantic net. For example, all concept frames—with the exception of ALL—are specified for an IS-A relationship, which locates the particular frame in the hierarchy.

1.4.3.3.2 Slots and facets

The fundamental mechanism for expressing relationships between concepts in the ontology is the *slot*. An unbounded number of slots, which are themselves (except for a few special cases) PROPERTY concepts expressed in the ontology, are specified for each frame in the ontology. For example, the concept CORPORATION—whose frame is especially rich, since a primary early task of the Mikrokosmos ontology was to understand/translate texts on corporate mergers—currently has 35 slots, including

PRODUCER-OF, HAS-CUSTOMER, and NATIONALITY. The average frame in the ontology contains approximately 10 slots.

The large majority of slots are PROPERTYs, themselves ontological concepts specified in the ontology. Besides these "content"-filled slots, there are two other classes of slots that deserve mention here. First, several "meta-ontological" slots are available, such as:

- DEFINITION: for recording a dictionary-type definition of a concept;
- TIME-STAMP: for recording when a frame was last updated;
- IS-A: for specifying the parent-concept for the concept in question;
- SUBCLASSES: for specifying child(ren)-concepts for the concept in question;
- INVERSE: for specifying "mirror-image" relationships between PROPERTYs, e.g., ADDRESS vs. ADDRESS-OF.

These meta-ontological slots are used in ontology development and maintenance. DEFINITION and TIME-STAMP play no role whatsoever in computational processes, while IS-A, SUBCLASSES, and INVERSE express structural (i.e., hierarchical) relationships between concepts in the ontology but are not themselves defined as concepts in the ontology.

Second, there are 11 slots designating semantic case-roles (i.e., roughly, predicate arguments), which are only available for EVENT concepts:

- AGENT: the entity that causes or is responsible for an action. (the subject in a transitive sentence is often, but not always, the agent);
- THEME: the entity whose state or location is being described, or whose state is affected by an action (direct object of an action; subject in an intransitive sentence);
- EXPERIENCER: the entity that undergoes psychological experience (perception, cognition);
- BENEFICIARY: the entity that benefits from an action;
- SOURCE: conceptual places where various types of movement and transfer start (used for direction in verbs of motion);
- DESTINATION: an endpoint for actions & processes involving change of location, transfer;

- LOCATION: the place where an event takes place or where an object exists;
- PATH: the route along which an entity (e.g., a THEME) travels;
- CO-THEME: an entity whose state is described in relation to another;
- ACCOMPANIER: an entity which joins the AGENT in the event, but is not the initiator of the event;
- DEGREE: the extent to which something occurs or is done.

These slots will play an important role in the representation of complex-event knowledge.

Whereas slots can be understood as specifying a domain for a variable, *facets* specify a type of variable for each slot domain. There are five facets (a sixth, Salience, is used only in the lexicon to denote a slot's centrality or importance):[4]

- Value: denotes that the filler is an actual value (see section 1.4.3.2.3);
- Sem: denotes that the filler serves as a selection restriction of the slot (i.e., is criterial for that slot);
- Default: denotes that the filler is a value, in the absence of any explicit value (i.e., is assumed for that slot;
- Measuring-Unit: denotes that the filler serves as a domain for a number listed in either a Value, Default, or Sem facet, one of which will also be specified for the slot;
- Relaxable-to: denotes that a usual Sem selection restriction may, in certain cases, be relaxed to account for metaphor and metonymy.

1.4.3.3.3 Fillers

A *filler* denotes the acceptable range of a slot. Although the large majority of fillers are concepts in the ontology, there are two other types of fillers: (i) a non-interpretable string used only for ontology maintenance/development; and, (ii) an actual value for a variable. An example of (i) is the DEFINITION slot, which takes a Value facet and a short dictionary-type definition, in English. For example, the definition of CORPORATION, "A single company or a group of companies organized for a certain business purpose," has absolutely no effect on computational processes. Examples of (ii)

[4] In Chapter Four, a new facet, Optional, will be proposed.

actually are of two types: (a) a numerical value, e.g., "29" for the slot AGE; and, (b) an *instance* of a concept, e.g., BILL-CLINTON-1, an instantiation of the concept HUMAN, which may, for example, fill the AGENT slot in some EVENT frame. (Instances are not represented in the ontology but rather in the onomasticon; see the following section.)

1.4.4 Concepts vs. instances

Whereas a concept is a variable providing a range for a slot-filler, an instance is an instantiation of a variable. Mahesh (1996: 59) provides two heuristics for deciding whether a given symbol is a concept or an instance:

(7)
(a) Can the symbol have its own instance? If yes, it is a concept. If no, it is an instance.
(b) Does the symbol have a fixed position in time and/or space? If yes, it is an instance. If no, it is a concept.

Notice that, technically, an instance indeed may have its own instance: e.g., BILL-CLINTON-AGE18 is an instance of BILL-CLINTON-1, itself an instance of the concept HUMAN. Although proceeding in this manner, of course, leads to an infinite regression of instances, the question here is that in the case of

(8) Concept: HUMAN. Instance-of Concept: BILL-CLINTON-1. Instance-of Instance: BILL-CLINTON-AGE18,

is an instance of a concept an instance or a concept? This appears to be where the analogy of "concept = variable" and "instance = value" breaks down. However, while this may be an interesting philosophical question, it has no bearing on the performance of the system.

All fillers in the ontology are concepts, i.e., variables, with only a few exceptions, owing to previous differences in ontological philosophy and architecture: names of GEOPOLITICAL-ENTITYs (e.g., PAKISTAN), NATURAL-LANGUAGES (e.g., MANDARIN), and MONETARY-UNITS (e.g., DINAR). Current ontological methodology, however, makes a clear distinction between concepts and instances and assigns the latter to two separate databases, the onomasticon and the fact database.

The onomasticon is a store of people names (e.g., Charlie Rose), places/objects (e.g., Paris, France, Amazon River), organizations (e.g., IBM), and events (e.g., World War II, Super Bowl XXXII). The *fact database* lists such information as current monetary exchange rates for all international currencies (all based on the U.S. dollar) and leaders of countries.

The current treatment of complex events, however, requires further elaboration of this otherwise clear distinction between concept-fillers and instance-fillers. Within the frame of a complex event, it is useful to represent co-reference relationships, or *variable bindings*, between fillers of two or more events, in order to simplify semantic case-role assignment. For example, in an event of buying, the one who receives possession of goods is the same person who pays. Thus, within the complex event BUY-OBJECT (where the AGENT slot-filler is the buyer), the BENEFICIARY slot of the PAY event is filled is co-referent with the AGENT of the BUY-OBJECT event. This has the effect of restricting the range of slot-fillers allowable in the BENEFICIARY slot of PAY. However, since the AGENT slot of BUY-OBJECT takes the concept HUMAN as its selection restriction, the distinction between concepts and instances is maintained for complex events: an entity co-referenced with a concept is still a concept, not an instance. A much more detailed discussion is offered in Chapters Three and Four.

1.5 Building text meaning representations

Ontological semantics, as practiced in Mikrokosmos, uses the ontology, as well as several other modules, to build a text meaning representation (TMR) for a text. A TMR provides a semantic representation for the input text. Based on work on multilingual machine translation, it has been argued (Nirenburg and Raskin, to appear; Mahesh 1996) that the required richness of this TMR depends on the particular NLP task. Machine translation (MT), for example, does not seem to require that knowledge of, for example, complex events and speakers' plans be encoded in the system, since such knowledge appears to be recoverable in the translated text. However, for text summarization, information retrieval, and question-answering tasks, a finer-grained TMR does seem to be required. The process of building TMRs for such situations is the subject of this section.

1.5.1 The role of the ontology in building text meaning representations

An NLP system that derives a (more or less) complete meaning representation for a given natural language text must be able to determine three types of information:

- lexical meaning, by disambiguation and determination of word senses;
- propositional meaning, by determining dependencies between words at the sentence level;
- background knowledge, by determining relevant meanings not explicitly stated in the text;

An ontology contributes to each of the three, although not necessarily linearly. First, word senses are mapped to ontological concepts, such that identifying the relevant word sense amounts to assigning that word sense to an ontological concept (or concepts, since there is no one-to-one correspondence in the ontology). The morphological and syntactic analyzers also may contribute to the determination of word sense. Second, propositional-type meaning, e.g., speech-acts (cf. Austin 1975 [1962], Searle 1972), is determined in part by referencing ontological EVENT concepts, which are specified for semantic case-role (i.e., predicate arguments) structure. Third, background knowledge, e.g., of complex events, is determined through instantiation of conceptual knowledge encoded within EVENT concepts.

1.5.2 An example text meaning representation

TMRs are built by instantiating concepts linked to lexical items and by instantiating modalities dealt with specifically in the TMR shell (e.g., modalities such as aspect, telicity, expectation.) Consider an example TMR—much simplified, but retaining the relevant features—of the following sentence (taken from a newspaper article):

(9) Service Merchandise said it expects to emerge from Chapter 11 bankruptcy in 2001.

First, the relevant word senses must be assigned to the lexemes *Service Merchandise*, *said*, *expects*, *emerge*, *Chapter 11 bankruptcy*, and *2001*. Service Merchandise and 2001 are resolved to instances: *Service Merchandise* is listed in the onomasticon as an instance of a corporation; i.e., it is the name of a particular corporation; and, *2001* is resolved to an instance of a time, in this case a year.

The lexemes *said*, *expects*, and *Chapter 11 bankruptcy* are treated in ontological semantics as event-type concepts. The verb *said* is mapped to the ontological concept SPEECH-ACT. This is represented in TMR form as:

(10) speech-act-1

agent	Service-Merchandise
theme	hope-for-1

The numerical indexes following SPEECH-ACT and HOPE-FOR identify these terms as instantiations of concepts, as is the practice in NLP and artificial intelligence. The AGENT and THEME slots are those associated with the SPEECH-ACT concept frame. SERVICE-MERCHANDISE is a filler for the AGENT slot. (The fact that the agent is not actually Service Merchandise, but some human representative of the corporation, is ignored here.) Note that no index follows SERVICE-MERCHANDISE, as it is itself an instance of the concept corporation.) The filler of the theme slot, HOPE-FOR-1 (explained below), identifies an instantiation of a HOPE-FOR event as the THEME of the speech-act.

The verb *expects* is assigned the word sense expects-v6, based on both its "hope-for" meaning and its syntactic position as "V *to* V," e.g.., "expect to emerge." Therefore, *expects* is mapped to the ontological EVENT concept HOPE-FOR. (An alternative would be to assign an "expectative" meaning of the emerging from bankruptcy, which would be instantiated in the TMR as a modality. However, this would not retain the crucial distinction between "Service Merchandise expects" and "It is expected that.") *Chapter 11 bankruptcy* maps to the ontological concept BANKRUPTCY, an event that takes a corporation, here Service Merchandise, as an agent. Based only on lexical mappings to the ontology, the TMR thus far is:

(11) speech-act-1

	agent	Service-Merchandise
	theme	hope-for-1
	time	time-1 [=Feb. 16, 2000]

hope-for-1

	agent	Service-Merchandise
	theme	bankruptcy-1[aspect-1]

bankruptcy-1

	agent	Service-Merchandise
	theme	bankruptcy-1
	effect	reorganization-1

Finally, the verb *emerge* takes on a figurative meaning: a corporation may emerge from a bankruptcy, as a ship may emerge from a fog. Although Mikrokosmos has inference mechanisms with which to treat such metaphorical language, *emerge* is treated here, literally, as the ending of the bankruptcy (with certain specifiable results). The ending of events is treated as modality specified in the TMR shell. Thus, associated with the BANKRUPTCY-1 event will be:

(12) aspect-1

	phase	end
	telicity	yes
	time	time-2
	scope	bankruptcy-1

Finally, it remains to be decided what exactly will happen in the year 2001, according to the expectations of Service Merchandise. True, the expectations are for the bankruptcy to be completed. However, there is a further expectation that a reorganization—acceptable by the corporation's creditors—will take place by 2001. This information is not explicitly stated in the text, but is recoverable by a human understander with a general idea of how corporate bankruptcies work. In fact, without such knowledge, an understander (whether human or computer) may be said to have misunderstood the entire meaning of the sentence. The rest of this dissertation is concerned with representing such knowledge for an NLP system.

Therefore, the full TMR is as follows:

(13) speech-act-1

agent	Service-Merchandise
theme	hope-for-1
time	time-1 [=Feb. 16, 2000]

hope-for-1

agent	Service-Merchandise
theme	bankruptcy-1[aspect-1]

bankruptcy-1

agent	Service-Merchandise
theme	bankruptcy-1
effect	reorganization-1

reorganization-1

agent	Service-Merchandise
theme	Service-Merchandise
time	time-2 [= the year 2001]

aspect-1

phase	end
telicity	yes
time	time-2
scope	bankruptcy-1

after-1

arg1	time-2 [= time of emerge-2, i.e., 2001 (year)]
arg2	time-1 [= time of statement, i.e., Feb. 16, 2000]

1.6 Conclusion

The Mikrokosmos ontology is a richly interconnected hierarchy of conceptual knowledge that provides a grounding for world knowledge and contributes, along with morphological and syntactic analyzers, to the building of a text meaning representation

(TMR). The TMR shell is the repository of information such as aspect, speaker attitude, and other discourse-situation knowledge. World knowledge, as defined here, includes lexical information that is mapped to the ontology and knowledge of complex events, which "hang" from ontological concepts.

The ontology helps ensure equal coverage of conceptual entities and is ideally suited for the representation of complex events, which the following chapters show to be consistent with, and even dependent on, hierarchical representation. Previous treatments of complex events suffered most notably from this lack of organization. Past research on programs implementing complex-event knowledge is the subject of the following chapter.

CHAPTER TWO:

COMPLEX EVENTS IN THE LITERATURE

2.1 Introduction

Chapter One discussed the frame-based representational formalism of the ontological-semantic NLP system employed in Mikrokosmos, which is the framework for the treatment of complex events described in this dissertation. It was demonstrated that the Mikrokosmos system is able to provide rich text meaning representations (TMRs) for natural language texts, with the use of an ontology that provides mappings for lexical items and in conjunction with morphological and syntactic analyzers. Furthermore, it was demonstrated that this system is designed to allow seamless integration of complex-event knowledge.

The primary goal of this chapter is to review past research into complex events (section 2.4), in order to provide background for the recommendations to be made in Chapter Three and the implementation described in Chapter Four. First, however, it is necessary to clarify some differences in terminology (section 2.2) and to provide an example demonstrating the usefulness of representing complex-event knowledge for NLP tasks (section 2.3).

2.2 What are complex events?

2.2.1 Terminology in the literature

The representation of complex events resembles, roughly, a merging of a knowledge frame (in the sense of Minsky 1975) and a script (e.g., Schank and Abelson 1977; Cullingford 1978). Scripts are similar to the scenarios in Lakoff (1987) (cf. also Raskin 1986). The notion of story schema(ta), in Rumelhart (1994), is also related to the notion of a script. Because there is often a conflation of these terms in the literature—academic terminology being no less subject to lexical drift—we specify here how the terminology is used in this dissertation. The notion of a frame, developed most notably in Minsky (1975), posits that knowledge can be "chunked." The example provided by Minsky is that of the contents of a room: a kitchen may contain such items as a refrigerator, table and chairs, oven, sink, cabinetry, etc. Thus, the usual organizational pattern for frames is conceptual in nature.

While both frames and scripts chunk information, scripts traditionally imply temporal ordering (e.g., Nagao 1990, Allen 1995). That is, scripts specify not only conceptual information such as the roles (actors, themes, etc.) associated with a situation, but also the component subevents, ordered according to the stereotypical order these subevents are likely to occur. The traditional example of the restaurant script, for example, may specify that "leaving a tip" occurs after "ordering a meal."

Of course, there are good reasons why the terms *frame* and *script* are often merged. Both imply grouping of knowledge under some data structure—the first primarily conceptual, the second primarily temporal. However, the architectures are not mutually exclusive, but rather may be merged: conceptual information about atoms (e.g., objects, concepts, properties) may be implemented in a system that infers a sequence of events; and, conversely, temporal sequencing information about events may be implemented in a system that provides conceptual information about particpants in an event.

The term *complex event* has been used in past research (e.g., Charniak 1977, Carlson and Nirenburg 1990, Mahesh 1996) to denote that an event may be composed of other events. In this way, *complex event* and *script* denote much the same thing. In fact, throughout this dissertation the terms are taken to be synonymous.

In brief, a complex event is represented, in an ontological-semantic system, as an EVENT frame listing both conceptual information and temporal sequencing of component subevents, as well as such information as whether subevents are optional/criterial, branching structures (i.e., “either…or”), conditionals (i.e., “if…then”), or loops (i.e., “do X until Y”). The Mikrokosmos system provides such an architecture for this representation.

2.2.2 Complex events vs. “simple” events

Because this is the first attempt at building a system that incorporates both simple and complex events, it is necessary to provide a motivated distinction between the two. First, a distinction that does not work is explained. It is tempting to argue that an event like “going to a restaurant” is complex and “throwing a ball” is simple because the former has more easily discernable parts to it. However, two problems immediately arise. The first of these is that “grain-size” matters: if a text explicates all the necessary movements in the act of throwing, then “The woman threw the ball” itself suddenly becomes a complex event made up of, at the very least, picking up the ball, cocking the arm back, bringing it forward, and releasing the ball. Similarly, it is quite possible to construe “Susan ate at the restaurant” as a simple event, if we are not concerned with whether she seated herself, was given a menu, or left a tip, but merely that she ate a meal in a particular place. The second problem with this initial definition is that although there may be some more or less clear-cut preferences for classification at a certain grain-size, there will be plenty of events in the gray area in the middle. For example, should “John bought a candy bar” be classified as a simple or a complex event? How about “Mary unlocked the door”? It is therefore argued that the presence of discernable of subevents is not a good criterion by which to distinguish simple and complex events.

The distinction settled on is a rather pragmatic one, dictated by constraints on representation in a computer system: an event is considered complex if its frame includes subevents, preconditions on the event’s occurrence, and postconditions on its completion, where subevents of a given event are included if and only if the items in the former are exploited by the text. That is to say, all events are theoretically complex. However, complex-event knowledge is only represented when the tasks of the system warrant such information.

2.3 The usefulness of representing complex-event knowledge

It was remarked in section 1.3.2.2 that some tasks may not require representation of complex-event knowledge (e.g., machine translation). However, there clearly are numerous cases in which such knowledge is necessary. This section addresses the motivation for the usefulness of representing complex-event knowledge by way of a sample task for an NLP system. Of course, even the trotting out of hundreds of such example sentences will not allay potential criticisms that the data may be biased; however, the example presented here should be intuitive enough.

Suppose the task for an NLP system is to determine, on the basis of inputted news article texts, which corporations have declared bankruptcy. Suppose further that the system encounters such fairly normal sentences as those in (1-11) (for clarity, corporations that are not in bankruptcy are italicized; and, anaphoric referents not recoverable given this limited context are provided in brackets):

(1) The owners of bankrupt Mount Airy Lodge and several affiliated resorts have won more time to seek refinancing or find a buyer for the properties before they come up for sheriff's sale.
(2) Iridium LLC, the operator of the first satellite-telephone service, said it [= Iridium] will shut down after failing to attract a bid for the bankrupt company's [= Iridium] assets.
(3) Faced with mounting debts, Humpty Dumpty filed for bankruptcy protection on Dec. 29.
(4) A source close to Loehmann's said the Bronx-based retailer [= Loehmann’s] was proceeding with plans to emerge from Chapter 11 bankruptcy in mid-March as an independent entity.
(5) It [*Nabisco Group Holdings*] may have some residual liability if *R.J. Reynolds* is driven into bankruptcy, because it is a former parent of *Reynolds*.
(6) *Dakota Bank*, which made a $5 million loan to Excelsior-Henderson, had taken exception to the manufacturer's [= Excelsior-Henderson] use of cash during the bankruptcy period.
(7) After the bankruptcy made clear that SubMicron shareholders would get nothing from their investments, share prices dropped to less than a penny.
(8) *Nextel* said it reconsidered its hostile bid Wednesday after the U.S. Court of Appeals for the 2nd Circuit issued an explanation for its decision in November to reverse the rulings of two other courts in NextWave's bankruptcy case.
(9) Cellular telephone pioneer Craig McCaw held talks with Iridium's creditors and *Motorola* about purchasing the company [=Iridium] for about $600 million, before dropping the proposal earlier this month.
(10) The Dayton, Ohio-based chain [= Roberds, Inc.] said it obtained $25 million of debtor-in-possession financing from *PPM Finance Inc*. in Chicago. The line has a

27-month term, which is expected to be long enough to carry the retailer [= Roberds, Inc.] through reorganization.

(11) At a bankruptcy hearing Friday, lawyers for Guy's and *General Products* said the company [=Guy's] had resurfaced with another, unspecified proposal.

The bankrupt/non-bankrupt status of the corporations listed in (11) is indeterminate in this context, and so it is ignored for the rest of the analysis; its presence, however, should remind us that we cannot reasonably expect more from an NLP system than a human understander is capable of. Concerning the remaining sentences, note that two simplistic attempts fail at determining whether a mentioned company is/is not in bankruptcy: (i) those corporations named in the same sentence as the words "bankrupt" or "bankruptcy" identifies only 10 out of 15 correctly; and, (ii) those corporations named within plus/minus 3 lexemes of the words "bankrupt" or "bankruptcy" identifies only 11 out of 15 correctly. Furthermore, although using the word string "bankrupt [Corporation-1]" for searching will likely achieve near 100% accuracy in picking out Corporation-1 as bankrupt, as in (1) and (2), the construction is far too infrequent to be of much use.

Sentences (3-5) are more difficult to be handled by searching for word strings and require a text meaning representation that:

(i) *represents effects/consequences of an event*, e.g., that a corporation that files for bankruptcy is bankrupt, as in (3); and, that a corporation that will emerge from bankruptcy is bankrupt, as in (4);

(ii) *represents a conditional "if" statement as counterfactual*, as in (5).

The Mikrokosmos system has both of these capabilities: an EFFECTS slot, taking as a filler another EVENT, is specified in an EVENT frame; and, counterfactualness is represented as a modality in the TMR shell.

However, (6)-(10) require a text meaning representation much more sensitive and fine-grained than (3)-(5). That is, determining the bankrupt/non-bankrupt status of each of these corporations mentioned in (6)-(10) requires that an understander make some inferences not explicitly represented in the texts; instead, an understander must draw on knowledge of the way events in the world proceed. The inferences that seem to be required are:

(6'a) a bank that has lent a corporation funds is a creditor of that company;
(6'b) a creditor of a bankrupt corporation has a legal right to object to that corporation's use of funds;
(6'c) a corporation may not make a loan while bankrupt;
(7') share prices of a bankrupt corporation usually fall drastically;
(8') a corporation making a "hostile bid" is not bankrupt;
(9') a bankrupt corporation's creditors must be consulted about the sale of the bankrupt corporation;
(10'a) a bankrupt corporation is known as a "debtor-in-possession";
(10'b) a corporation undergoing "reorganization" is in bankruptcy.

Each of these necessary inferences can be represented as complex-event knowledge, as will be demonstrated in Chapter Four. For now, however, it is simply noted that information-extraction-type NLP tasks may require knowledge of complex events.

The next section discusses previous systems designed to use such complex event knowledge in order to generate required inferences.

2.4 Previous research into representing complex-event knowledge

The relative dearth of research into, and NLP systems implementing, the representation of complex events in knowledge bases in the past 15 or so years belies the groundbreaking work Roger Schank and his colleagues and students did in the 1970s and 1980s at Yale University's Artificial Intelligence laboratory. These two periods are, of course, related: after early theoretical successes concerning representing the structure of event memory and recall, the sobering realization that a tremendous amount of knowledge needed codified stymied further research. Indeed, Schank himself all but abandoned the idea of building intelligent computer systems over 10 years ago (Schank 1999).

In this section we review the initial contributions of Schank and Abelson's (1977) seminal work, including the conceptual dependency (CD) framework (section 2.4.1). Then, SAM is reviewed in section 2.4.2, followed by a similar program, Charniak's (1977, 1979) Ms. Malaprop, in section 2.4.3. Three other systems developed at Yale under Schank are then reviewed: FRUMP (section 2.4.4), PAM (section 2.4.5), and POLITICS (section 2.4.6). Finally, Miikkulainen's (1993, 1995) connectionist approach to the representation of complex events is reviewed in section 2.4.7.

2.4.1 Schank and Abelson (1977)

The publication of Schank and Abelson's influential *Scripts, plans, goals and understanding*, along with Minsky's (1975) paper, signalled a shift in AI from simple and fragmented knowledge structures to ones that are complex and "chunked." Simultaneously, there was also a shift in focus from isolated sentenes to paragraph- or story-level discourse. Schank and Abelson's (1977) main point is that the understanding of stories (and, indeed, any natural language text) amounts to constructing and connecting causal chains that explain what events happened and why, as in the following example:

(12) John ordered a new suit. He paid the bill with his credit card. (*Ibid*.: 32)

Understanding (12), according to Schank and Abelson, amounts to reconstructing the causal chain that connects the first sentence in (12) with the second; i.e., (i) one orders a new suit, which (ii) causes a need to pay for the suit, and (iii) a credit card is one way of paying. As Schank and Abelson note, the knowledge represented in (ii) is not recoverable from (12). Since natural language texts regularly underspecify links in the relevant causal chains, other information about the world is needed to enable understanding of texts.

For Schank and Abelson (1977), this information can take the form of a *script*, "specific detailed knowledge about a situation" (*Ibid*. 37); a *goal*, general information about what an actor might want to achieve; a *plan*, "general information about how actors achieve goals" (*Ibid*. 70); and, *themes*, knowledge about "the background information upon which we base our predictions that an individual will have a certain goal" (*Ibid*. 132). The following subsections review detailed programs developed at Yale to address the role of these knowledge types in story understanding.

Before doing so, however, two notes relevant to both the Schank and Abelson (1977) and the dissertations based upon it deserve mention. First, a philosophical point. An overt goal of all the work done by Schank and his followers at Yale was to simulate the actual structure of human memory and how it was brought to bear on the understanding process; that is, these AI projects strived for psychological reality: the subtitle of Schank and Abelson (1977) is *An inquiry into human knowledge structures*. Moreoever, Wilensky (1980: 35) intends that his program, PAM, "be judged by its psychological appeal." This dissertation, however, makes no claims about how the human understanding process works. While comparing a NLP system to the process of human understanding may sometimes serve as a useful heuristic for discovering and defining necessary components of a (computational) understanding process, the end goal is virtually never to equate the two. For example, NLP researchers are not concerned with building systems that will exhibit Freudian slips, error in recall, and other errors in performance. Moreover, computer systems display a level of consistency and accuracy that cannot reasonably be expected of a human. Thus, while the goals in representation for a NLP system may overlap significantly with a human's cognitive abilities, this dissertation is concerned much less with psychological reality than it is with inferencing capability. Second, and more importantly, all systems built under Schank at Yale analyze input texts in terms of Schank's conceptual dependency (CD) framework (Schank 1972, 1975). Here, the relevant features of CD are described, along with its limitations. CD is a conceptual representation formalism based on the following premise:

(13) "For any two sentences that are identical in meaning, regardless of language, there should be only one representation" (Schank and Abelson 1977: 11).

While this is pleasant enough, the corresponding corollary, (14), which is not addressed in Schank and Abelson (1977), is, of course, debatable.

(14) Any two sentences that are *not* identical in meaning, regardless of language, should *not* share the same representation.

Though two non-synonymous sentences might be reasonably expected not to share a representation, if differences between them are, at some grain-size, not salient, representing these two sentences as having the same representation may be motivated. Thus, in most contexts, the differences between (15a) and (15b) are minimal enough to motivate the same representation.

(15a) Mary chose John.
(15b) John was chosen by Mary.

However, the CD formalism provides no mechanism to distinguish (15a) and (15b) with different conceptual representation, should the difference between them become salient.

The primary weakness of CD—as well as any other conceptual representation formalism that attempts to employ just a handful of primitives—is this inability to generate fine-grained representations. CD posits 11 "primitive acts": ATRANS (= abstract transfer), PTRANS (= physical transfer), PROPEL, MOVE, GRASP, INGEST, EXPEL, MTRANS (= mental transfer), MBUILD (= building new information from old), SPEAK, and ATTEND (i.e., focus on) (Schank and Abelson 1977: 12–14). There would be no problem, of course, if inference mechanisms were sensitive only to these broad classifications. Yet, it should be quite obvious that (16a) and (16b) are different, though the verb in each sentence is represented as ATRANS in CD notation:

(16a) John bought a new car.
(16b) John leased a new car.

Because the parser for SAM, PAM, POLITICS, and FRUMP is a version of ELI (English Language Interpreter; Riesbeck 1975) that generates CD meaning structures, this is a weakness for all the systems. As will be discussed, the results of "bottom-up" processing (parsing a text and assigning a (partial) text-meaning representation) may facilitate "top-down" processing (activating knowledge structures). Thus, coarseness in bottom-up processing leads to inappropriately applied knowledge structures.

2.4.2 SAM

The Script Applier Mechanism (SAM)—the subject of Cullingford's (1978) dissertation at Yale—reads simple stories and applies knowledge about stereotypical events in order to understand these stories. In SAM, scripts are organized as "collections of events linked into causal chains" (Cullingford 1981: 76); each event in the script enables the next event specified in the script. Script events are coded as generalized CD representations, or "patterns," that are matched to input events. To simplify the process of pattern recognition, script sequences are ordered as "situations." For example, the events comprising a stereotypical motor-vehicle accident are organized into the VEHACCIDENT situation.

Input natural language is first converted to a CD representation using the English Language Interpreter (ELI). At this stage, reference and coreference relations are established, for use by the Script Applier module. Thus, in the following example (minimally changed from Cullingford 1978: 165):

(17) Mary Jones died Tuesday of head injuries she received in a car accident on Sunday,

"Mary Jones" will be identified as the referent of the theme of the event, and "she" will be resolved as coreferent with "Mary Jones." Then, the Script Applier attempts to instantiate a script that will account for the story, using a "backbone matching" process. The first problem encountered arises because of the lack of a parser robust enough: presuming the need for a VEHACCIDENT script, such a script will need to be activated even if "in a car accident" is replaced with near-equivalents such as:

(18)
(a) in an auto accident
(b) in a car crash
(c) in a three-car pile-up
(d) when her Accord skidded off the road and struck a telephone pole
(e) when she was broadsided by a semi at the Main and Elm intersection.

Ignoring this problem, "backbone matching" proceeds by finding input CD patterns that match script patterns. Spurious inferences are prevented by requiring further pattern-matching at the role-filler level. For example, in the sentence "John decided to throw a ball at the museum," the MUSEUM script is blocked from activation based on a specification in the script that in sentences of the form "Actor ACTION Object," Object is coreferent with Actor. Furthermore, some script roles may be marked as requiring explicit mention in order for the script to be activated. However, such a specification presumably still will not block sentences like "John thought about going to the museum" from activating the script. After a particular script has been chosen as being active, the Script Applier begins making predictive assumptions about what will follow.

During the processes of "Rolefit" and "Rolemerge," the Script Applier attempts to assign text instantiations of role values (i.e., semantic case-roles) to CD pattern variables. Application of a sequence of script patterns proceeds sequentially: beginning with the first pattern in the script, the Script Applier attempts to bind pattern variables. If successful, the next pattern in the sequence is checked and candidate patterns are loaded into active memory for pattern-checking. The script-applying process can be seen as a moving window: as patterns are matched, they are dumped from active memory and new candidate patterns are loaded. Because SAM assumes that the input text order of events will conform to the sequence of patterns in the script, the system will have trouble with a text such as the following:

(19) After eating a burnt hamburger, John left the restaurant. He wished he had left a smaller tip.

After processing the first sentence, SAM will presumably have dumped the RESTAURANT script altogether, meaning that the second sentence may not be recognized as being part of the script. Moreover, because SAM relies on "top-down," predictive processing, strategies for aborting a script, recognizing deviations, and reordering events must be explicitly coded in the script itself, with little or no help from the parser. Placing such a heavy burden on the Script Applier module leads to *brittleness* of the system, i.e., the inability to recover from deviations from expecation, since it requires that, in order for script application to be accurate, all possible deviations and reorderings must be encoded—probably an impossible task.

SAM was intended as a limited-domain experiment to test the viability of script-applying mechanisms. As such, it succeeds. However, given the more ambitious task of developing complex-event knowledge for use in domain-unrestricted NLP systems, it is wise to ask questions concerning the extendability of a SAM-type system. First, script recognition will become more of problem as scripts less disparate than VEHACCIDENT, VIPVISIT, and OILSPILL are added to the system. For instance, the fine-grained, but important, distinctions between “buying a car” and “leasing a car” require a sensitivity to context much greater than simply the status of theme case-roles. Second, the construction of a large knowledge base of script knowledge will undoubtedly call for the flexibility to make certain subevents available to more than one script. SAM applies script knowledge in a modular fashion, meaning that the subevent “buy a ticket” in SUBWAY will need to be explicitly represented in MOVIE, FOOTBALLGAME, AIRPLANE, etc. Such duplication of efforts is not only unwieldy in terms of system memory considerations, but also may lead to missed opportunities in inferencing.

Schank (1983) later incorporated this idea into the theory as *memory organization packets* (MOPs), which are, basically, groups of stereotypical sequences available to any number of different scripts. None of the programs developed at Yale—or anywhere else for that matter—implemented this idea. The Mikrokosmos system, however, requires that a given event that is contained within a complex event also be represented in the ontological hierarchy, making this event available to other complex events as well.

2.4.3 Ms. Malaprop

Based on much the same principles as Cullingford's (1978) SAM, Ms. Malaprop (Charniak 1977, 1979) was developed to handle simple stories on the topic of painting. Like SAM, Ms. Malaprop relies on "top-down" processes; i.e., it reads a story with the explicit intention of filling data structures it has available. As such, difficulties arise when it encounters input that is not represented in the system.

Unlike SAM, which represents script information in LISP-like Conceptual Dependency templates, knowledge in Ms. Malaprop is represented in frame structures arranged hierarchically. The painting complex event is comprised of several frame statements that provide information about the process of painting. Although Charniak may have in fact coined the term complex event, these *frame statements* do not all have the status of events. For example, PAINTING6 describes the state of a painting instrument being in contact with an object.

Frame statements specifically about painting are also connected to knowledge of the physical world. Thus Ms. Malaprop is equipped with the knowledge that: (i) a paintbrush should be washed after it is used; (ii) not doing so will cause the paint to dry on the brush, making it unusable; (iii) in turn, liquids dry when left in the open air; and, (iv) this is because of the process of evaporation.

Despite these insights into knowledge representation, Ms. Malaprop still suffers from the lack of a parser to handle knowledge not represented in its system.

Furthermore, the existence of only one complex event makes it impossible to evaluate whether the system could handle similar complex events, such as wallpapering and painting a portrait.

2.4.4 FRUMP

FRUMP (Fast Reading Understanding and Memory Program) (DeJong 1979, 1982), developed at Yale, skims and summarizes newspaper articles with the help of less-detailed scripts, or *sketchy scripts*, than are used in SAM. FRUMP was one of the first systems that attempted to use real-world natural language inputs, although the domain was limited to those for which a sketchy script was available. Included in the system are about 60 such sketchy scripts. The goal of the system is to demonstrate that rapidly deployable sketchy scripts may be able to generate usable one-sentence summaries of documents.

FRUMP differs from programs like Ms. Malaprop and SAM in that the inference mechanism (i.e., the script applier; or, the "PREDICTOR," in DeJong's (1979, 1982) terms) may provide feedback to the parser, which can then substantiate these predictions, formulated as (truncated) CD structures. These other programs had separate parsing and inferencing modules with only one-way communication: the results of parsing were sent to the inferencing mechanism, which then applied the script knowledge. The benefit of the FRUMP architecture should be obvious: it is many times the case that anaphoric reference cannot be resolved by parsing alone. The oft-cited example (20) demonstrates this clearly.

(20) The police arrested the demonstrators, because
(a) *they* feared violence.
(b) *they* advocated violence.

The point is that world knowledge must be activated in order to assign the correct anaphor for *they* in (20a) and (20b). Section 1.4.1 of this dissertation describes the workings of this process for the Mikrokosmos system.

DeJong (1982) describes three mechanisms for script recognition: explicit-reference activation, implicit-reference activation, and event-induced activation. Because script recognition is a recognized problem in the implementation of this kind of knowledge, DeJong's claims are briefly reviewed here. *Explicit-reference activation* involves detecting a word-sense (not simply a word form) that "identifies the script that is to be activated" (*Ibid*. 161). Thus, the presence of the police-apprehension word-sense of "arrest" activates the ARREST script in:

(21) John Doe was arrested last Saturday morning after holding up the New Haven Savings Bank. (*Ibid*. 161)

While this may be true of a limited number of verbs, at least 20 – 25 of the 60 scripts included in FRUMP seem not identifiable based on a "matrix" verb. Furthermore, the explicit-reference mechanism seems less useful the more complex an event is: thus, WAR, ELECTION, and SPORT-GAME would seem to elude recognitions based on this mechanism. Finally, activating complex-event knowledge based on one mention can become a problem unless the system has a way of coping with simultaneously active scripts.

Sketchy scripts in FRUMP may also be activated based on *implicit-reference activation:* "a sketchy script is activated by implicit reference when a sketchy script that is known to often precede it is activated" (DeJong 1982: 162). The example given is that if CRIME is activated by explicit reference, then ARREST may be activated as well, since a crime often results in an arrest. The problem with this approach appears when, for example, we try to implement a script for GET-AWAY (as an possible alternative to ARREST event) in the system. The only way this can be done is by introducing a branching structure in the script. But, of course, this nullifies the original generalization that allowed ARREST to be activated based on CRIME. So we are left with a knowledge base that activates an incorrect script a significant percentage of the time. Clearly, this is unappealing. Furthermore, allowing a script to be activated when a temporally or causally related script is activated will simply not do in a rich system of interconnected scripts: we may well find an infinite chain of script activation: e.g., CRIME activates ARREST, which activates JAIL, which activates PAROLE, etc. What is needed is for the knowledge base to recognize when complex-event-1 is part of complex-event-2, and then search for further confirming evidence that complex-event-2 should be activated.

Finally, FRUMP recognizes scripts by *event-induced activation*, which amounts to an explicit reference of a near-synonym of a script-activating concept. DeJong (1982: 163) states that "the event of the police's apprehending a suspected criminal is sufficient for people to realize that knowledge about arrests (that is, ARREST) will be relevant." This may be true, but such an algorithm will not be a very important part of script recognition, given the variety in natural language input.

In conclusion, the contributions of the FRUMP system are incorporating a larger number of scripts, attempting to find a niche by generating quick accurate summaries, and trying to solve the script recognition problem. Solutions to the latter seem to be insufficient, especially as the knowledge base grows.

2.4.5 PAM

The Plan Applier Mechanism (PAM) (Schank and Abelson 1977; Wilensky 1978, 1980, 1981) is the subject of Wilenksy's (1978; reprinted as Wilensky 1980) dissertation at Yale. An elaboration of Schank and Abelson's (1977) initial treatment of goals and plans as generalized knowledge structures needed to process texts, PAM takes as input stories in English 2–9 sentences long, produces CD representations of the stories, using the English Language Interpreter (ELI), and draws inferences about plans and goals that are needed to explain the conceptual representation. PAM encodes knowledge about, and attempts to draw inferences about, plans and goals that people may have, in order to understand simple stories. To see how knowledge of plans and goals may be instrumental to text understanding, consider the following, from Wilensky (1980: 12):

(22a) John loved Mary. One day, John saw a truck coming down the street toward Mary. John ran up behind Mary and gave her a shove.
(22b) John hated Mary. One day, John saw a truck coming down the street toward Mary. John ran up behind Mary and gave her a shove.

Here, determining the result of John shoving Mary (Mary is saved in (22a), but killed in (22b)) seems to depend entirely on knowledge people have about how someone who loves/hates another person is likely to act.

PAM encodes 180 rules concerning goal behaviors, expressed as "situation-action" pairs, i.e., "If…Then" conditionals. A taxonomy is given, including:

(23)
(a) *goal subsumption*, i.e., when a goal enables another goal;

(b) *goal conflict*, i.e., when two goals (of one person) conflict for precedence;
(c) *goal competition*, i.e., when a person's goal competes with another person's goal;
(d) *goal concord*, i.e., when a person shares a goal with another person.

Examples of rules used by PAM are the following:

(24) IF: a character has a positive belief about an action that may be dominated by a recurring goal, THEN: that character may want to subsume that goal;
(25) IF: a goal conflict is due to a shortage of a consumable object, THEN: the plannner can resolve the conflict by acquiring more of that consumable.
(26) IF: (1) two planners have goals that are in concord and (2) one of the goals is in competition with the goal of another character, THEN: one planner can anti-plan [i.e., plan to thwart another's plan—CM] against his competitor while the first planner pursues the original goal.

Though, to be fair, the above rules are a convenient shorthand for the CD representations in the program, we must still ask whether rules like (24)-(26) are in principle tractable for unrestricted domains, even assuming the parser can identify when, for instance, a character has a "positive belief."

It should be noted that PAM addresses a higher-order concern than does this dissertation. Implementing goal-type knowledge will follow, in the Mikrokosmos project, implementation of complex-event knowledge, much like SAM preceded PAM at Yale. In passing, it should be noted that knowledge about goals and plans will only be of use to an NLP system that deals with texts addressing human actions. Therefore, a large body of texts—e.g., academic journal articles, news stories on natural disasters, texts on government legislation—will probably not require this knowledge module. Some requirements for representing such knowledge in an ontological-semantic system are addressed in Chapter Five.

2.4.6 POLITICS

Carbonell's (1981) POLITICS models understanding of themes, one level more abstract than plans/goals in Schank and Abelson's (1977) original hierarchy of generalized knowledge. POLITICS encodes information about liberal and conservative political ideologies and then makes predictions about what courses of action each would follow in given situations. While the primary goal of the POLITICS system is on a higher order than that of this dissertation, it should be pointed out that the system does incorporate 13 scripts specifically about political events. As such, the system represents the first attempt to build knowledge for closely related complex events; it is on this module that the review here concentrates.

Of the 13 scripts used by POLITICS, some are similar, such as ARMSALE and MILITARY-AID or INVADE and CONFRONTATION. Inference rules may be *context-dependent*, meaning that they only fire when a certain script is activated. *Context-switching* rules handle the difficult problem of script recognition and switching. Again, the heuristic used is that if a conceptual representation matches either an initial event in a script sequence or the "main concept" of the scene (cf. Schank and Abelson 1977), then the script is activated. As was mentioned in section 2.4.4 on FRUMP, assigning a main concept to a script may be problematic if used alone, as appears to be the case for POLITICS.

2.4.7 DISCERN

Miikkulainen's (1993, 1995) DISCERN system is a subsymbolic artificial neural network system that processes simple, stereotypical texts such as the following (Miikkulainen 1995: 137):

(27) John went to MaMaison. John asked the waiter for a lobster. John left a big tip.

The program was designed as an initial demonstration that a distributed parallel processing model of script processing is theoretically feasible. DISCERN does not handle script branching or explain deviations from the script.

It is not wished to enter into a debate here over the relative advantages and disadvantages of symbolic vs. subsymbolic systems in this dissertation; such a debate may be postponed until either subsymbolic systems begin to approach the results obtained by symbolic systems or when symbolic systems encounter a ceiling beyond which they cannot pass. Rather, discussed here are two proposed advantages of subsymbolic processing mentioned in Miikkulainen (1993, 1995) that may indeed be handled by the (symbolic) system proposed in this dissertation. First, as Miikkulainen rightly notes, previous symbolic systems attempting script processing are rather brittle. However, this dissertation argues that this arises not because of the representational paradigm but rather from the limited expressiveness of the conceptual analyzer. The second related criticism of symbolic systems levied by Miikkulainen is that they do not allow for the emergence of statistical regularities in the data. Two facts counter this argument. First, statistical regularities are already built in to a symbolic system via knowledge engineering; that the system cannot do so on its own belies the fact that a great deal of statistically oriented knowledge is already available. Second, to make reliable use of “statistical regularities in the data” (Miikkulainen 1995: 157), a system—any system—must have an incredible sensitivity to context. Good and bad cues must still, even in a subsymbolic system, be discerned by a human user, in order for the subsymbolic system to adjust its weights. This amounts to nothing more than indirect symbolic processing.

2.5 Conclusion

The necessity of representing complex event knowledge is demonstrated not only through the very informal example NLP task given in section 2.3, but also by virtue of the kinds of inferences made possible by the actual programs reviewed in section 2.4. Schank and his followers and Charniak deserve credit for laying the groundwork for the ideas presented in this dissertation and in the Mikrokosmos project.

Though each of these programs helped demonstrate the feasibility of this approach, there are still many problems that prevent these programs from successfully generating inferences for unrestricted input, which should be the long-term goal. The important shorcomings are summarized here. First, no amount of knowledge engineering will substitute for a module that can generate a fairly fine-grained semantic representation. Ms. Malaprop and SAM suffer the most from this lack. As Dreyfus (1985: 85-86) has pointedly remarked, “no matter how stereotyped, going to the restaurant is not a self-contained game but a highly variable set of behaviors which open out into the rest of human activity.” Second, given the variable nature of natural language texts, it is wise to first produce a representation of the propositional meaning (to the extent that it can be disambiguated prior to the application of complex event knowledge) before activating complex-event knowledge. Again, Ms. Malaprop and SAM suffer from this inability. Third, it is clear that the complexity of representing complex-event knowledge increases significantly as more and more complex events are added to the system. While complex events for going to a restaurant, painting, and a car accident may work well enough in isolation, the addition of similar complex events to the knowledge base may prove problematic—unless each complex event is built with the totality of world knowledge in mind. Fourth, and finally, given the eventual enormous size of a relatively complete knowledge base of complex events, it makes sense for complex-event knowledge to be non-modular, to be able to "share" events. That is, whenever possible, components of a particular complex event should be designed for use in other complex events, to reduce redundancy. For example, "buy a ticket" should not be represented in "attend football game," "ride subway," and "go to the movies"; rather, a pointer, or placeholder, should reference "buy a ticket" in each of these complex events. Relatedly, similar complex events should be represented in a conceptual hierarchy so that knowledge common to both can be "factored out."

In Chapter Three, a formalism that solves these problems for an ontological-semantic system is discussed.

CHAPTER THREE:

REPRESENTING COMPLEX-EVENT KNOWLEDGE

3.1 Introduction

Now that the ontological-semantic NLP framework for the development of complex events has been described (in Chapter One) and the literature on complex events has been reviewed, with special attention to the limitations of these previous approaches (in Chapter Two), this chapter presents specific recommendations for the implementation of complex-event knowledge, within an ontological-semantic NLP system. Verification of these recommendations is presented in Chapter Four.

This chapter seeks to establish principles and heuristics for the representation of complex-event knowledge. Importantly, the scope is taken to be the entirety of world knowledge of complex events. That is, the recommendations put forth in this chapter are meant to be extendable for an NLP system. Therefore, avoided here is Hobbs' (1985: xi) criticism of systems encoding commonsense knowledge "too specific to be of much use to the field [i.e., AI—CJM] in general" (cf. also Dreyfus' (1986) fallacy of the "micro-world"). This goal is consistent with Hayes' (1985: 2) recommendations that the formalization of knowledge about the *physical* world should have: (i) *breadth*, i.e., knowledge that covers the "whole range" of phenomena; (ii) *density*, i.e., "the ratio of facts to concepts needs to be fairly high"; and, (iii) *uniformity*, i.e., there should be a "common formal framework." Although there are important differences between the physical world and the complex-event world—chief among these being the presence of *choice* (in events describing human behavior) in the latter, the similarities should be apparent enough. In fact, many physical-world phenomena will be represented as complex events in the ontology: for example, a tornado-event, evaporation of liquids, and the growth of a seedling.

The number of complex events is, of course, very high and may easily number in the tens of thousands. This dissertation develops only a small subset. However, this chapter does endeavor to develop a framework capable, in principle, of representing all complex events—therefore fulfilling Hayes' requirement of *breadth* of knowledge. Furthermore, the recommendations made in this chapter are sensitive enough to handle subtle differences in complex events—therefore fulfilling Hayes' requirement of *density* of knowledge—within the framework of an ontological-semantic system.

3.2 EVENT slots for complex events

Some modifications to the available slots in EVENT frames in the Mikrokosmos system are necessary to represent complex events. These changes are explained in section 3.2.2, following a brief synopsis of the current treatment of EVENT frames given in section 3.2.1.

3.2.1 The current treatment of EVENT frames

In (1), a typical frame for an EVENT concept, that for BUY, is shown.

(1)

Frame	Slot	Facet	Filler
BUY	DEFINITION	Value	"A transfer of possession event (buying or selling) that involves the exchange of money for merchandise"
	IS-A	Value	EVERYDAY-FINANCIAL-EVENT, TRANSFER-OF-POSSESSION
	SUBCLASSES	Value	AUCTION, BUYOUT
	ACCOMPANIER	Sem	HUMAN
	AGENT	Sem	HUMAN
	BENEFICIARY	Sem	HUMAN
	DESTINATION	Sem	HUMAN
	LOCATION	Sem	PLACE
	SOURCE	Sem	HUMAN
	THEME	Sem	OBJECT, (Not HUMAN)
	COST	Sem	(>0)
		Meas-Unit	MONETARY-UNIT
	HAS-PARTS	Default	TRANSFER-OBJECT
		Sem	EVENT

BUY is provided with 12 slots that specify:

- a definition (not used, however, for computation);
- its placement in the ontological hierarchy, i.e., IS-A, SUBCLASSES;
- semantic case roles associated with the event: ACCOMPANIER, AGENT, BENEFICIARY, DESTINATION, LOCATION, SOURCE, and THEME;
- other information associated with the event, e.g., here, that buying entails a cost and a transferring of an object.

Note that although the buyer, beneficiary, and destination of a piece of merchandise are often co-referent, the frame representation is sufficiently general to allow for sentences such as in (2):

(2) Mary (=Agent) bought her son (=Beneficiary) a yacht at the harbor (=Destination).

Many EVENT concepts in the ontology are provided with a HAS-PARTS slot, in which to specify (at most) one other associated component event. Furthermore, an EFFECTS slot, in which to specify (at most) one associated consequent event, is also defined for some EVENT concepts. However, this frame structure is unable to represent complex-event knowledge such as: that a buyer and a seller may first have to agree on the cost, that a buyer must tender money, that a buyer may receive money if the amount tendered is greater than the cost, etc.

To represent such complex-event knowledge, the ontology requires a richer frame structure, with more complex HAS-PARTS and EFFECTS slots. The next section details the necessary elaborations.

3.2.2 Modifications to frame structure

We would like, in organizing knowledge about complex events—in fact, about *any* Event-X—to represent such information as:

- the necessary events that must obtain for Event-X to occur;
- the component events of Event-X, if any; and,
- the consequences (i.e., postconditions or effects) of Event-X occuring.

Providing this information in the knowledge base will allow an NLP system to make a wide range of inferences, as will be shown in Chapter Four. The Mikrokosmos system will represent these three types of information in three slots within an EVENT concept frame: PRECONDITIONS, HAS-PARTS, and EFFECTS.[5] It is further stipulated that each of these slots may take an unlimited number of fillers.

Because the fillers for these three slots may be events contained within complex events, a brief explanation of terminology used in this section is offered here. First, a *subevent* is any event included in a complex event; a *candidate event* refers to a subevent

---

[5] A fourth slot, BINDING-ROLE, is motivated in section 3.5.5.

under consideration for inclusion in the complex event. The term *event*, when used without a modifier, will be used only refer to the complex event. Finally, a *component event* refers to a subevent specified in the HAS-PARTS slot of a complex event. Concerns general to the PRECONDITIONS, HAS-PARTS, and EFFECTS slots are discussed first.

In assigning a candidate event to the PRECONDITIONS, HAS-PARTS, or EFFECTS slots, the typical criterion is whether the candidate should be included is strict logical entailment; that is, if event A happened, then necessarily event B happened, where B may occur either *prior to* (i.e., Precondition), *during* (i.e., Has-Parts), or *after* (i.e., Effects) the occurrence of event A. Note that the only concern here is with the *truth values* of A, B, and $A \Rightarrow B$ (i.e., A implies B) and not with their temporal ordering: for example, if B is a precondition of A, this does not mean that if B is true then A will be true also. In other words, although John having at least a quarter in his pocket is a necessary precondition of him buying a newspaper, this does not mean that every time John has at least a quarter in his pocket he will buy a newspaper. We will want to include candidate events that are logically entailed by an event in the PRECONDITIONS, HAS-PARTS, or EFFECTS slots. Such conceptual information will be marked with a Sem facet.

However, we will not want to abide so slavishly by the rules of *modus ponens* (i.e., if $A \Rightarrow B$ is true and A is true, then B is true) and *modus tollens* (i.e., if $A \Rightarrow B$ is true and B is false, then A is false), since the real world offers many exceptions. For example, suppose John bought a 25-cent paper, but that the clerk only realized after John had left that he had paid with a nickel. Rather, we will want to include a candidate event in, for example, the PRECONDITION slot of an event even if the candidate event only *usually* precedes (or is only *usually* necessary for) the occurrence of the event. This is represented in the Mikrokosmos ontology by using a Default facet. A Default facet for a slot takes, as a filler, conceptual information assumed to be true, unless explicitly stated otherwise. For example, given the information that Mary wants to drive her car somewhere and that she has entered her car and closed the door, we would like to make the default assumption that she has her car key with her. Yet, further information may reveal that she has forgotten it.

We can, of course, go one step further and include in the PRECONDITIONS slot conceptual information that only sometimes obtains when an event occurs. For example, "Use-Turn-Signal"[6] and "Adjust-Radio-Volume" might be listed in the HAS-PARTS slot for the complex event "Drive-Car," though they are neither logically necessary nor assumed to be true without explicit mention. Representing this conceptual information may be useful, for example, in generating a TMR for "Mary got in her car. She *buckled up* and drove away." Of course, those pieces of conceptual information listed in the PRECONDITIONS, HAS-PARTS, or EFFECTS slots that are only weakly implied by the occurrence of the complex event will need to be specially marked: they will need to be marked with a facet Optional.

It should be pointed out that the Optional facet is not currently available or axiomatized for the ontology. Because specifiying it for the ontology is not without its problems, a justification for its inclusion, as well as its axiomatization, is provided in section 3.3.2.

[6] Informal examples of events, such as this one, will be placed in quotation marks throughout this dissertation and do not have any status in the actual ontological representation of complex events. In fact, "Event + Case-Role" for the name of an EVENT concept masks what turns out to be a very difficult problem: variable binding. This is explained in section 3.5.2.

In summary, three types of candidate events are included in the PRECONDITIONS, HAS-PARTS, and EFFECTS slots. These are distinguished by the facet associated with each: a Sem facet identifies a slot-filler as logically necessary; a Default facet identifies a slot-filler as assumed to obtain, unless input to the contrary is presented, and an Optional facet identifies a slot-filler that is not asserted unless explicit, confirming input is encountered. Section 3.2.3 elucidates principles for deciding whether a given candidate event should be included in the PRECONDITIONS, HAS-PARTS, or EFFECTS slots and for deciding whether the candidate should then be marked with a Sem, Default, or Optional facet.

First, the following three subsections describe modifications to the EVENT frame slot structure specific to PRECONDITIONS, HAS-PARTS, and EFFECTS.

3.2.2.1 The PRECONDITIONS slot

The occurrence of any event generally depends on other things happening, or obtaining, for that event to occur. A PRECONDITIONS slot will represent conceptual information that is not within the scope of the event referenced by an EVENT concept, but rather is logically prior to it (cf. "enabling conditions" in Cullingford 1978).

Temporal order is generally a good cue in distinguishing whether a candidate should be included in either the PRECONDITIONS or HAS-PARTS slot. That is, candidate events that are preconditions generally precede, in time, the event referred to by the event concept. However, there are clearly cases where this is not so: for example, for an object to fall, gravity must obtain, yet gravity still obtains even during and after the falling event.

Still, the situation is complicated by the fact that natural language texts may not make clear-cut distinctions between events that are prior to the event in question and those that are subsumed by it. For example, it is not clear the extent to which (3a) is subsumed by (3b):

(3a) (i) As John walked down Main Street, (ii) he realized he hadn't checked his stock values lately. (iii) He ducked into the drugstore (iv) and took a newspaper off the rack. (v) He fished a quarter out of his pocket, (vi) then plunked the quarter and the paper on the counter. (vii) The clerk took the money. (viii) John walked out with his paper, (ix) and continued down Main Street.

(3b) John bought a newspaper.

In (3a), clauses (i), (ii), and (ix) seem linked to, but neither preconditions or effects of, the buying event. And, clauses (vi) and (vii) are good candidates for component events of the buy-event, i.e., of (3b). The status of (iii) and (iv), however, is unclear: they might be construed as either preconditions or component events of the buy-event, i.e, of (3b).

While this may be an interesting subject of an empirical study, classifying these borderline cases as either preconditions or component events will not matter to the system. In fact, the distinction between the PRECONDITIONS and HAS-PARTS is of importance only for (i) resolving time issues (i.e., a component event and a precondition may be posited to occur at Time(0) and Time(-1), respectively), and (ii) convenience of ontological development and maintenance.

Finally, there are two types of events that specifically are not represented in the PRECONDITIONS slot, namely existential presuppositions and modalities. Existential presuppositions are of the form: given "Bob threw the ball," it is presupposed that "Bob" and "ball" exist. The existential quantifier takes as scope the entire ontology, and so expression of such presuppositions within a frame is redundant. Second, modalities such as deonticity (“have to”) and volition (“want to”) are treated in the TMR shell and thus need not be expressed in the ontology.

3.2.2.2 The HAS-PARTS slot

The HAS-PARTS slot of an EVENT frame contains the component events of that EVENT frame. This requirement entails an adjustment to the current ontological hierarchy, since although the RELATION concept HAS-PARTS is defined as “the relation between an entity and its parts” (which would appear to cover a relationship between an event and its subevents), HAS-PARTS is currently defined only as a child of a PHYSICAL-OBJECT-RELATION, with its DOMAIN and RANGE fillers both being PHYSICAL-OBJECT. Therefore, to allow “has-parts” relationships between both objects-to-objects and events-to-events, a will have to create a new HAS-PARTS-EVENT slot, as a child of EVENT-RELATION and taking EVENT in its DOMAIN and RANGE slots. Furthermore, to prevent confusion in ontology maintenance, the current HAS-PARTS slot should be altered to HAS-PARTS-PHYSICAL-OBJECT. However, since this thesis is concerned only with the has-parts of events, HAS-PARTS-EVENT will be shortened to HAS-PARTS throughout.

Just as in the PRECONDITIONS slot, fillers of the HAS-PARTS slot may be specified for either Sem, Default, or Optional facets, depending on whether the candidate event is, respectively, logically necessary, assumed unless otherwise stated, or not assumed but possible and not unexpected. Moreover, deciding whether to include a candidate for the HAS-PARTS slot as Optional or to exclude it entirely from the knowledge base is subject to the same principles as candidates for the PRECONDITIONS slot. (Again, such principles are explained in section 3.2.3.) Of primary importance for the HAS-PARTS slot is temporal ordering of the component events—although temporal ordering is also relevant for the PRECONDITIONS and EFFECTS slots.

The component events of the HAS-PARTS slot are often ordered temporally. For example, in a buy-event, the buyer tendering money for merchandise generally precedes the buyer receiving change (although the author has on occasion received a 99-cent cup of coffee and a penny before proffering the dollar). However, some component events of an event may happen at any time during the course of the event. For example, a "Fundraising-Event" may happen at any time during the event referenced by "Political-Campaign." Such situations are handled with the PRECONDITIONS slot: for example, if Subevent-A is followed by Subevent-B, then Subevent-B will take Subevent-A as a precondition. This is represented schematically as (4):

(4) If Subevent-A ⇒ Subevent-B, then:

Frame	Slot	Facet	Filler
SUBEVENT-B			
	|		
	SUBEVENT-OF	Sem	EVENT-X
	PRECONDITIONS	Sem	SUBEVENT-B
	|		

If there is no ordering relationship, then subevents are simply listed in the HAS-PARTS slot of the event without specifying preconditions. (The SUBEVENT-OF slot is described in section 3.5.4, rather than in this section, since it is concerned with the arrangement of EVENT concepts in the ontological hierarchy.) The PRECONDITIONS slot of subevents listed in the HAS-PARTS slot of an EVENT concept is of course subject to the same requirements for preconditions explained in section 3.2.2.1.

3.2.2.3 The EFFECTS slot

The completion of an event often entails (or, usually precedes) the occurrence of one or more events. The latter are specified in the EFFECTS slot. Because there is little to add here beyond which has been described above, this section only briefly mentions the analogues. As with preconditions, it is sometimes difficult to discern whether a candidate event referred to by an event concept should be included in the HAS-PARTS or EFFECTS slot: for example should "Exit-Store" be listed as an effect of a component subevent of "Buy-Newspaper"? Again, it is argued that such borderline cases will not affect the quality of the output.

3.2.3 Principles to determine inclusion in PRECONDITIONS, HAS-PARTS, and EFFECTS slots

This section provides some principles to guide acquisition of candidate events into the PRECONDITIONS, HAS-PARTS, and EFFECTS slots of complex events. Given some complex event, Event-X, the first consideration should be determining whether a given candidate, Event-Y, should be a classified as either a candidate precondition, a candidate component subevent, or a candidate effect of Event-X. If within the boundaries of the event referred to by Event-X, Event-Y might also occur, then Event-Y should be considered a candidate component subevent. If Event-Y is prior to or may be a contributing causal factor of Event-X, then Event-Y should be considered a candidate precondition. Or, if Event-Y might occur after or may be a result of Event-X, then Event-Y should be considered a candidate effect.

The second consideration is to determine Event-Y's criteriality for Event-X, or the extent to which Event-Y may be assumed given Event-X. To aid in these determinations, we will draw on Cruse's (1986) *statuses of semantic traits*: criterial, expected, possible, and unexpected. We will furthermore equate criteriality with the Sem facet, expectedness with the Default facet, and possibility with the Optional facet. Making a slight modification to terminology in order to handle events, rather than words, we will say that a *criterial event* is one that is logically entailed by Event-X. Thus, if Event-Y (regardless of whether it is a candidate precondition, component event, or effect) is logically entailed by (i.e., always occurs when) Event-X, then it should be considered criterial and therefore marked with the Sem facet. Furthermore, an *expected event* is one that would be assumed to occur given the occurrence of Event-X. If Event-Y falls into this category, then it should be marked with the Default facet. Finally, it should be determined whether Event-Y is a possible event or an unexpected event given the occurrence of Event-X. If the former, then Event-Y should be marked with the Optional facet; if the latter, then Event-Y should not be included at all in Event-X, the complex event. These suggestions can be formalized in an algorithm presented in (5).

(5)

Given Event-X, the complex event, and Event-Y, the candidate event:

(a) Is Event-Y *within the boundaries of* the event referred to by Event-X?
If YES, then Event-Y is a candidate for Has-Parts; Go to (d).
If NO, then go to (b).

(b) Is Event-Y logically *before* Event-X?
If YES, then Event-Y is a candidate for Preconditions; Go to (d).
If NO, then go to (c).

(c) Is Event-Y logically *after* to Event-X?
If YES, then Event-Y is a candidate for Effects; Go to (d).
If NO, then FAIL; Event-Y should not be included in Event-X.

(d) Is the occurrence of Event-Y *criterial/necessary* for the occurrence of Event-X?
If YES, then Event-Y should be marked with a Sem facet and included in Event-X.
If NO, then go to (e).
(e) Is the occurrence of Event-Y *expected* given the occurrence of Event-X?
If YES, then Event-Y should be marked with a Default facet and included in Event-X.
If NO, then go to (f).
(f) Is the occurrence of Event-Y *possible* given the occurrence of Event-X?
If YES, then go to (g).
If NO, then FAIL; Event-Y should not be included in Event-X.
(g) Is the occurrence of Event-Y *unexpected* given the occurrence of Event-X?
If YES, then FAIL; Event-Y should not be included in Event-X.
If NO, then Event-Y should be marked with a Default facet and included in Event-X.

Though (5) may serve as a helpful guide for ontological knowledge acquirers, the final judgment will, of course, need to be made by them.

3.3 Representing the paths that a complex event may take

In the course of the occurrence of an event, many things can happen; an event may take many paths (cf. "tracks" in Schank and Abelson 1977 and Cullingford 1978). These event-paths, of course, do not simply conform to a temporal, linear order, but rather are more accurately viewed as a network of subevents arising because of situational factors, or contingencies. These factors might be conditionality ("if…then"), branching ("either X or Y"), default and optional subevents, and even reiterations (i.e., "loops"). For example, the event "ATM-Withdrawal" might include:

- the user will receive the amount of cash requested if the balance in the account is at least as much (conditional event);
- the user must select whether to withdraw from a checking or savings account (branching event);
- the user usually receives a receipt upon completion of the transaction (default event);

- the user may make another transaction (optional event);
- the user must continue to enter a PIN until it matches the PIN associated with her card (looping event).

This section explains how the formalism in this dissertation handles such situations, drawing primarily on the PRECONDITIONS/EFFECTS slots, as well as the (new) Optional facet. The representation here does not rely on temporal ordering. Instead, subevents are ordered with respect to preconditions and effects on their occurrence. The TMR shell handles actual temporal ordering.

3.3.1 Default subevents

A Default subevent is one that is assumed to take place, unless explicit contradictory input is encountered. Such events are marked with the Default facet, which is already specified for the ontology. In contrast, a subevent necessarily occurring in the complex event is marked with the Sem facet. Although the Default facet is not used much throughout the rest of the ontology, it plays an important role in the representation of complex events. The real-world events referenced by Default subevents turn out to be, in natural language texts, commonly both *implicit* and *necessary for* accurate inferencing, as will be demonstrated in Chapter Four. Thus, it is especially important to include Default subevents in complex events.

3.3.2 Optional subevents

Here, we motivate the creation of a new facet, Optional, to represent "possible and not unexpected information." The utility in specifiying optional subevents in complex events has been described in section 3.2.2; and, section 3.3.5, will explain their utility in expressing branching (i.e., "either…or") subevents. Furthermore, since acquiring optional entities entails much more sensitivity during the acquisition process than adding criterial or default entities, a formal procedure for adding optional entities has been described in section 3.2.3. Finally, the Optional facet would appear to complement the already existing Sem (criterial information) and Default (assumed information) facets. Given the putative Optional facet's utility, formalizability, and consistency with ontological theory, the rest of this section is devoted to discussing the consequences of adding it to the ontological repertoire.

The ontology is currently an "And-Graph," in which a child concept inherits all the slots of its parent concept and in which a frame is a conjunction of its slots. In the event that a child concept should not inherit a slot in its parent concept, that slot can be filled with the special symbol, *nothing*. This simplifies the ontological hierarchy in instances when, say, two siblings should inherit all parent slot and a third sibling shares all but one. The alternative is to make the third child concept a sibling to the parent, thus duplicating much information and creating unnecessary (and counterintuitive) branching. However, the ontology currently provides no middle ground between "something and *nothing*." That is, a frame may not assert a slot-filler only sometimes, as might be useful for a filler of a "Number-of-Doors" slot for AUTOMOBILE. This creates ontological inheritance problems, as shown in (6).

(6)

Frame AUTOMOBILE

Slot	Facet	Filler
DEFINITION	Value	"Small vehicles for carrying people"
IS-A	Value	WHEELED-ENGINE-VEHICLE
SUBCLASSES	Value	SEDAN, STATION-WAGON, COUPE
NUMBER-OF-WHEELS	Sem	4
OPERATED-BY	Default	DRIVER
"Number-of-Doors"	Optional	4

In this example, children of AUTOMOBILE (e.g., SEDAN, STATION-WAGON, COUPE) will also inherit "Number-of-Doors" slot with the Optional filler "4." This, in turn, precludes the ontology from specifying a different slot-filler, with the Sem or Default facets, for the children, e.g., "5" for Station-Wagon or "2" for COUPE. Allowing the children to inherit *nothing* for "Number-of-Doors" also precludes specifying a different slot-filler. Such information may, of course, be useful. This will not help for representation of complex events either, since they also may have children (i.e., SUBCLASSES).

There are two possible remedies to the inheritance problem. First, the Optional facet may be defined only for leaf-node concepts in the ontology. In other words, only concepts with no children would be allowed to specify a slot with the Optional facet. This can be axiomatized in predicate calculus, as in (7):

(7) Given T as slot, and concepts X, Y, and Z,
$slot(X, T, \text{Optional}, Y) \Rightarrow \neg(\exists Z \wedge slot(Z, \text{Subclasses}, \text{Value}, X))$

Thus, given that *slot* is a four-place predicate taking the arguments frame, slot-name, facet, and filler, then if X is a frame that takes, in any slot W, the frame Y as a filler specified with the Optional facet, then there must be no frame Z that is in the the subclasses of X. (See Mahesh 1996 for the current axiomatization of the ontology.) However, such a torturous axiomatization is counter to the ontological principles in the

Mikrokosmos system and in fact will cause serious problems if a leaf-node is later given child concepts.

The preferred solution is to specify that Optional facets are not inherited. This is consistent with the current ontological framework, in which Sem facets are inherited and Default facets are not. Furthermore, although a slot taking an Optional filler in a parent concept would not be inherited by any child concepts, that slot would still be available for use in the child concept, if needed. One other requirement needs to be mentioned. If a slot in a parent concept is specified with a filler marked with the Sem facet, then a filler marked with the Default facet for that same slot in any children concepts must be a subclass of the parent concept's slot-filler. In predicate calculus, this is rendered as (8):

(8) Given T as slot, and concepts W, X, Y, and Z,
slot(W, T, Sem, X) ∧ slot(Y, T, Default, Z) ∧ *slot*(Y, Is-A, Value, W) ⇒
slot(X, Is-A, Value, Z)

This provides an intuitive set-theoretic relationship between Sem and Default fillers: a Sem filler is included in the scope of a Default filler. An Optional facet can then be viewed as being within the scope of the Default filler. The relationship between Sem, Default, and Optional facets is represented, in (9), in set-theoretic terms. Incidentally, (10) is also well-formed, but is not axiomatized in the ontology; rather, it seems generally true for the *world*.

(9) Given frames X, Y, and Z,
$\text{Sem}(X) \subseteq \text{Default}(Y) \subseteq \text{Optional}(Z)$

(10) Given frame X,
$\text{Sem}(X) \subseteq \text{Default}(X) \subseteq \text{Optional}(X)$

In summary, because the Optional facet can be limited so as to be non-inheritable, the primary objection to its inclusion as an ontological constant has been countered. Its availability is therefore assumed throughout the rest of this dissertation.

3.3.3 Conditional subevents

A conditional subevent is one whose occurrence depends on some other subevent occurring. Such sequences of subevents are represented by making use of the PRECONDITIONS and EFFECTS slots. Thus, if Subevent-A is contingent on Subevent-B happening first, then Subevent-A will list Subevent-B in the PRECONDITIONS slot. Or, if the occurrence of Subevent-C entails the occurrence of Subevent-D, then Subevent-C will list Subevent-D in the EFFECTS slot. Finally, mutual entailment ($E \Leftrightarrow F$) is representable as well: in this case, Subevent-E will list Subevent-F in its EFFECTS slot, and in turn Subevent-F will list Subevent-E in its PRECONDITIONS slot.

3.3.4 Looping subevents

It is sometimes the case that, during the course of a complex event, a subevent may occur recursively, or may be looped, until some criterion is met. For example, take a driver's test or enter PIN. It is generally difficult to formalize criteria for a subevent's failure and subsequent looping, since the initial precondition(s) of the subevent's occurrence are sometimes different from the subevent's looped occurrence and since the ending effects of the two (or more) occurrences may be different as well. One try is to do the following:

(11)

Frame: "Take-Driver-Test-1"[7]

Slot	Facet	Filler
PRECONDITIONS	Optional	"Fail-Test"
HAS-PARTS	Default	"Sign-Up"
	Sem	"Start-Car"
	Sem	"Drive-Car"
EFFECTS	Optional	"Pass-Test"
	Optional	"Fail-Test"

Frame: "Fail-Test"

Slot	Facet	Filler
PRECONDITIONS	Sem	"Take-Driver-Test-1"
EFFECTS	Optional	"Take-Driver-Test-2"

Thus, an event's "looping condition" can be specified as a sub-subevent ("Take-Driver-Test-2") in the EFFECTS slot of a subevent ("Fail-Test") of the event itself ("Take-Driver-Test-1). We then mark the looping condition with the Optional facet to prevent it from applying infinitely many times. A true looping subevent should be distinguished from: (i) an event that appears to be marked as "reiterative" by the lexical item(s) referencing that event: e.g., "to hammer (a nail)" entails repeated swinging of the hammer until the nail is driven (or hopelessly bent). Such exceedingly fine-grained cases are handled in the TMR shell, using an Iteration value. And, (ii) a subevent that may simply happen more than once during the course of a complex event, without any apparent connection the multiple instances: e.g., a speech during the course of a political campaign, or grading essays in the course of teaching a class. In these cases, the subevent should simply be listed once. Then, if that subevent is encountered in the input again, semantic analysis determines whether the text refers to the first or to a new instance.

[7] The indexes appended to the two occurrences of "Take-Driver-Test" are included here merely for the sake of convenience. They are not required in the actual complex-event representation.

Finally, we will not want to bother representing information that could possibly happen over and over again, but that would be unexpected to do so. (Cf. the principles for including subevents in complex events in section 3.2.3.) For example, the subevent "Give-Money," as part of a "Buy-Event," might be said to loop until the amount of money given equals the cost of the merchandise. (Imagine a buyer offering penny after penny to pay for a cup of coffee.) Representing such information would cause more harm than good.

Looping subevents that meet these criteria turn out to be quite rare, and are anticipated to be of little use in the representation of complex-event knowledge.

3.3.5 Branching subevents

Branching subevents are of the type "either X or Y." For example, in an election, voters may vote for or against someone/something; or, in the course of an ATM withdrawal, a person may elect to withdraw from a savings or checking account. Such contingencies are difficult to represent given the current ontological machinery, primarily because frames are not designed to accept "Or" statements, only "And" conjunctions. That is, representing a branching structure in which branches later feed into a common subevent is difficult, since this common subevent will need to take both branches as Optional preconditions, meaning of course that neither precondition is binding. However, branching subevents can sometimes be represented as in the following (where "Enter-Savings-Money" = "enter the amount to be withdrawn from the savings account"):

(12)

Frame: “ATM-Withdrawal”

Slot	Facet	Filler
PRECONDITIONS	Sem	“Have-ATM-Card”
HAS-PARTS	Sem	“Insert-Card”
	Sem	“Enter-PIN”
	Sem	“Select-Account”
	Sem	“Enter-Money”

Frame: “Select-Account”

Slot	Facet	Filler
EFFECTS	Optional	“Select-Savings-Account”
	Optional	“Select-Checking-Account”

Frame: “Select-Savings-Account”

Slot	Facet	Filler
EFFECTS	Sem	“Enter-Money”

Frame: “Select-Checking-Account”

Slot	Facet	Filler
EFFECTS	Sem	“Enter-Money”

Frame: “Enter-Money”

Slot	Facet	Filler
PRECONDITIONS	Sem	“Select-Account”
	Optional	“Select-Savings-Account”
	Optional	“Select-Checking-Account”

That is, in the course of a complex event (e.g., “ATM-Withdrawal”), the component subevents may branch (e.g., “Select-Savings-Account” and “Select-Checking-Account”). These branching subevents may be specified as Optional subevents in the EFFECTS slot of a “matrix” event (“Select-Account”). Furthermore, these branching subevents can both specify, in their respective EFFECTS slots, a further subevent (“Enter-Money”), which may in turn then specify the matrix event as a criterial precondition, and even may specify the branching subevents as Optional preconditions.

Though a "matrix event" is a rather tenuous concept, it does serve as an "anchoring" for future contingent subevents. Furthermore, something like it may be necessary because of the difficulty of representing states in the ontology.

3.4 Situating complex events in the ontological hierearchy

This section explains how complex events are situated in the ontology. The ontology makes no theoretical distinction between a complex event concept, with PRECONDITIONS, HAS-PARTS, and EFFECTS slots, and an EVENT concept without them. The former simply has a more richly specified frame structure than the latter. Moreover, each complex event is situated in the ontology, as are all EVENT concepts. As such, complex events are subject to all ontological axioms defining the structure of EVENT concepts, e.g., inheritance and subclass relationships.

Each complex event may comprise any number of subevents. Furthermore, these subevents may comprise any number of sub-subevents. Although in principle the number of subevents and subsubevents for a complex event is unlimited, there are generally less than 15 of each per complex event. A schematic for an abbreviated complex event frame, showing the relevant features, is presented in (13).

```
(13)    Event-X
                HAS-PARTS:    Subevent-A
                                      HAS-PARTS:    Sub-subevent-1
                                                    Sub-subevent-2
                                                    |
                              Subevent-B
                              |
        Event-Y
        |
```

Note that subevents and sub-subevents may also appear in the PRECONDITIONS and EFFECTS slots.

Subevents and sub-subevents are, in their own right, EVENT concepts in the ontology, much like, for example, a PROPERTY slot of an OBJECT concept frame is also a concept under the PROPERTY branch in the hierarchy. Thus, the terminology "subevent" and "sub-subevent" is frame-specific. For example, Sub-subevent-1 of Event-X may itself be a complex event in the ontology, with subevents and sub-subevents of its own. Furthermore, Sub-subevent-1 may be a subevent of sub-subevent of Event-Y. Arranging complex event knowledge in this non-modular, interconnected fashion simplifies the knowledge acquisition process by decreasing redundancy. For instance, suppose Event-X is "Attend-Baseball-Game" and Event-Y is "Attend-Musical." Both will make use of the subevent "Buy-Ticket." Therefore, it makes sense to represent the information only once, rather than list "Buy-Ticket" separately for each complex event for which it is relevant. This arrangement is much like Schank's (1983) notion of MOPs (memory organization packets). However, while for Schank the advantage of allowing information to be "shared" between knowledge structures was primarily its apparent psychological validity, here we argue that the main advantage is that this allows complex event knowledge to be interconnected in a semantic network fashion, to facilitate inferencing. (Of course, we do not deny that this may be so by virtue of the psychological validity of the arrangement of knowledge.)

3.5 Variable binding for subevents in complex events

3.5.1 The relationship between subevents and complex events

Events in the world that are complex are quite sensitive to context; this much is obvious. From the standpoint of the ontology, this creates an enormous problem for *variable binding*, or co-reference relationships, of slot-fillers (which, as conceptual entities, have the status of variables). That is, it is generally the case that slot-fillers of a subevent contained within a complex event are dependent on that complex event. An example is shown in (14).

(14)
(14a) "Attend-Baseball-Game"
PRECONDITIONS: "Buy-Ticket"
|
HAS-PARTS: *"Select-Seat"*
"Tender-Money"
"Receive-Ticket"
"Receive-Change"
|
|
|

(14b) "Attend-Fair"
PRECONDITIONS: "Buy-Ticket"
|
HAS-PARTS: "Tender-Money"
"Receive-Ticket"
"Receive-Change"
|
|
|

In (14a) and (14b), the subevents of the precondition "Buy-Ticket" change depending on which concept contains them. In (14a), "Select-Seat" is a component subevent of "Buy-Ticket" for the event "Attend-Baseball-Game; however, in (14b), this information has been omitted from the precondition "Buy-Ticket" of the event "Attend-Fair," since one does not normally does not sit down, but rather walks around visiting prize hogs, lemon shake stands, and the like.

Although it is desirable to allow the complex event knowledge base to be sensitive to these differences, the arrangement in (14a) and (14b) creates a problem for frame representation in the ontology, since each of the slots in a frame must be asserted for that frame. Furthermore, allowing the structure of sub-subevents to change depending on the EVENT frame in which they are asserted defeats the purpose of an interconnected conceptual hierarchy, since it will result modular complex event frames incapable of drawing on conceptual information outside the frame to aid in inferencing.[8] Thus, it is unacceptable to instantiate "Buy-Ticket" with the Has-Parts slot-filler "Select-Seat" sometimes yes and sometimes no. However, there is an alternative in this case. When a subevent is at least "possible," and is furthermore "not unexpected" (cf. section 3.2.3), it can be marked with the facet Optional, such that "Buy-Ticket" would look like:

(15)
Frame: "Buy-Ticket"

Slot	Facet	Filler
AGENT	Sem	HUMAN
THEME	Sem	TICKET
ACCOMPANIER	Sem	HUMAN
INSTRUMENT	Sem	MONEY
HAS-PARTS	Optional	"Select-Seat"
	Sem	"Tender-Money"
	Sem	"Receive-Ticket"
	Optional	"Receive-Change"

"Buy-Ticket" could then be inserted into the PRECONDITIONS slot of either "Attend-Baseball-Game" or "Attend-Fair." Yet, this solution turns out to be entirely inadequate.

[8] It is possible, however, for a frame's slot-filler to prevent inheritance of that filler from a parent concept, using the special symbol, *nothing*. This symbol cannot be used to block inheritance of just one (of many) fillers of a slot.

3.5.2 The variable binding-problem

The problem is that the frame structure of EVENT concepts is very general. And so, when they are specified as subevents in a complex event, they are "blind" to both the semantic case-roles and the ordering relationships associated with the complex event. Thus, in actual practice, an event such as "Select-Seat" is much too specific to be contained in the ontological hierarchy. If we allowed such a concept—say, as a child of a "Select-Event," then the door is opened to "Select-Clothes," "Select-Map-Route," "Select-Mate," etc., or even to EVENT concepts that are specified for both a THEME role and an AGENT role, e.g., "Woman-Select-Clothes-Event." The absurdity of following this path should be obvious. Furthermore, there will rarely, if ever, be two EVENT concepts that are implicitly ordered in the way that is necessary to represent the information referenced by a complex event. Thus, while Subevent-A may be a necessary precondition of Subevent-B within Event-X, we cannot simply specify, in the frame for Subevent-B, that it takes Subevent-A as a precondition, since this will not be *generally true*. We are left with, at this point, a rather paltry and uninformative complex event, such as (16):

(16)
Frame: "Buy-Ticket"

Slot	Facet	Filler
DEFINITION	Value	"to buy a ticket (using money)"
IS-A	Value	BUY-OBJECT
AGENT	Sem	HUMAN
THEME	Sem	TICKET
ACCOMPANIER	Sem	HUMAN
INSTRUMENT	Sem	MONEY
HAS-PARTS	Optional	SELECT
	Sem	GIVE
	Sem	RECEIVE
	Optional	RECEIVE

Note that the inclusion of this particular EVENT concept itself would be difficult to justify for the ontology. There are two ways to get the required knowledge into the complex event: either instantiate the subevents entirely from within the complex event, or augment the schema in (16) with variable bindings specified in the complex event. First, it is argued that the former representation should be rejected.

3.5.3 A rejected solution to the problem of variable-binding

Instantiating subevents entirely from within the complex event leads to modular representation separate from the rest of the ontological hierarchy. Thus, in a "Buy-Ticket" complex event, instead of specifying generic EVENT concepts from the ontology, new EVENTS would be instantiated with variable-bindings in place. This would allow subevents named, for example, "Select-Seat," "Tender-Money," and "Receive-Change," or even "Select-Seat-Baseball-Game." Furthermore, since these very specific events are specified based only on the requirements of the complex event in question and are not situated in the EVENT hierarchy, we escape the (theoretical) need to acquire all—perhaps millions—such events. However, serious several shortcomings argue against this approach. First, since an events like “Select-Seat-Baseball-Game” have no status in the ontological hierarchy, they will have to be created entirely anew. Creating perhaps 20 such event frames per complex event is particularly costly. (For reference, there are about 7600 concepts in the ontology currently.) Furthermore, because the lexicon module provides mappings of lexical items to concepts, entirely new mappings will need to be created, entailing a prohibitively high cost: for instance, the lexical item “choose” is currently mapped to the SELECT concept, but would now need to be mapped to concepts like “Choose-Seat,” “Choose-Clothes,” “Choose-Map-Route,” etc. Little of this acquisition could be automated to save time and effort.

Moreover, this approach turns out not to have the characteristic “effective, but costly.” Rather, because these newly created frames will not participate in IS-A relations in the ontology, it will be difficult to use satisfaction of selection constraints to resolve ambiguities. To see why, consider the following: suppose the system encounters the sentence “Bob chose a seat from the diagram.” A question that might be posed is: does the system understand that Bob has expressed an intention here? That is, can the system designed in this way discern between Bob’s active and passive involvement in a scene? The answer is no. The lexical item “choose” is mapped to the ontological concept SELECT, which is linked (through an IS-A slot) to the concept INTEND, which expresses information about intention. But because “Choose-Seat” is not linked in this way (i.e., it has no IS-A slot since it is not a member of the ontological hierarchy), the required information cannot be inferred. Ultimately, then, constructing complex events in this way leads to dead-end concepts that do not participate in the inferencing process.

3.5.4 Proposals in Carlson and Nirenburg (1990) and Mahesh (1996)

Carlson and Nirenburg (1990) suggested a different way of handling the difficult problem of variable binding that eliminates some of the problems with the above approach, by positing the "hybrid" entity *ontological instance*. While the ontology catalogs conceptual entities and the onomasticon catalogs instances of those concepts (i.e., where a concept is constrained to refer to one unique entity in the world), an ontological instance has properties of both: it is conceptual in nature because it refers to a class of conceptual entities, not to one instantiation of it; and, it is "instancey" in nature because it severely constrains the class of conceptual entities it refers to. Thus, instead of creating a concept frame like "Select-Seat" entirely anew and situating it entirely within the complex event, as in the rejected proposal above, Carlson and Nirenburg (1990) suggest placing it in the ontological hierarchy as child to the parent concept (i.e., "Select-Seat" would be specified for an IS-A slot with filler SELECT), and specifying this ontological instance with the variable bindings necessary for the complex event.

Mahesh (1996) offers further elaboration of the new machinery required to handle variable binding in this way. Subevents (= ontological instances) are represented as orphans (i.e., they have no IS-A slot) and instead are provided with the (new) ONTO-INSTANCE-OF ("ontological instance of") slot linking these subevents to their parent concepts. For example, "Select-Seat" would have the ONTO-INSTANCE-OF slot filled by SELECT. Furthermore, subevents would be provided with the (new) SUBEVENT-OF slot linking them to the complex events in which they take part. Again, these subevents would be specified with all the relevant knowledge (preconditions for their firing, co-reference relations to other slot-fillers within the complex event, etc.). Thus, within the complex event, only subevents would need to be specified.

Problems with this proposal, all having to do with the difficulty in ontology development and maintenance, and a modified proposal, which eliminates these problems, is explained in the following section.

3.5.5 The solution offered here

In order to maintain the sharp distinction in the ontology between events and instances, it is proposed that all variable-binding information be contained within the complex event. Therefore, a new slot is introduced, BINDING-ROLE.EVENT.SLOT (which will be abbreviated to BR.EVENT.SLOT), in which to specify variable-binding information. (The "dot-notation" (e.g., Select.Theme) identifies the frame-slot whose filler should be co-referenced; cf. Carlson and Nirenburgh (1990) and Mahesh (1996)). This will also help minimize the proliferation of frames in the course of large-scale complex-event acquisition. The trade-off is a proliferation of slots within the complex event frame; however, this will be easier to maintain than a proliferation of frames. Therefore, each subevent of a complex event maintains its status in the ontological hierarchy and is "imported" into the complex event frame intact. The mechanism for doing so is described below.

First, because a subevent introduced into a complex event may exhibit contingencies with *other subevents* in the complex event, all EVENT concepts must be provided with PRECONDITIONS, HAS-PARTS, and EFFECTS slots to make them available for variable-binding relationships; if necessary, these slots may take the special *nothing* filler. Next, when the subevent is specified in the complex event—either in the PRECONDITIONS, HAS-PARTS, or EFFECTS slot—it will be marked with a facet specifying its criteriality/ expectedness for the complex event. Then, for each subevent slot marked with a Default, Sem, Measuring-Unit, or Optional facet,[9] a BR slot will be instantiated in the complex event frame.[10] Variable-binding information is then specified within the complex event's BINDING-ROLE slots, which are all specified for the Default facet, meaning that they will be assumed unless contrary input is received. This allows the subevent's slot-fillers to maintain their Sem facets as well, which may aid in inferencing. An example of a subevent and a complex event containing that subevent is presented in (17a) and (17b).

(17a)
Frame: SELECT

Slot	Facet	Filler
DEFINITION	Value	"to make a choice"
IS-A	Value	INTEND
AGENT	Sem	HUMAN
THEME	Sem	EVENT, OBJECT
PRECONDITIONS	Default	*nothing*
HAS-PARTS	Default	*nothing*
EFFECTS	Default	*nothing*

(17b)
Frame: "Buy-Ticket"

Slot	Facet	Filler
DEFINITION	Value	"to buy a ticket"
IS-A	Value	BUY-OBJECT
AGENT	Sem	HUMAN
THEME	Sem	TICKET
ACCOMPANIER	Sem	HUMAN
INSTRUMENT	Sem	MONEY
HAS-PARTS	Optional	SELECT

[9] That is, slot-fillers marked with the Value facet remain the same and will not engage in variable binding.
[10] Although this may seem rather messy, instantiation of the relevant BR slots can be automated. This is explained in section 3.5.6.

BR.SELECT.AGENT	Default	Buy-Object.Agent
BR.SELECT.THEME	Default	SEAT
BR.SELECT.PRECONDITION	Default	ATTEND
BR.SELECT.HAS-PARTS	Default	*nothing*
BR.SELECT.EFFECTS	Default	*nothing*

Each subevent included in a complex event will need to be specified with PRECONDITIONS-OF, HAS-PARTS-OF, and EFFECTS-OF slots to denote which complex events it participates in, since the axiomatization of the Mikrokosmos ontology requires that fillers of PROPERTY slots have, in their frames, a PROPERTY-OF slot expressing the inverse relationship. However, it seems wise to delay this until the number of complex events is reasonably high, since such information is likely to bias the inferencing process.

One as yet unresolved problem with the above is the status of the BINDING-ROLE slots themselves. Slots are generally defined as conceptual entities in the ontological hierarchy: they are PROPERTYs. Thus, creating slots would actually seem to entail creating full-blown frames for each of them, thereby running into the same "frame proliferation" problem discussed in conjunction with Carlson and Nirenburg's (1990) and Mahesh's (1996) proposals. We avoid this problem by defining BINDING-ROLE as a "special slot," which, like IS-A, SUBCLASSES, DEFINITION, and others, is not defined as a PROPERTY in the ontology. Though this may at first appear to be a rather *ad hoc* move, the BINDING-ROLE's "special slot" status can be justified as follows. Suppose that X is a slot in a complex event and that Y is a slot in one of the subevents of the complex event, and that we want to bind Y to X, i.e., that Y = X. Since both X and Y are already defined in the ontology as RELATIONS, a subclass of PROPERTY, creating a new RELATION, Z, to express the relationships X = Z and Y = Z is superfluous. The ontology does not require RELATION concepts to express relationships between two RELATION concepts.

Finally, it is entirely possible that a complex event will later become a subevent in a complex event. The proposal developed in this section supports this. Because the complex-event-turned-subevent has the necessary variable-bindings specified as slots and fillers, this information will be automatically imported to the new complex event.

In summary, the primary advantages of this proposal, over the ones proposed in Carlson and Nirenburg (1990) and Mahesh (1996), are that:

- it maintains the usual sharp distinction between concepts and instances crucial to the understanding, development, and maintenance of the ontology, and avoids the theoretically slippery notion of an “ontological instance”;
- it chooses proliferation of slots over the much more problematic proliferation of frames;
- it intuitively specifies binding roles in the complex event frame in which they are relevant, rather than in another frame linked to it;
- it maintains the original slot-fillers of the subevent, with Sem facets, which may then be used in inferencing;
- it requires only one new slot, BINDING-ROLE, versus four, ONTO-INSTANCE, ONTO-INSTANCE-OF, SUBEVENT, and SUBEVENT-OF, in the Carlson/Nirenburg/Mahesh approach;
- it requires less modification to current ontological development/maintenance tools: all variable binding takes place in one frame, the complex event frame.

The next section briefly outlines how ontological development/maintenance tools might be enhanced to facilitate the otherwise confusing and messy acquisition of complex events.

3.5.6 Automating acquisition of complex events

A reasonably rich complex event may easily require upwards of 50 BINDING-ROLE slots. Given this large number, given the fact that variable-binding information will be entered as disjoint from the subevent added to the complex event, and given the fact that determining contingency relations between component subevents is taxing itself, this section offers some brief suggestions for what might be automated to simplify the acquisition process.

The Mikrokosmos Knowledge Acquisition Editor (MKAE) currently automates much of the acquisition process. A few new features might also aid in the acquisition of complex-event knowledge. For example, when displaying a frame, the MKAE will need a "Add New Subevent" command. Choosing this option should then call up a list of available EVENT frames. Once the appropriate frame has been chosen, the MKAE may then automatically instantiate BINDING-ROLE slots, within the complex-event frame, for each relevant subevent slot. Since fillers of variable bindings will still be required to fall within the scope of the subevent's original slot-fillers, the MKAE might confirm that the concept entered into the BINDING-ROLE filler is a child concept to the original slot-filler. Finally, the subevent frame should be viewable alongside the complex-event frame for comparison.

3.6 Heuristics for complex-event knowledge acquisition

This section outlines heuristics to aid in the acquisition process of complex-knowledge. The following questions are answered: To what extent will the current ontological hierarchy need to be modified to represent complex-event knowledge? (section 3.6.1); What should be added to a given complex event and what should not be added? (section 3.6.2); and, In what order should complex events be developed? (section 3.6.3).

3.6.1 Modifications to the ontology

Previous sections have outlined some necessary changes to the ontology that must be made in order to represent complex-event knowledge. Briefly, these are: adding a HAS-PARTS-EVENT as a RELATION concept, adding the Optional facet, and adding the "special slot" BINDING-ROLE. Of course, questions about whether to add, delete, or modify an existing concept will arise, as the ontology is always being revised. Complex-event acquisition will be no exception and will have to address the following specifically:

(i) Is the repertoire of EVENT concepts sufficient to represent subevents in a complex event?
(ii) Should an EVENT concept be developed as a complex event itself, or should it be "broken down" into children concepts which should then be developed as complex events?

Though the ontology is still under development, the answer to (i) is that the repertoire of EVENT concepts does indeed appear to be sufficient to represent subevents. As will be demonstrated in Chapter Four, the complex events developed thus far have not required the creation of *any* new EVENT concept as a subevent.

Concerning (ii), it is, however, often the case that developing a complex event motivates splitting an EVENT concept into two or more children concepts, which are then each developed as complex events. The reason for this is that although representation does allow for contingency relationships between events, this can often become awkward when there are many branches. If possible, the parent concept is then developed as a complex event specifying the knowledge shared by the children concepts. The concept BANKRUPTCY is a good example. The dichotomies between individual/corporate and Chapter 7/Chapter 11 (putting aside, for now, the less frequent other kinds of bankruptcy) necessitate four different branches of subevents within the single complex event BANKRUPTCY. Though this would be technically feasible, it would also be a nightmare for acquirers, because of all the preconditions and effects that would need to be specified. Instead, Bankruptcy has been broken down into more manageable chunks: its subclasses are BANKRUPTCY-INDIVIDUAL[11] and BANKRUPTCY-CORPORATE. Then, for example, BANKRUPTCY-CORPORATE has been further broken down into BANKRUPTCY-CORPORATE-CHAPTER-SEVEN and BANKRUPTCY-CORPORATE-CHAPTER-ELEVEN.

It should be no surprise that the EVENT concepts most amenable to development as complex events (and most useful to a given NLP task) occur far down on the ontological hierarchy, many of them leaf-nodes. Certainly, a complex event for MENTAL-EVENT seems unwarranted. Moreover, component subevents themselves are rarely more than one or two nodes up from the leaf-node concept (i.e., the terminal node of an branch) in their respective branches, as will be seen in Chapter Four. While this may not be of any theoretical significance, it can point the acquisition process in the right direction.

3.6.2 What information to add to a complex event

While section 3.2 provides guidelines to help acquirers navigate the difficult process of deciding what information to add to a complex event (based on criteriality, expectedness, and unexpectedness), two other considerations are discussed here.

First, there is the consideration of "grain-size": how deep should a given complex event go? That is, should we include sub-sub-subevents? sub-sub-sub-subevents? The

[11] Since BANKRUPTCY is a child of CORPORATE-EVENT, the new BANKRUPTCY-INDIVIDUAL has been assigned (with apologies) to the subclasses of EVERYDAY-FINANCIAL-EVENT.

answer to this question depends, of course, on the requirements of the task this knowledge is used for. In practice, the complex events developed thus far generally use the sub-subevent as the cutoff point, with only occasional representation of sub-sub-subevents when necessary. The cutoff point also depends on the magnitude of the complex event being considered. While the complex events developed thus far have concentrated on those EVENT concepts near leaf nodes, representing very complex events, such as “Political-Campaign,” “War-Event,” or even “Human-Life-Event,” may require much greater depth. Of course, granularity will still be a factor: complex events should only be developed based on need, not on possibility.

Second, the knowledge to be represented in complex events is very culturally specific—at least as specific as lexical entries. For example, the structure of the complex event BANKRUPTCY-CORPORATE-CHAPTER-ELEVEN is peculiar to the laws of the United States and will not generalize well to other cultural settings. The same care taken in lexical acquisition will be needed for complex-event knowledge acquisition. This contrasts with the usual process of ontology acquisition, which, since the ontology is language-neutral, requires much less familiarity with cultural idiosyncrasies.

3.6.3 Order of acquisition

Given that acquisition of complex events will be an immense task, it makes sense to consider whether there is any preferred order of acquisition. There are two parameters to consider: (i) should all leaf nodes be developed as complex events before moving to the second-to-last node (a reversed “breadth-first” strategy), or should acquisition start at the leaf node of one branch and then move up that branch (a reversed “depth-first” strategy)?; and, (ii) should acquisition start with complex events with only, at most, sub-subevents ("breadth-first"), or should acquisition begin by representing very complex events with considerable depth ("depth-first")? The very simple answer is that neither will matter a great deal; breadth-first or depth-first acquisition is supported for either the branch or subevent parameter. As mentioned in section 3.6.1, EVENT concepts might be broken down into children concepts and then each developed, in clustered fashion, as complex events. Beyond this caveat, the leaf-nodes-first and the branches-first acquisition order are equally feasible. In fact, there appears to be no obstacle to developing first those EVENT concepts higher up on an ontological branch. Furthermore, acquisition could begin by developing a very complex event, such as "War-Event," to many-levels deep. However, it may be more intuitive to use shallower complex events as "building blocks" for complex events.

In summary, complex event knowledge is not sensitive to order of acquisition. Neither leaf-node-first, branches-first, shallow-event-first, nor very-complex-event-first methods should have a significant effect on the quality of the knowledge base. Thus, acquisition of complex-event knowledge may proceed based almost wholly on the particular needs of the task to be performed.

3.7 Conclusion

By way of concluding this chapter, this section first reviews some usual criticisms of the representation of complex-event knowledge and how the formalism developed in this chapter meets those criticisms (section 3.7.1), and then provides a summary of the chapter.

3.7.1 Addressing prior criticisms to complex-event knowledge representation

Two usual criticisms to the representation of complex-event knowledge—brittleness and the related problem pattern matching/following insufficiency—are answered by the Mikrokosmos system as a whole, not simply by its representation of complex events. Brittleness refers to a system's inability to handle novel, unstereotypical input. Both the SAM (Cullingford 1978) and Ms. Malaprop (Charniak 1977, 1979) programs are equipped only with knowledge of complex events without methods for handling input not specified in their respective knowledge bases. Holland *et al*. (1986) and Dreyfus (1985), among others, have noted the folly of trying to build semantic representations based solely on complex-event knowledge. In contrast, representations of knowledge of complex events in this dissertation are supported by the Mikrokosmos system's semantic and syntactic analyzers. Thus, when input text matches complex-event knowledge patterns, the requisite inferences can be drawn; and, when input text goes beyond the scope of complex-event knowledge, semantic and syntactic analysis proceeds without it.

A second but related problem with previous systems is their inability to provide robust pattern matching and following. In Schankian systems, this has primarily to do with the coarseness of Conceptual Dependency representation. That is, matching patterns in the complex-event knowledge base requires that input meaning representations be much more fine-grained than CD allows. Thus, patterns may fail to be activated/asserted not because the input text is nonstereotypical (and hence outside the scope of the complex-event knowledge base) but because the meaning representation extracted from the text fails to convey the pattern to the complex-event knowledge base. In contrast, the Mikrokosmos semantic analyzer has demonstrated 97% accuracy in assigning meaning representations to open class words (e.g., verbs, nouns, and adjectives) encountered in real-world texts (see Viegas *et al*. 1999).

Previous systems have also been criticized for the restricted domains for which they have developed. Though Ms. Malaprop, with only one complex event (for painting a chair), is the most obvious target, it is also true that none represents more than 60 complex events FRUMP (DeJong 1979)—and even these are “sketchy scripts” with only enough detail to skim news articles for a handful of pieces of information. With only a limited number of complex events and with no other meaning-representation module, scripts are likely to “overextend” and apply in places they should not. Furthermore, since previous systems developed complex events in relative isolation, there is no guarantee that the knowledge represented therein will not interfere and overlap with subsequent complex events. The representation argued for in this dissertation meets these criticisms by:

- having the support of other input analysis modules (i.e., semantic and syntactic analyzers);
- using an ontology to ground and guide acquisition to ensure broad, even coverage; and,
- developing complex events in semantically closely related clusters to ensure that knowledge is sensitive to fine-grained conceptual differences.

3.7.2 Summary

This focus of this chapter has been on: (i) considering the complexity of real-world events that are complex; (ii) developing a representation formalism for complex events that is able to represent this real-world complexity and that meets the criticisms levied against previous systems complex events; and, (iii) providing the necessary background information to understand the representation of complex events presented in Chapter Four.

It has been argued that the ontological hierarchy provides the necessary apparatus for which to capture information about complex events. The PRECONDITIONS, HAS-PARTS, and EFFECTS slots, for instance, aid in the representation of contingencies common in real-world events that are complex. Furthermore, the Optional slot allows representation of events that are perhaps not assumed, but are possible or likely. This allows complex-event knowledge to have wider scope, to capture more than simply stereotypical information.

Slight changes to the ontological formalism, of course, have had to been made. In addition to the Optional facet noted above, this chapter has also introduced the BINDING-ROLE relation to deal with the complexities of variable-binding, due to the fact that events can "see" down to their sub-subevents (and even further). Finally, along the way, principles and heuristics have been presented to guide implementation of the theory set forth in this chapter—since even a sound theory is of little use unless there are clear principles to guide its implementation.

Now that the representation of complex-event knowledge within the Mikrokosmos system has been made clear, Chapter Four illustrates several complex events that have been thus far developed and outlines their sufficiency to handle inferencing for real-world texts.

CHAPTER FOUR:

COMPLEX-EVENT DATA AND ANALYSIS

4.1 Introduction

Drawing on the ontological-semantic system described in Chapter One and on specific recommendations for representing complex-event knowledge based on that system described in Chapter Three, this chapter adduces evidence to support the formalism. In sections 4.2 through 4.5, several example complex events are presented in order to demonstrate:

(i) the adequacy of the formalism to represent real-world events that are complex;
(ii) the feasibility of implementation of the formalism; and,
(iii) the extendability of the formalism to handle, in principle, the entirety of complex-event knowledge.

In section 4.2, a complex-event representation is specified for a common and richly specified event, the concept "to buy," as in "John bought a newspaper." Then, subevents of this complex event are themselves developed as complex events, to establish the ability of the complex-event knowledge base to represent interconnected knowledge (section 4.3). Furthermore, EVENT concepts clustered around BUY are developed as complex, to establish the sensitivity of the knowledge base to subtle differences in the input (section 4.4). In section 4.5, one very complex event is developed, to establish the extreme depth to which knowledge in this formalism can be specified. Section 4.6 reviews the linking of EVENT frames possible in this formalism. Finally, section 4.7 provides a conclusion for this chapter.

4.2 The complex event for BUY

First, a few words about what an event of buying entails.[12] It has many synonyms, e.g., purchase, procure, obtain, get, secure. Each of these lexical items is, in the Mikrokosmos lexicon, mapped to the BUY concept. A buy-event entails a buyer (the AGENT of the buy-event), a seller (the ACCOMPANIER), and an entity that is bought, the merchandise (or, the THEME). These first two must be of the class HUMAN, while the third can be nearly any entity other than a HUMAN (though in rare cases even this is possible. The THEME is, in the course of a buy-event, transferred (in some sense) from the seller to the buyer. There is an implied transfer, from the buyer to the seller, of something categorizable as money (the instrument of the exchange). That is, in American culture at least, a buy-event clearly can only be said to have occurred if money is involved (cf. (1); an asterisk denotes an ill-formed sentence):

(1)
(a) John bought the book from Mary with a twenty-dollar-bill.
(b) *John bought the book from Mary with a tropical fish.

Although John and Mary may engage in an exchange of a book for a fish, the exchange is expressly prohibited from being called an event of buying. (Instead, this would be

[12] We do not address its non-literal meanings, e.g., *John* bought *the theory*, meaning that he accepted the truth of it; *John* bought *some time*, meaning that he delayed a deadline; *John* bought *the farm*, as a metaphor for John dying; or, *A dollar* buys *a cup of coffee.*

classified as an EXCHANGE event in the ontology.) All of this information is expressed in the ontological concept, BUY:

Frame: BUY

Slot	Facet	Filler
DEFINITION	Value	"A transfer of possession event (buying or selling) that involves the exchange of money for merchandise"
IS-A	Value	EVERYDAY-FINANCIAL-EVENT, TRANSFER-POSSESSION
SUBCLASSES	Value	AUCTION, BUYOUT
ACCOMPANIER	Sem	HUMAN
AGENT	Sem	HUMAN
BENEFICIARY	Sem	HUMAN
COST	Sem	(>0)
	Meas-Unit	MONETARY-UNIT
DESTINATION	Sem	HUMAN
HAS-PARTS	Default	TRANSFER-POSSESSION
	Sem	EVENT
LOCATION	Sem	PLACE
SOURCE	Sem	HUMAN
THEME	Sem	OBJECT (Not HUMAN)

However, there is much more information that can be specified for a buy-event. For example, there are many other options besides the seller physically giving the buyer the merchandise: for example, the merchandise may be too heavy to carry (it may be a car that the buyer drives away), the buyer and seller may be in different locations (in which case the merchandise must be sent by mail), the merchandise may be a service (in which case buying the service entails some other event occurring). Moreover, it is sometimes the case that the human agent acts only as a proxy for an organization. And, although there is generally an implied transfer-of-ownership from the seller to the buyer, when the merchandise is an object, this is certainly not the case when the merchandise is a service.

Furthermore, other events may also happen during the course of a buy-event: a buyer may receive change if the amount of money tendered is greater than the cost of the merchandise, the buyer and seller may haggle over the cost, the buyer and seller may engage in a contractual agreement concerning the terms of the sale, the buyer may not want to buy the merchandise but might do so anyway, the buyer may be required to provide personal information before the buy-event can be completed, etc. All of this information can be specified in the complex-event knowledge formalism.

In developing BUY as a complex event, this leaf-node concept was given two children: BUY-OBJECT and BUY-SERVICE. The motivation for this is that the two events have starkly different *criterial* (Sem) events. For example, buying an object logically entails gaining ownership of that object, while buying a service logically entails just the opposite. Subsuming such branching under BUY would require that the TRANSFER-OF-POSSESSION event be listed (in the EFFECTS slot) with the Optional facet.[13] This means that a transfer-of-possession would not be inferred by the system unless it is presented with explicit evidence. On the other hand, by splitting BUY into BUY-SERVICE (i.e., THEME = EVENT) and BUY-OBJECT (i.e., THEME = OBJECT), we allow this inference to be made necessarily if the event is classified as a BUY-OBJECT event. Furthermore, being able to classify the merchandise as either EVENT or OBJECT increases the inferability of such possible subevents as: the buyer searching for the merchandise, holding it, or receiving it later in the mail (all optional for BUY-OBJECT, but impossible for BUY-

[13] One way around this is to specify the TRANSFER-OWNERSHIP event with a precondition. That precondition would then take the BUY event as its BINDING-ROLE. Finally, a second BINDING-ROLE—for Buy.Theme—could then be filled with OBJECT, yielding the correct inference. Besides the fact that structuring complex events this way would be torturous for acquirers, it is not clear whether searching mechanisms in the Mikrokosmos system would support self-referentiality, or "if the event of which the subevent is a part has property X, then…"

SERVICE), or the seller performing some action(s) entailed by the buy-event (optional for BUY-SERVICE, but unlikely for BUY-OBJECT). Complex-event representations are shown, respectively, for both BUY-OBJECT and BUY-SERVICE. To demonstrate the feasibility of the formalism, BUY-OBJECT is shown here with subevents collected together first, and then their binding-roles, collected at the end of the frame, as a complex event might look in the Mikrokosmos Knowledge Base Acquisition Editor. However, subsequent complex-event schematics will note binding-roles together with the subevents of which they are a part. (Note that clarifications for the reader are bracketed and that slots and fillers inherited from the parent concept are shown beneath the child-concept frame, separated by a line.)

Frame: BUY-OBJECT

Slot	Facet	Filler
DEFINITION	Value	"A transfer of possession event (buying or selling) that involves the exchange of money for an object"
IS-A	Value	BUY
SUBCLASSES	Value	AUCTION, BUYOUT
THEME	Sem	OBJECT
PRECONDITIONS	Sem	DESIRE-1
	Sem	OWN-1
	Sem	OWN-2
	Optional	LEND-1
	Optional	WITHDRAW-ATM
	Optional	DIAL
HAS-PARTS	Optional	LOCATE
	Optional	CHANGE-LOCATION
	Optional	NEGOTIATE-TRANSACTION
	Default	GIVE [object]
	Sem	PAY
EFFECTS	Sem	TRANSFER-POSSESSION
	Sem	DECREASE-1

Frame: BUY		
ACCOMPANIER	Sem	HUMAN
AGENT	Sem	HUMAN
BENEFICIARY	Sem	HUMAN
COST	Sem	(>0)
	Meas-Unit	MONETARY-UNIT
DESTINATION	Sem	HUMAN
LOCATION	Sem	PLACE
SOURCE	Sem	HUMAN
INSTRUMENT	Sem	MONEY

The above information lists the semantic case roles associated with BUY-OBJECT, as well as its (first-level) subevents (preconditions, components, and effects), followed by the slots and fillers it inherits from BUY. Thus, the default assumption is that the buyer wants to buy the object (DESIRE), although some purchases, like auto insurance, may be compulsory. Furthermore, the seller must own the object being sold to the buyer (OWN-1). Prior to the buy event, the buyer may have had to secure a loan (LEND) or have had to withdraw money from an ATM machine (WITHDRAW-ATM), from a bank teller, or may even have had to retrieve a credit card from a desk drawer; regardless, it may be useful to represent events related to the securing of money to be used for the purchase. Furthermore, a buyer may make a purchase over the phone (or on the Internet), in which case a series of subevents will ensue (DIAL). If, however, the buyer, seller, and merchandise are all in the same place (in a store, for example), then we might expect the buyer to locate the merchandise (LOCATE) and then bring it to the seller, at a cash register (CHANGE-LOCATION). A slightly different situation—such as buying a car—may then call for haggling over price of the merchandise (NEGOTIATE-TRANSACTION). Assuming the merchandise is small enough to carry, we might expect the buyer to give it to the seller (GIVE), thus expressing an intention to buy. The buyer then pays for the merchandise (PAY), effecting a transfer of possession (TRANSFER-POSSESSION) and entailing that the buyer has less money than before (DECREASE) and that the seller has more. What follows is the variable bindings for the associated subevents. In the event that a given subevent specifies its own subevent(s)—i.e., a sub-subevent of the complex event—variable bindings for these events follow those for the subevents. When a subevent has a BINDING-ROLE for the complex event that is different from its filler in its EVENT concept frame, the BINDING-ROLE is always listed first. A subevent's slots not bound to complex-event variables are not shown. Finally, a BINDING-ROLE slot always takes a Sem facet.

Frame: BUY-OBJECT [continued]

Slot	Facet	Filler
BR.DESIRE.AGENT	Sem	Buy-Object.Agent, HUMAN
BR.DESIRE.THEME	Sem	Buy-Object.Theme, OBJECT
BR.OWN-1.AGENT	Sem	Buy-Object.Accompanier, HUMAN
BR.OWN-1.THEME	Sem	Buy-Object.Theme, OBJECT
BR.OWN-2.AGENT	Sem	Buy-Object.Agent, HUMAN
BR.OWN-2.THEME	Sem	(>0)
	Meas-Unit	MONETARY-UNIT
BR.LEND-1.AGENT	Sem	HUMAN
BR.LEND-1.BENEFICIARY	Sem	Buy-Object.Agent, HUMAN
BR.LEND-1.THEME	Sem	(>0)
	Meas-Unit	MONETARY-UNIT
BR.WITHDRAW-ATM.AGENT	Sem	Buy-Object.Agent, HUMAN
BR.WITHDRAW-ATM.THEME	Sem	(>0)
	Meas-Unit	MONETARY-UNIT
BR.DIAL.AGENT	Sem	Buy-Object.Agent, HUMAN
BR.DIAL.INSTRUMENT	Sem	PHONE
BR.DIAL.THEME	Sem	Buy-Object.Accompanier.Phone-Number, PHONE-NUMBER
BR.DIAL.EFFECTS	Sem	INFORM, EVENT
BR.LOCATE.AGENT	Sem	Buy-Object.Agent, HUMAN
BR.LOCATE.THEME	Sem	Buy-Object.Theme, OBJECT
BR.LOCATE.EFFECTS	Sem	Buy-Object.Has-Parts. Change-Location, EVENT
BR.CHANGE-LOCATION.AGENT	Sem	BUY-OBJECT.AGENT, HUMAN
BR.CHANGE-LOCATION.PLACE-OF	Sem	Buy-Object.Accompanier, OBJECT

BR.CHANGE-LOCATION.PRECOND	Sem	Buy-Object.Has-Parts.Locate
BR.NEGOTIATE-TRANS.AGENT	Sem	Buy-Object.Agent, HUMAN
BR.NEGOTIATE-TRANS.ACCMPNR	Sem	Buy-Object.Accompanier, HUMAN
BR.NEGOTIATE-TRANS.EFFECTS	Optional	DECREASE-2, EVENT
BR.GIVE-1.AGENT	Sem	Buy-Object.Agent, HUMAN
BR.GIVE-1.BENEFICIARY	Sem	Buy-Object.Accompanier, HUMAN
BR.GIVE-1.THEME	Sem	Buy-Object.Theme, OBJECT
BR.GIVE-1.EFFECTS	Sem	PAY, EVENT
BR.PAY.AGENT	Sem	Buy-Object.Agent, HUMAN
BR.PAY.BENEFICIARY	Sem	Buy-Object.Accompanier, HUMAN
	Relax-to	CORPORATION
BR.TRANSFER-POSSESSION.AGENT	Sem	Buy-Object.Accompanier, HUMAN
BR.TRANSFER-POSSESSION.BENEF	Sem	Buy-Object.Agent, HUMAN
BR.TRANSFER-POSSESSION.THEME	Sem	Buy-Object.Theme, OBJECT
BR.DECREASE-1.INITIAL-VALUE	Sem	Own-2.Theme
	Meas-Unit	MONETARY-UNIT
BR.DECREASE-1.FINAL-VALUE	Sem	(Own-2.Theme - Buy-Object.Cost)
	Meas-Unit	MONETARY-UNIT

Next, variable-bindings for the PRECONDITIONS, HAS-PARTS, and EFFECTS slots of subevents (if any) are specified as sub-subevents. Generally, however, most subevents will be defined in terms of other subevents in the complex event, and require no further binding-roles.

BR.INFORM.AGENT	Sem	Buy-Object.Agent
BR.INFORM.THEME	Sem	Buy-Object.Theme
BR.INFORM.PRECONDITIONS	Sem	Buy-Object.Preconditions.Desire,
BR.INFORM.HAS-PARTS	Sem	SPEECH-ACT
BR.INFORM.EFFECTS	Default	PAY, EVENT
BR.GIVE-2.AGENT	Sem	Buy-Object.Accompanier, HUMAN
BR.GIVE-2.BENEFICIARY	Sem	Buy-Object.Agent, HUMAN
BR.GIVE-2.THEME	Sem	Buy-Object.Theme, OBJECT
BR.DECREASE-2.INITIAL-VALUE	Sem	Own-2.Theme
BR.DECREASE-2.FINAL-VALUE	Sem	(Own-2.Theme – Buy-Object.Cost)

Here, the sub-subevents SPEECH-ACT (the act of saying something) and PAY have not been developed with binding-roles. However, in section 4.3.2, PAY will be developed as a complex event in its own right. Next, for comparison, the complex event representation for BUY-SERVICE is shown.

Frame: BUY-SERVICE

Slot	Facet	Filler
DEFINITION	Value	"to use money to obtain a service"
IS-A	Value	BUY
SUBCLASSES	Value	[None]
THEME	Sem	EVENT
PRECONDITIONS	Default	DESIRE
	Optional	WITHDRAW-ATM
	Sem	OWN-1
	Optional	DIAL
HAS-PARTS	Optional	NEGOTIATE-TRANSACTION
	Sem	PAY
EFFECTS	Sem	DECREASE-1
	Sem	SERVICE-EVENT

Frame: BUY		
ACCOMPANIER	Sem	HUMAN
AGENT	Sem	HUMAN
BENEFICIARY	Sem	HUMAN
COST	Sem	(>0)
	Meas-Unit	MONETARY-UNIT
DESTINATION	Sem	HUMAN
LOCATION	Sem	PLACE
SOURCE	Sem	HUMAN
INSTRUMENT	Sem	MONEY

Next, binding-roles for the subevents of BUY-SERVICE are shown.

BR.DESIRE.AGENT	Sem	Buy-Service.Agent, HUMAN
BR.DESIRE.THEME	Sem	Buy-Service.Theme, EVENT, OBJECT
BR.WITHDRAW-ATM.AGENT	Sem	Buy-Service.Agent, HUMAN
BR.WITHDRAW-ATM.THEME	Sem	MONEY
BR.OWN-1.AGENT	SEM	Buy-Service.Agent, HUMAN
BR.OWN-1.THEME	Sem	(>0)
	Meas-Unit	MONETARY-UNIT
BR.DIAL.AGENT	Sem	Buy-Service.Agent, HUMAN
BR.DIAL.INSTRUMENT	Sem	PHONE
BR.DIAL.THEME	Sem	Buy-Service.Accompanier.Phone-

		Number, PHONE-NUMBER
BR.DIAL.EFFECTS	Sem	INFORM, EVENT
BR.NEGOTIATE-TRANS.AGENT	Sem	Buy-Service.Agent, HUMAN
BR.NEGOTIATE-TRANS.ACCMPNR	Sem	Buy-Service.Accompanier, HUMAN
BR.NEGOTIATE-TRANS.EFFECTS	Optional	DECREASE-2, EVENT
BR.PAY.AGENT	Sem	Buy-Service.Agent, HUMAN
BR.PAY.BENEFICIARY	Sem	Buy-Service.Accompanier, HUMAN
	Relax-to	CORPORATION
BR.DECREASE-1.INITIAL-VALUE	Sem	Own-1.Theme
BR.DECREASE-1.FINAL-VALUE	Sem	(Own-1.Theme - Buy-Service.Cost)
BR.SERVICE-EVENT.AGENT	Sem	Buy-Service.Accompanier, HUMAN
BR.SERVICE-EVENT.BENEFICIARY	Sem	Buy-Service.Agent, HUMAN

Note that the THEME of the SERVICE-EVENT, specified in the EFFECTS slot of BUY-SERVICE, cannot be determined and so is not provided with a BINDING-ROLE. Finally, BINDING-ROLES for the sub-subevents of BUY-SERVICE are provided.

BR.INFORM.AGENT	Sem	Buy-Service.Agent, HUMAN
BR.INFORM.THEME	Sem	Buy-Service.Theme, EVENT
BR.INFORM.PRECONDITIONS	Sem	Buy-Service.Preconditions.Desire,
BR.INFORM.HAS-PARTS	Sem	SPEECH-ACT
BR.INFORM.EFFECTS	Default	PAY, EVENT
BR.DECREASE-1.AGENT	Sem	Buy-Service.Agent, Buy-Service. Accompanier, HUMAN
BR.DECREASE-1.THEME	Sem	Buy-Service.Cost

Again, the SPEECH-ACT and PAY sub-subevents are not decomposed here. Thus, the complex-event knowledge specified for BUY-SERVICE and BUY-OBJECT highlights the important difference between purchasing a physical-object and purchasing a service. Furthermore, specifying subevents of each complex event provides knowledge that can be used to further aid inferencing and the building of TMRs. The following section provides complex-event knowledge for three component events of BUY, as further evidence of the formalism's ability to represent knowledge of greater depth.

4.3 Subevents of BUY-OBJECT and BUY-SERVICE developed as complex events

In this section, three subevents of BUY-OBJECT and BUY-SERVICE are developed as complex events themselves: WITHDRAW-ATM, PAY, and NEGOTIATE-TRANSACTION.

4.3.1 The WITHDRAW-ATM complex event

First, new concepts are motivated. The current ontology contains a BANKING-EVENT concept, under which are the following (two-levels deep):

(3) BANKING-EVENT
- CANCEL-OBLIGATION
- CLOSE-ACCOUNT
- COMPOUND
- DEMAND-PAYMENT
- DEPOSIT
- ISSUE-OBLIGATION
- OPEN-ACCOUNT
- OWE
- USURY
- WITHDRAW

There are at least three ways one (in the United States) can go about withdrawing money from an account: speak with a bank teller, use an ATM machine, and (increasingly) get "cash back" after a debit-card purchase. This third way does not seem common enough to motivate an ontological concept of its own and might even be handled in the EFFECTS slot of a BUY-OBJECT event, as a special case of RECEIVE [THEME: MONEY], i.e., "receive change." However, WITHDRAW-TELLER and WITHDRAW-ATM[14] do appear to be

[14] Both of these concept names might be unfavorable, on final analysis, for different reasons. In an ontology with nearly 8,000 concepts, concept naming conventions are especially important to prevent confusion. Generally, naming conventions specify that when a concept name includes an argument, that argument plays the THEME role in that event (see Mahesh 1996). Thus, WITHDRAW-TELLER is possibly misconstrued as "to withdraw a teller." Second, the term "ATM," although becoming more widespread, is

sufficiently different enough to be made child concepts of WITHDRAW and sufficiently specifiable to be developed as complex events. Only WITHDRAW-ATM is developed here.

A WITHDRAW-ATM event is remarkably well-defined. Although it has a rich subevent structure, the preconditions, component events, and effects of the subevents yield a limited set of possible continuations. That is, though WITHDRAW-ATM has, like any complex event, a number of Optional subevents, these Optional subevents have specifiable preconditions, unlike, for example, a BUY-OBJECT event, in which the Optional subevent NEGOTIATE-TRANSACTION may arise for many reasons and in many situations. Of course, any complex event, including WITHDRAW-ATM, may have deviations from the stereotypical complex event structure that must be handled by the other meaning-building modules (e.g., the semantic and syntactic analyzers).

Binding-roles will be shown together with the subevent and sub-subevents of which they are a part, for ease of comparison.

Frame: WITHDRAW-ATM

Slot	Facet	Filler
DEFINITION	Value	"to withdraw money from a machine (using a type of credit card)"
IS-A	Value	WITHDRAW
SUBCLASSES	Value	[None]
INSTRUMENT	Sem	ATM-CARD[15]
LOCATION	Sem	ATM-MACHINE, PLACE
SOURCE	Sem	BANK
THEME	Sem	(>0)

still perhaps a regionalism. Again, confusion may arise. Although this may seem like nit-picking, it is to be remembered that the ontology is meant to be language-independent. Still, the concept name WITHDRAW-ATM will be adopted in this dissertation, with the explanation that an ATM is "an automated teller machine used for withdrawing money from an account."

[15] The OBJECT concepts ATM-CARD and ATM-MACHINE are putative new concepts.

	Meas-Unit	MONETARY-UNIT
PRECONDITIONS	Sem	DESIRE
	Sem	OWN-1 [=have ATM card]
	Default	OWN-2 [=have available credit]
	Sem	CHANGE-LOCATION
HAS-PARTS	Sem	INSERT
	Sem	INFORM-1
	Optional	AGREE
	Sem	SELECT-1 [i.e., withdrawal or deposit]
	Sem	SELECT-2 [i.e., checking or savings]
	Sem	SELECT-3 [i.e., amount]
	Sem	RECEIVE-1 [theme: money]
	Sem	RECEIVE-2 [theme: card, receipt]
EFFECTS	Sem	DECREASE
	Sem	TRANSFER-POSSESSION
	Sem	OWN-3 [i.e., agent possesses money]

Frame: WITHDRAW		
AGENT	Sem	HUMAN
DIRECTION-OF-CHANGE	Sem	NEGATIVE
TYPE-OF-CHANGE	Sem	SOCIAL

Following are the binding-roles for the subevents and sub-subevents of WITHDRAW-ATM.

<u>Withdraw-ATM.Preconditions</u>

DESIRE

AGENT	Sem	Withdraw-ATM.Agent, HUMAN
THEME	Sem	MONEY, EVENT, OBJECT

<u>Withdraw-ATM.Preconditions</u>

OWN-1

AGENT	Sem	Withdraw-ATM.Agent, HUMAN
THEME	Sem	Withdraw-ATM.Instrument, OBJECT

<u>Withdraw-ATM.Preconditions</u>

OWN-2

AGENT	Sem	Withdraw-ATM.Agent, HUMAN

THEME	Sem	CREDIT, OBJECT

Withdraw-ATM.Preconditions

CHANGE-LOCATION

AGENT	Sem	Withdraw-ATM.Agent, ANIMAL
DESTINATION	Sem	Withdraw-ATM.Location, PLACE

Withdraw-ATM.Has-Parts

INSERT

AGENT	Sem	Withdraw-ATM.Agent, HUMAN
THEME	Sem	Withdraw-ATM.Instrument, OBJECT
DESTINATION	Sem	Withdraw-ATM.Location, PLACE
EFFECTS	Sem	INFORM-1, EVENT

Withdraw-ATM.Has-Parts

INFORM-1

AGENT	Sem	Withdraw-ATM.Agent, HUMAN
BENEFICIARY	Sem	Withdraw-ATM.Source
THEME	Domain	INFORMATION
	Range	PIN-NUMBER
INSTRUMENT	Sem	KEYPAD[16]
EFFECTS	Default	ACCEPT
	Optional	REJECT

Inform-1.Effects

ACCEPT-1

AGENT	Sem	BANK, HUMAN
THEME	Sem	Inform-1.Theme
EFFECTS	Sem	SELECT-1, EVENT

Inform-1.Effects

REJECT

AGENT	Sem	BANK, HUMAN
THEME	Domain	Inform-1.Theme, EVENT
EFFECTS	Default	INFORM-1, EVENT

Withdraw-ATM.Has-Parts

AGREE-1

AGENT	Sem	Withdraw-ATM.Agent, HUMAN
EFFECTS	Default	PREMIUM, EVENT
	Sem	SELECT-1

[16] KEYPAD is a putative new OBJECT concept.

Agree.Effects

PREMIUM

AGENT	Sem	Withdraw-ATM.Agent, HUMAN
BENEFICIARY	Sem	Withdraw-ATM.Source, OBJECT
THEME	Sem	(>0)
	Meas-Unit	MONETARY-UNIT

Withdraw-ATM.Has-Parts

SELECT-1

AGENT	Sem	Withdraw-ATM.Agent, HUMAN
THEME	Sem	INFORMATION
INSTRUMENT	Sem	KEYPAD
PRECONDITIONS	Sem	ACCEPT-1, EVENT
	Optional	AGREE-1, EVENT
EFFECTS	Default	WITHDRAW, EVENT
	Optional	DEPOSIT, EVENT

Select-1.Effects

WITHDRAW

AGENT	Sem	Withdraw-ATM.Agent, HUMAN
EFFECTS	Sem	SELECT-2, EVENT

Withdraw-ATM.Has-Parts

SELECT-2

AGENT	Sem	Withdraw-ATM.Agent, HUMAN
THEME	Sem	CHECKING-ACCOUNT, SAVINGS-ACCOUNT, OBJECT
INSTRUMENT	Sem	KEYPAD
PRECONDITIONS	Sem	SELECT-1
EFFECTS	Sem	SELECT-3

Withdraw-ATM.Has-Parts

SELECT-3

AGENT	Sem	Withdraw-ATM.Agent, HUMAN
THEME	Sem	(>0)
	Meas-Unit	MONETARY-UNIT
INSTRUMENT	Sem	KEYPAD, OBJECT
SOURCE	Sem	CHECKING-ACCOUNT, SAVINGS-ACCOUNT, OBJECT
PRECONDITIONS	Sem	SELECT-2
EFFECTS	Default	RECEIVE-1, EVENT

Withdraw-ATM.Has-Parts

RECEIVE-1

AGENT	Sem	Withdraw-ATM.Agent, HUMAN
THEME	Sem	Select-3.Theme, OBJECT
SOURCE	Sem	Withdraw-ATM.Source, OBJECT
PRECONDITIONS	Sem	OWN-2, EVENT
	Sem	SELECT-3, EVENT
EFFECTS	Default	RECEIVE-2, EVENT

Withdraw-ATM.Has-Parts

RECEIVE-2

AGENT	Sem	Withdraw-ATM.Agent, HUMAN
THEME	Sem	ATM-CARD, RECEIPT, OBJECT
SOURCE	Sem	Withdraw-ATM.Source, OBJECT

Withdraw-ATM.Effects

DECREASE

AGENT	Sem	Withdraw-ATM.Agent, HUMAN
ACCOMPANIER	Sem	Withdraw-ATM.Source, HUMAN
DIRECTION-OF-CHNG	Sem	NEGATIVE
FINAL-VALUE	Sem	(Own-2.Theme – Select-3.Theme) [i.e., original credit amount minus amount withdrawn]
INITIAL-VALUE	Sem	Own-2.Theme
INSTRUMENT	Sem	Withdraw-ATM.Instrument, OBJECT
THEME	Sem	CHECKING-ACCOUNT, SAVINGS-ACCOUNT, OBJECT
LOCATION	Sem	Withdraw-ATM.Location
TYPE-OF-CHANGE	Sem	SOCIAL

Withdraw-ATM.Effects

TRANSFER-OF-POSSESSION

AGENT	Sem	Withdraw-ATM.Source, HUMAN
BENEFICIARY	Sem	Withdraw-ATM.Agent, HUMAN
SOURCE	Sem	ATM-MACHINE
THEME	Sem	Receive-2.Theme, OBJECT

Withdraw-ATM.Effects

OWN-3

AGENT	Sem	Withdraw-ATM.Agent, HUMAN
THEME	Sem	Select-3.Theme, Receive-2.Theme, OBJECT

The complex event WITHDRAW-ATM can then, of course, be instantiated in the course of a BUY-SERVICE or BUY-OBJECT event. Before proceeding to PAY, note that if a WITHDRAW-ATM event is asserted in a text, the system may then set up expectations that a BUY event will occur based on the following: the WITHDRAW-ATM event takes, in its THEME role, MONEY. In turn, MONEY is specified for the INSTRUMENT-OF slot, taking the filler BUY.

4.3.2 The PAY complex event

An event of buying necessarily entails an event of paying. Conversely, paying entails that one is engaged in an event of buying. The PAY concept, in the ontology is limited to applying in cases where (i) the instrument of the event is a MONETARY-UNIT, and (ii) there is a mutual exchange. It is interesting (but inconsequential computationally) that the English word "pay" also generally exhibits these conceptual classifications, thus (4):

(4)

(i)	(a)	Mary paid for the book at the garage sale. (=used money)
	(b)	Mary paid for the book with a twenty-dollar bill at the garage sale.
	(c)	*Mary paid for the book at the garage sale with a stamp collection.
(ii)	(a)	Mary paid. (=got something in return)
	(b)	Mary paid her insurance company (=was provided insurance)
	(c)	*Mary paid her favorite charity.

In (4i.c), a verb such as "traded" or "exchanged" is required, while in (4ii.c) "donated" or "gave" is required.

Frame: PAY

Slot	Facet	Filler
DEFINITION	Value	"to give to (a person) what is due, as

		for goods or services"
IS-A	Value	EVERYDAY-FINANCIAL-EVENT
SUBCLASSES	Value	SUBSCRIBE-TO
AGENT	Sem	HUMAN
	Relax-to	CORPORATION
BENEFICIARY	Sem	HUMAN
	Relax-to	CORPORATION
INSTRUMENT	Sem	(>0)
	Meas-Unit	MONETARY-UNIT
LOCATION	Sem	PLACE
THEME	Sem	OBJECT, SERVICE-EVENT
COST	Sem	(>0)
	Meas-Unit	MONETARY-UNIT
PRECONDITIONS	Default	BUY
	Default	OWN-1 [have money]
	Default	DESIRE
HAS-PARTS	Default	RETRIEVE-1 [money from container]
	Sem	GIVE-1 [money to payee]
	Optional	RECEIVE-1 [change]
	Default	RECEIVE-2 [receipt]
EFFECTS	Optional	OWN-2
	Optional	SERVICE-EVENT

The complex event representation continues with a specification of variable bindings.

<u>Pay.Preconditions</u>

BUY

AGENT	Sem	Pay.Agent, HUMAN
THEME	Sem	Pay.Theme, OBJECT, SERVICE-EVENT
EFFECTS	Optional	RETRIEVE-1, EVENT
	Default	GIVE-1, EVENT

<u>Pay.Preconditions</u>

OWN-1

AGENT	Sem	Pay.Agent, HUMAN
THEME	Sem	(>0)
	Meas-Unit	MONETARY-UNIT

<u>Pay.Preconditions</u>

DESIRE

AGENT	Sem	Pay.Agent, HUMAN
THEME	Sem	Pay.Theme, OBJECT, SERVICE-

			EVENT

Pay.Has-Parts

RETRIEVE-1			
	AGENT	Sem	Pay.Agent, HUMAN
	THEME	Sem	Own-1.Theme, MONEY
	SOURCE	Sem	CONTAINER, OBJECT
		Default	PURSE, WALLET[17]
	EFFECTS	Default	GIVE-1, EVENT

Pay.Has-Parts

GIVE-1			
	AGENT	Sem	Pay.Agent, HUMAN
	THEME	Sem	Retrieve-1.Theme, OBJECT
	BENEFICIARY	Sem	Pay.Beneficiary, HUMAN
	PRECONDITIONS	Default	RETRIEVE-1, EVENT
	EFFECTS	Optional	OWN-2, EVENT
		Optional	SERVICE-EVENT

Pay.Has-Parts

RECEIVE-1			
	AGENT	Sem	Pay.Agent, ANIMAL
	THEME	Domain	MONETARY-UNIT
		Range	(Give-1.Theme - Pay.Cost)
	PRECONDITIONS	Sem	GREATER-THAN
		Sem	GIVE-1, EVENT

Receive-1.Preconditions

GREATER-THAN			
	Domain	Sem	MONETARY-UNIT
	Range	Sem	(Give-1.Theme > Pay.Cost)

Pay.Has-Parts

RECEIVE-2			
	AGENT	Sem	Pay.Agent, HUMAN
	THEME	Sem	RECEIPT
	SOURCE	Sem	Pay.Beneficiary, HUMAN
		Relax-to	CORPORATION
	PRECONDITIONS	Sem	GIVE-1, EVENT

[17] WALLET is a putative new OBJECT concept

Pay.Effects

OWN-2			
	AGENT	Sem	Pay.Agent, HUMAN
	THEME	Sem	Pay.Theme, OBJECT

Pay.Effects

SERVICE-EVENT			
	AGENT	Sem	Pay.Beneficiary, HUMAN
		Relax-to	CORPORATION
	BENEFICIARY	Sem	Pay.Agent, Human
	THEME	Sem	SERVICE-EVENT

The PAY complex event could, if needed, be nested within the BUY-SERVICE or BUY-OBJECT complex events, without much difficulty. The only binding-roles affected would be those referring to the PAY event's slot-fillers. Because, within BUY-SERVICE and BUY-OBJECT, the PAY event's slot-fillers are bound to those of BUY-SERVICE and BUY-OBJECT, a simple binding-chain is created, which may be easily resolved: for example, Retrieve-2.Agent = Pay.Agent and Pay.Agent = Buy-Service.Agent, yielding (correctly) Retrieve-2.Agent = Buy-Service.Agent. That is, the person retrieving money with which to pay is also the person purchasing the merchandise. Binding-chain-resolution might be automated to further simplify this process.

4.3.3 The NEGOTIATE-TRANSACTION complex event

First, consider what events might be within the scope of NEGOTIATE-TRANSACTION and who might be involved in such an event. NEGOTIATE-TRANSACTION is descended from CORPORATE-EVENT in the ontology and is defined therein as "to work out the terms of a transaction in order to reach an agreement." Although, intuitively, many other entities might engage in such action, the complex event developed here will assume that both parties involved are corporations or their proxies. It will be further assumed that in a NEGOTIATE-TRANSACTION event, each of the parties involved stands to receive something of some sort; otherwise, the event seems more aptly categorized as a DEMAND or COMMAND event. Thus, it will be assumed that a NEGOTIATE-TRANSACTION event has the following properties:

- X wants something, A;
- Y is in a position to provide A for X;
- X offers to provide something, B, for Y in exchange for Y providing A for X;
- Y either accepts the terms of this exchange, proposes new terms, or rejects them without proposing new terms;
- X, in the event that Y has not accepted the previous terms, may then accept Y's terms (if any) or may propose new terms; or X may not propose new terms.

Then, of course, the negotiation may loop until either X and Y reach an agreement or else there are no new proposals to consider. The following is the complex event for NEGOTIATE-TRANSACTION (which will be abbreviated NEG-TRANS).

Frame: NEGOTIATE-TRANSACTION

Slot	Facet	Filler
DEFINITION	Value	"to work out the terms of a transaction in order to reach an agreement"
IS-A	Value	CORPORATE-EVENT
ACCOMPANIER	Sem	HUMAN
	Relax-to	CORPORATION
AGENT	Sem	HUMAN
LOCATION	Sem	PLACE
THEME	Sem	EVENT, OBJECT
PRECONDITIONS	Sem	DESIRE
	Sem	OWN
HAS-PARTS	Sem	PROPOSE-1
	Sem	VOTE-1
	Optional	PROPOSE-2
	Optional	VOTE-2

EFFECTS	Sem	EXCHANGE-1

<u>Neg-Trans.Preconditions</u>

DESIRE

AGENT	Sem	Neg-Trans.Agent, HUMAN
THEME	Sem	Neg-Trans.Theme, OBJECT, EVENT

<u>Neg-Trans.Preconditions</u>

OWN

AGENT	Sem	Neg-Trans.Accompanier, HUMAN
THEME	Sem	Neg-Trans.Theme, OBJECT, EVENT

<u>Neg-Trans.Has-Parts</u>

PROPOSE-1

AGENT	Sem	Neg-Trans.Agent, HUMAN
THEME	Sem	EXCHANGE, EVENT
Beneficiary	Sem	Neg-Trans.Accompanier, HUMAN

<u>Propose-1.Theme</u>

EXCHANGE-2

AGENT	Sem	Neg-Trans.Agent, HUMAN
INSTRUMENT	Sem	EVENT, OBJECT
THEME	Sem	Neg-Trans.Theme, EVENT, OBJECT
BENEFICIARY	Sem	Neg-Trans.Accompanier, HUMAN

<u>Neg-Trans.Has-Parts</u>

VOTE-1

AGENT	Sem	Neg-Trans.Accompanier, HUMAN
THEME	Sem	EXCHANGE-2, EVENT
PRECONDITIONS	Sem	PROPOSE-1
EFFECTS	Optional	REJECT-1
	Optional	ACCEPT-1

<u>Vote-1.Effects</u>

REJECT-1

AGENT	Sem	Neg-Trans.Accompanier, HUMAN
THEME	Sem	EXCHANGE-2, EVENT, OBJECT
BENEFICIARY	Sem	Neg-Trans.Agent, HUMAN
PRECONDITIONS	Sem	PROPOSE-1, EVENT

<u>Vote-1.Effects</u>

ACCEPT-1

AGENT	Sem	Neg-Trans.Accompanier, HUMAN
THEME	Sem	EXCHANGE-2, EVENT, OBJECT
BENEFICIARY	Sem	Neg-Trans.Agent, HUMAN

PRECONDITIONS	Sem	PROPOSE-1, HUMAN
EFFECTS	Sem	EXCHANGE-1, EVENT

Neg-Trans.Has-Parts

PROPOSE-2

AGENT	Sem	Neg-Trans.Agent, Neg-Trans. Accompanier, HUMAN
THEME	Sem	EXCHANGE, EVENT
BENEFICIARY	Sem	Neg-Trans.Agent, Neg-Trans. Accompanier, HUMAN
EFFECTS	Sem	VOTE-2, EVENT

That is, following the accompanier's rejection of the initial proposal by the agent, either party is now free to make a new proposal.

Propose-2.Effects

VOTE-2

AGENT	Sem	Neg-Trans.Agent, Neg-Trans. Accompanier, HUMAN
THEME	Sem	PROPOSE-2, EVENT
EFFECTS	Optional	REJECT-2, EVENT
	Optional	ACCEPT-2, EVENT

Vote-2.Effects

ACCEPT-2

AGENT	Sem	Neg-Trans.Agent, Neg-Trans. Accompanier, HUMAN
THEME	Sem	PROPOSE-2
EFFECTS	Sem	EXCHANGE

Vote-2.Effects

REJECT-2

AGENT	Sem	Neg-Trans.Agent, Neg-Trans. Accompanier, HUMAN
THEME	Sem	PROPOSE-2
EFFECTS	Optional	PROPOSE-2, EVENT

At this point, if PROPOSAL-2 has been rejected (by either party), the optional consequence is a new proposal; thus, the complex event loops back to PROPOSAL-2. This is, however, not exactly accurate, since the THEME of a third proposal will not be the same as for the first. What is needed here is for the THEME slot filler to be "erased" upon reiteration of PROPOSAL-2, to be replaced by some new terms for exchange. This appears to be beyond the capabilities of ontological frame representation.

<u>Neg-Trans.Effects</u>

EXCHANGE

AGENT	Sem	Neg-Trans.Agent, HUMAN
ACCOMPANIER	Sem	Neg-Trans.Accompanier, HUMAN
INSTRUMENT	Sem	Neg-Trans.Instrument, EVENT, OBJECT
THEME	Sem	Neg-Trans.Theme, EVENT, OBJECT
PRECONDITIONS	Sem	ACCEPT-1, ACCEPT-2, EVENT

That is, if the terms of the exchange proposed have been accepted, then the exchange takes place. In perhaps a large number of instances, the exchange will be of money for goods/services. If so, then NEGOTIATE-TRANSACTION can be linked to either BUY-OBJECT or BUY-SERVICE. Or, to look at it from the other angle, it is certainly possible that while engaged in a buy-event, the terms of that exchange will have to be negotiated.

4.4 Complex events clustered near BUY in the ontological hierarchy

In this section, three complex events are developed for concepts similar in meaning to BUY, namely LEND, RENT, and AUCTION.

4.4.1 The LEND complex event

Although the English word "lend" may be used to mean either a bank loan or a loan between friends, the ontological concept LEND is restricted to this former usage.

Like BUY, a LEND event implies an exchange. Because LEND allows for the lending of both money and commodities, there is motivation to separate the two.

"Lend-Money" appears to be sufficiently different from a concept such as "Lend-Commodity," given that (i) commodities lent generally require a premium or fee, not interest and (ii) that returning a commodity and returning money entail much different subevents, such as, in the case of lending money, the number of terms for the "giving back" of the money originally given—e.g., an interest rate, loan length, number of payments—as well as preconditions for the loan being granted—e.g., the lendee must be credit-worthy (have a job, have no/few late payments on other loans). It seems reasonable that "Lend-Commodity" and RENT might in fact be the same concept, in which case the LEND concept could be altered so as to only entail "lending of money at interest." In fact, this is what will be done here. The LEND complex event is developed as follows.

Frame: LEND

Slot	Facet	Filler
DEFINITION	Value	"to let out money at interest"
IS-A	Value	EVERYDAY-FINANCIAL-EVENT
SUBCLASSES	Value	[None]
AGENT	Sem	HUMAN
	Relax-to	CORPORATION
BENEFICIARY	Sem	HUMAN
	Relax-to	CORPORATION
DESTINATION	Sem	HUMAN
	Relax-to	CORPORATION
LOCATION	Sem	PLACE
SOURCE	Sem	PLACE
THEME	Sem	(>0)
	Meas-Unit	MONETARY-UNIT
PRECONDITIONS	Sem	DESIRE
	Sem	OWN-1
HAS-PARTS	Sem	APPLY-FOR-1

		Sem	VOTE
		Sem	GIVE-1
EFFECTS		Sem	OWN-2
		Sem	OWE
		Sem	ACCRUE
		Sem	PAY-1
		Sem	AMORTIZATION-1

<u>Lend.Preconditions</u>

DESIRE			
	AGENT	Sem	Lend.Beneficiary, HUMAN
	THEME	Sem	(>0)
		Meas-Unit	MONETARY-UNIT

<u>Lend.Preconditions</u>

OWN-1			
	AGENT	Sem	Lend.Agent, HUMAN
	THEME	Sem	(>= Desire.Theme)
		Meas-Unit	MONETARY-UNIT

<u>Lend.Has-Parts</u>

APPLY-FOR-1			
	AGENT	Sem	Lend.Beneficiary, HUMAN
	THEME	Sem	Desire.Theme
		Meas-Unit	MONETARY-UNIT
	EFFECTS	Sem	VOTE-1

<u>Lend.Has-Parts</u>

VOTE-1			
	AGENT	Sem	Lend.Agent, HUMAN
	THEME	Sem	Apply-For-1.Theme, EVENT
	PRECONDITIONS	Sem	Apply-For-1, EVENT
	EFFECTS	Optional	Reject-1, EVENT
		Optional	Accept-1, EVENT

<u>Vote-1.Effects</u>

REJECT-1			
	AGENT	Sem	Lend.Agent, HUMAN
	THEME	Sem	Apply-For-1.Theme, EVENT

We do not specify, as a consequence of REJECT-1, that no loan is thereby offered. Rather, a precondition of the effects of the loan being given will be that the application was accepted.

<u>Vote-1.Effects</u>

ACCEPT-1

AGENT	Sem	Lend.Agent, HUMAN
THEME	Sem	Apply-For-1.Theme, EVENT
PRECONDITIONS	Default	OWN-3, EVENT
EFFECTS	Sem	OWN-2, EVENT
	Sem	OWE, EVENT
	Sem	PAY-1, EVENT

<u>Accept-1.Precon</u>

OWN-3

AGENT	Sem	Lend.Beneficiary, HUMAN
THEME	Sem	(Lend.Beneficiary.Assets > Lend.Beneficiary.Liabilities)
	Meas-Unit	MONETARY-UNIT

That is, it is assumed that a precondition of the bank's approving the loan application is that the loan applicant has net assets and not net liabilities. To be more specific, the complex event would have to be developed several subevents deeper, perhaps specifying preconditions such as that the applicant is employed, has bills but pays them on time, has enough money to cover the monthly payment, etc. Such information only seems relevant when dealing with highly specialized texts, and so we go no deeper here.

<u>Lend.Has-Parts</u>

GIVE-1

AGENT	Sem	Lend.Agent, HUMAN
THEME	Sem	(=< Apply-For-1.Theme)
	Meas-Unit	MONETARY-UNIT
EFFECTS	Sem	OWN-2, EVENT

		Sem	OWE, EVENT
		Sem	PAY-1, EVENT

That is, the bank will give the applicant no more than the amount requested and perhaps less. Furthermore, doing so sets up expectations that the applicant will own the money, will therefore owe the bank, and will be required to give the money back

Lend.Effects			
OWN-2			
	AGENT	Sem	Lend.Beneficiary, HUMAN
	THEME	Sem	GIVE-1.THEME, OBJECT
		Default	Lend.Theme
Lend.Effects			
OWE			
	AGENT	Sem	Lend.Beneficiary, HUMAN
	THEME	Sem	Give-1.Theme, EVENT
	BENEFICIARY	Sem	Lend.Agent, HUMAN
Lend.Effects			
ACCRUE			
	THEME	Sem	Give-1.Theme
	PRECONDITION	Sem	GIVE-1, EVENT
Lend.Effects			
PAY-1			
	AGENT	Sem	Lend.Beneficiary, HUMAN
	THEME	Sem	(% * Give-1.Theme)
		Meas-Unit	MONETARY-UNIT
	PRECONDITIONS	Sem	(Give-1.Theme > 0)
		Meas-Unit	MONETARY-UNIT
	BENEFICIARY	Sem	Lend.Agent, HUMAN
	EFFECTS	Sem	PAY-1
		Sem	AMORTIZATION-1, Event

Since the PAY-1 is presumed to be the same amount, PAY-1 can be specified as an EFFECT, thereby looping the event, with the precondition that the outstanding balance is greater than zero. Furthermore, paying gives rise to loan amortization.

Lend.Effects

AMORTIZATION-1			
	THEME	Sem	Give-1.Theme, OBJECT
	PRECONDITION	Sem	PAY-1, EVENT
	EFFECTS	Sem	DECREASE-1, EVENT

Amortization-1.Effects

DECREASE-1			
	THEME	Sem	(Give-1.Theme = 0)
		Meas-Unit	MONETARY-UNIT

In the EFFECTS slot for LEND, we do not specify the interest rate at which the loan is made, the length of the loan, the periodic payment, or how often those payments must be made. However, this slot does specify that interest will accrue on the loan amount (ACCRUE), that the lendee will need to pay back the money (PAY-1), and that the loan amount will therefore gradually decrease (AMORTIZATION-1), until it reaches zero.

4.4.2 The RENT complex event

The concept RENT is very similar to BUY-OBJECT, especially since RENT only takes an OBJECT as THEME. The most apparent difference between the two concepts is that BUY-OBJECT implies a transfer of ownership, while RENT, like BUY-SERVICE, does not. Second, a RENT event generally implies the signing of a contract, specifying the conditions of the object's use by the renter. Typically, rental objects are "big-ticket" items, such as houses, office space, cars, furniture, appliances and the like, although this is difficult to represent in a complex event concept. Furthermore, much like LEND, RENT may often denote long-term rental, with periodic payments, although interest is not involved. Given these similarities, RENT is a good indicator of the sensitivity of the complex-event knowledge base.

Frame: RENT

Slot	Facet	Filler
DEFINITION	Value	"to pay at regular fixed intervals for the use of an item such as a house or a piece of land"
IS-A	Value	EVERYDAY-FINANCIAL-EVENT
SUBCLASSES	Value	[None]
AGENT	Sem	HUMAN
	Relax-to	CORPORATION
BENEFICIARY	Sem	HUMAN
	Relax-to	CORPORATION
LOCATION	Sem	PLACE
THEME	Sem	HOUSE, AUTOMOBILE
	Default	OBJECT
PRECONDITIONS	Sem	DESIRE-1
	Default	OWN
	Sem	REQUEST
	Sem	DESIRE-2
HAS-PARTS	Default	INSPECT
	Default	NEGOTIATE-TRANSACTION
	Default	SIGN-EVENT-1
	Sem	PAY-1
EFFECTS	Optional	INHABIT
	Sem	USE

<u>Rent.Preconditions</u>

DESIRE-1

AGENT	Sem	Rent.Beneficiary, HUMAN
THEME	Sem	Rent.Theme, OBJECT

<u>Rent.Preconditions</u>

OWN

AGENT	Sem	Rent.Agent, HUMAN
THEME	Sem	Rent.Theme, Desire-1.Theme, HUMAN

<u>Rent.Preconditions</u>

REQUEST

AGENT	Sem	Rent.Beneficiary, HUMAN
THEME	Sem	Rent.Theme, EVENT

BENEFICIARY	Sem	Rent.Agent, HUMAN

<u>Rent.Preconditions</u>

DESIRE-2

AGENT	Sem	Rent.Accompanier, HUMAN
THEME	Sem	Desire-1, EVENT

That is, the owner of the object also wants the renter to rent the object.

<u>Rent.Has-Parts</u>

INSPECT

AGENT	Sem	Rent.Beneficiary, HUMAN
THEME	Sem	Rent.Theme, OBJECT

<u>Rent.Has-Parts</u>

NEGOTIATE-TRANSACTION

AGENT	Sem	Rent.Agent, HUMAN
ACCOMPANIER	Sem	Rent.Beneficiary, HUMAN
THEME	Sem	Rent.Theme, EVENT
PRECONDITIONS	Sem	DESIRE-2, EVENT
HASPARTS	Sem	PROPOSE-1, EVENT
	Sem	ACCEPT-1, EVENT
EFFECTS	Sem	SIGN-EVENT-1, EVENT

<u>Negotiate-Transaction.Has-Parts</u>

PROPOSE-1

AGENT	Sem	Rent.Agent, HUMAN
THEME	Sem	Contract, OBJECT
BENEFICIARY	Sem	Rent.Accompanier, HUMAN
EFFECTS	Default	ACCEPT-1, EVENT

<u>Negotiate-Transaction.Has-Parts</u>

ACCEPT-1

AGENT	Sem	Rent.Beneficiary, HUMAN
THEME	Sem	Propose-1.Theme, OBJECT
BENEFICIARY	Sem	Rent.Agent, HUMAN
PRECONDITIONS	Sem	PROPOSE-1, EVENT
EFFECTS	Default	SIGN-EVENT-1, EVENT

<u>Rent.Has-Parts</u>

SIGN-EVENT-1

AGENT	Sem	Rent.Agent, HUMAN
ACCOMPANIER	Sem	Rent.Beneficiary, HUMAN

	THEME	Sem	Propose-1.Theme, OBJECT
	PRECONDITIONS	Sem	ACCEPT-1, EVENT
	EFFECTS	Sem	PAY-1, EVENT

<u>Rent.Has-Parts</u>

PAY-1			
	AGENT	Sem	Rent.Beneficiary, HUMAN
	THEME	Sem	(>0)
		Meas-Unit	MONETARY-UNIT
	BENEFICIARY	Sem	Rent.Agent, HUMAN
	PRECONDITIONS	Default	SIGN-EVENT-1, EVENT
		Sem	ACCEPT-1, EVENT
	EFFECTS	Optional	PAY-1, EVENT

<u>Rent.Effects</u>

INHABIT			
	AGENT	Sem	Rent.Beneficiary, HUMAN
	THEME	Sem	Rent.Theme, OBJECT
	PRECONDITIONS	Sem	GIVE-1, EVENT

<u>Rent.Effects</u>

USE			
	AGENT	Sem	Rent.Beneficiary, HUMAN
	THEME	Sem	Rent.Theme, OBJECT
	PRECONDITIONS	Sem	GIVE-1, EVENT

In other words, the GIVE-1 event has as a consequence a reiteration of that event. Its facet is specified as Optional, to prevent infinite assertion of the event and to handle cases where payment is only made once, e.g., renting a car. Regardless, the criterial EFFECT of a RENT event is that the renter uses the rental object.

4.4.3 The AUCTION complex event

The AUCTION complex event is a good test for the formalism, since AUCTION is descended from BUY-OBJECT.[18] Thus, if the complex-event knowledge base, as developed in this dissertation, is to be viable, each subevent of BUY-OBJECT should also true of AUCTION. For reference, (5) lists the subevents of BUY-OBJECT:

(5)	PRECONDITIONS	Sem	DESIRE
		Sem	OWN
		Optional	LEND
		Optional	WITHDRAW-ATM
		Optional	DIAL
	HAS-PARTS	Optional	LOCATE
		Optional	CHANGE-LOCATION
		Optional	NEGOTIATE-TRANSACTION
		Default	GIVE
		Sem	PAY
	EFFECTS	Sem	TRANSFER-POSSESSION
		Sem	DECREASE

However, since the ontological hierarchy specifies that children concepts only inherit slots with Sem facets, AUCTION inherits only the following subevents from BUY-OBJECT:

(6)	PRECONDITIONS	Sem	DESIRE
		Sem	OWN
	HAS-PARTS	Sem	PAY
	EFFECTS	Sem	TRANSFER-POSSESSION
		Sem	DECREASE

This is exactly as needed: each of these subevents is criterial for AUCTION. Furthermore, several subevents specified as Optional or Default in BUY-OBJECT (that is, WITHDRAW-ATM, LEND, DIAL, LOCATE, CHANGE-LOCATION, NEGOTIATE-TRANSACTION, and GIVE) are not inherited; none is criterial for AUCTION—again, exactly as needed. However, the formalism does allow a child concept to specify, in its own frame, any or all of its parent's non-inherited, Optional/Default subevents.

[18] Excluded are "charity auctions" in which people bid on a date with another person. Such events should be mapped to the concept DONATE.

It may, at first, seem counterintuitive that an Optional property of a parent concept is an excluded property in a child concept. Without question, an Optional property of a parent concept that is excluded for *all* its children concepts should be strictly prohibited by the principles of ontological design. However, it may be motivated in cases where at least one child concept does express the parent concept's Optional property in its own frame, thus elminating the need to create a new branch in the ontology for the child concept with an excluded property. This is true of the subevent NEGOTIATE-TRANSACTION, since it will be a Sem subevent of the BUYOUT concept, which is a child of BUY-OBJECT. A consequence of this design is that the ontological hierarchy becomes an "And-Or graph," where children concepts are allowed to be disjunctions of the parent concept properties (see Mahesh 1996 for a discussion). Thus, BUY-OBJECT, AUCTION, and BUYOUT may differ in the following way:

(7)
(a) BUY-OBJECT ∋ (PAY) ∧ (TRANSFER-POSSESSION) ∧
((NEGOTIATE-TRANSACTION) ∨ ¬(NEGOTIATE-TRANSACTION))
(b) BUYOUT ∋ (PAY) ∧ (TRANSFER-POSSESSION) ∧
((NEGOTIATE-TRANSACTION) ∨ ¬(NEGOTIATE-TRANSACTION))
(c) AUCTION ∋ (PAY) ∧ (TRANSFER-POSSESSION) ∧ ¬(NEGOTIATE-TRANSACTION)

That is, both BUY-OBJECT and its child, BUYOUT include the subevents PAY and TRANSFER-POSSESSION and optionally include NEGOTIATE-TRANSACTION, while AUCTION includes PAY and TRANSFER-POSSESSION and excludes NEGOTIATE-TRANSACTION. Instead, a BID event will be motivated for AUCTION.[19]

[19] Another option, in this case, might be to specify that BID is a child concept of NEGOTIATE-TRANSACTION. Then, according to the model above, NEGOTIATE-TRANSACTION would become a Sem facet of BUY-OBJECT (since each of its children would inherit the subevent), thus reinstating BUY-OBJECT as an "And graph," i.e., its children would be a conjunction of its properties. BID, in turn, as a subevent of AUCTION, would inherit all Sem properties of NEGOTIATE-TRANSACTION, as well special specifications,

The following is the complex event for AUCTION. (The Sem and Optional subevents that are also specified in BUY-OBJECT are shown in the AUCTION frame, though variable-bindings are not, since the variable-bindings are identical to those for BUY-OBJECT; see section 4.2 for these variable-bindings.)

Frame: AUCTION

Slot	Facet	Filler
DEFINITION	Value	“to sell an object to the highest bidder”
IS-A	Value	BUY-OBJECT
SUBCLASSES	Value	[None]
AGENT	Sem	HUMAN
	Relax-to	CORPORATION
ACCOMPANIER	Sem	HUMAN
	Relax-to	CORPORATION
THEME	Sem	OBJECT
BENEFICIARY	Sem	HUMAN
	Relax-to	CORPORATION
LOCATION	Sem	PLACE
PRECONDITIONS	Sem	DESIRE-1 [from BUY-OBJECT]
	Sem	OWN-1 [from BUY-OBJECT]
	Optional	LEND-1
	Optional	WITHDRAW-ATM
	Default	ROGATIVE-ACT-1
HAS-PARTS	Default	ROGATIVE-ACT -2
	Sem	BID-1
	Sem	PAY-1 [from BUY-OBJECT]
EFFECTS	Sem	TRANSFER-POSSESSION [from BUY-OBJECT]
	Sem	DECREASE-1 [from BUY-OBJECT]
Auction.Preconditions		
ROGATIVE-ACT-1		
AGENT	Sem	Auction.Agent, HUMAN
	Relax-to	CORPORATION
BENEFICIARY	Sem	HUMAN

such as that more than two parties may be involved in the transaction, etc. This possibility is, however, not pursued in this dissertation.

THEME	Sem	ROGATIVE-ACT-2, EVENT
PRECONDITIONS	Sem	OWN-1, EVENT

That is, the default assumption is that the person (or corporation) selling merchandise through an auction will ask another person, the auctioneer, to ask possible buyers to buy the merchandise.

<u>Auction.Has-Parts</u>

ROGATIVE-ACT-2

AGENT	Sem	Rogative-Act-1.Beneficiary, HUMAN
ACCOMPANIER	Sem	Auction.Agent, HUMAN
	Relax-to	CORPORATION
BENEFICIARY	Sem	HUMAN
THEME	Sem	Auction.Theme, OBJECT
EFFECTS	Sem	BID-1, EVENT

In other words, the auctioneer, the proxy for (or accompanier of) the seller, asks a group of people to bid on the object up for auction. Note that, since auctioneers generally do not have a corporation as audience, ROGATIVE-ACT-2 is specified as a Default subevent in which the BENEFICIARY role is not relaxed to CORPORATION. (A corporation may, however, fulfill the ACCOMPANIER role in this subevent, as in the case of an auction house.)

<u>Auction.Has-Parts</u>

BID-1

AGENT	Sem	Human
	Relax-to	CORPORATION
THEME	Sem	Auction.Theme, OBJECT
INSTRUMENT	Sem	(>0)
	Meas-Unit	MONETARY-UNIT
BENEFICIARY	Sem	Auction.Agent, HUMAN
	Relax-to	CORPORATION

EFFECTS	Default	BID-2, EVENT
	Optional	PAY-1, EVENT
	Optional	TRANSFER-POSSESSION, EVENT
	Optional	DECREASE-1, EVENT

Bid-1.Effects

BID-2

AGENT	Sem	HUMAN
	Relax-to	CORPORATION
THEME	Sem	Auction.Theme, OBJECT
INSTRUMENT	Sem	(>0)
	Meas-Unit	MONETARY-UNIT
BENEFICIARY	Sem	Auction.Agent, HUMAN
	Relax-to	CORPORATION
PRECONDITIONS	Sem	(Bid-2.Instrument > Bid-1.Instrument)
	Meas-Unit	MONETARY-UNIT
EFFECTS	Optional	BID-2, EVENT
	Optional	Pay-1, EVENT
	Optional	TRANSFER-POSSESSION, EVENT
	Optional	DECREASE-1, EVENT

Thus, after a bid on an object, another bid may be offered, provided of course that the second bid is higher than the first. The BID-2 event then may loop, until no bid higher than the last is made.

Auction.Effects

PAY-1

AGENT	Sem	Bid-2.Agent, HUMAN
BENEFICIARY	Sem	Auction.Agent, HUMAN
INSTRUMENT	Sem	Bid-2.Theme, MONEY
PRECONDITIONS	Sem	BID-2, EVENT
EFFECTS	Sem	TRANSFER-POSSESSION-1
	Sem	OWN-1

PAY-1 is not decomposed any further here, since it has already been described in section 4.3.2. The PAY event, of course, can be linked to AUCTION to yield subevents of greater depth and specificity.

Auction.Effects

TRANSFER-POSSESSION-1

AGENT	Sem	Auction.Agent, HUMAN
	Relax-to	CORPORATION
THEME	Sem	Auction.Theme, OBJECT
BENEFICIARY	Sem	Bid-2.Agent, HUMAN

Auction.Effects

OWN-1

AGENT	Sem	Bid-2.Agent, HUMAN
THEME	Sem	Auction.Theme, OBJECT

Thus, further evidence of the formalism's viability is that a complex event of a child concept can seamlessly incorporate the complex-event knowledge of its parent concept.

By way of summarizing section 4.4, it is noted that both LEND and RENT are similar to BUY in that all three specify that an exchange is made and that, furthermore, money is involved: either money-for-money (LEND), money-for-object (RENT and BUY-OBJECT), or money-for-event (BUY-SERVICE). As these complex events clearly show, however, there are many differences in entailments and component subevents that would be missed by a system unarmed with such knowledge. Furthermore, the complex event for AUCTION provides evidence that even when some of these subevents are the same for a parent as for a child concept, such information can be handled by the formalism for representing complex-event knowledge.

4.5 The BANKRUPTCY complex event

While the complex events shown so far in this section have specified commonsense knowledge the average speaker might be said to have, there are many other events in the world that require specialized knowledge to be fully understood; that is, such events require experts. Some examples are procedures for conducting a scientific experiment, installing a transmission for an automobile, and being engaged in a bankruptcy. In each case, a non-expert, of course, will know something about these events. And, furthermore, if the texts to be encountered by the complex-event knowledge base are intended for such a general audience, then there is only a need to specify as many subevents as a "general reader" might need.

However, if the texts to be encountered require expert knowledge of the event, then the complex-event knowledge base should be able to provide this knowledge. For example, suppose the task of the system is to understand a corporation's financial statements, quarterly reports, and assessments of that corporation, with respect to its relation to a bankruptcy. While such texts will make use of a specialized vocabulary, they will are also likely to make reference to a specialized event structure. Therefore, in this section, the complex event for BANKRUPTCY is developed in order to demonstrate the formalism's ability to (i) specify expert knowledge to handle expert texts, and (ii) represent this knowledge to a depth of many levels.

The original ontological frame for BANKRUPTCY, a child of CORPORATE-EVENT, is as follows.

Frame: BANKRUPTCY

Slot	Facet	Filler
DEFINITION	Value	"to be legally unable to pay one's debts"
IS-A	Value	CORPORATE-EVENT
SUBCLASSES	Value	[None]
AGENT	Sem	HUMAN
LOCATION	Sem	PLACE

THEME	Sem	EVENT

A BANKRUPTCY event, may follow any number of courses. For the purposes of this complex event, it is assumed that the corporation involved in a bankruptcy is a United States company and therefore subject its the bankruptcy laws. (Personal bankruptcies are ignored in this complex event.) Bankruptcies in the U.S. are governed by the U.S. Bankruptcy Code. Usually, a corporation filing for bankruptcy appeals for protection under one of two chapters of the Bankruptcy Code, Chapter 7 or Chapter 11. Chapter 7 bankruptcy leads to the dissolution of the corporation, while Chapter 11 attempts a reorganization. Both entail quite different subevents, and so BANKRUPTCY has been given the children concepts BANKRUPTCY-CHAPTER-SEVEN and BANKRUPTCY-CHAPTER-ELEVEN, which will be abbreviated to CHAPTER-SEVEN and CHAPTER-ELEVEN, respectively. Shared subevents for the two are specified in the complex event for their parent concept, BANKRUPTCY as follows.

Frame: BANKRUPTCY

Slot	Facet	Filler
PRECONDITIONS	Default	Bankruptcy.Agent, OBJECT
	Sem	OWE-1
	Sem	LEND
	Sem	GREATER-THAN
	Sem	APPLY-FOR-1
HAS-PARTS	Sem	MEETING
EFFECTS	Default	CANCEL-OBLIGATION

The following specify the variable-bindings for the subevents of BANKRUPTCY.

Bankruptcy.Preconditions		
Bankruptcy.Agent		
LOCATION	Sem	UNITED-STATES-OF-AMERICA

	ASSETS	Sem	(>= 0)
		Meas-Unit	MONETARY-UNIT
	LIABILITIES	Sem	(>0)
		Meas-Unit	MONETARY-UNIT

<u>Bankruptcy.Preconditions</u>

OWE-1			
	AGENT	Sem	Bankruptcy.Agent, CORPORATION
	BENEFICIARY	Default	BANK
		Sem	CORPORATION
	THEME	Sem	(>0)
		Meas-Unit	MONETARY-UNIT

<u>Bankruptcy.Preconditions</u>

LEND			
	AGENT	Sem	Owe-1.Beneficiary, BANK
	BENEFICIARY	Sem	Bankruptcy.Agent, CORPORATION
	THEME	Sem	Owe-1.Theme, OBJECT

<u>Bankruptcy.Preconditions</u>

GREATER-THAN			
	DOMAIN	Sem	MONETARY-UNIT
	RANGE	Sem	(Bankruptcy.Agent.Liabilities > Bankruptcy.Agent. Assets)

<u>Bankruptcy.Preconditions</u>

APPLY-FOR-1			
	AGENT	Sem	Bankruptcy.Agent, CORPORATION
	BENEFICIARY	Sem	COURT, OBJECT
	THEME	Sem	Bankruptcy.Theme, EVENT

<u>Bankruptcy.Has-Parts</u>

MEETING			
	AGENT	Sem	Apply-For-1.Beneficiary, OBJECT
	ACCOMPANIER	Sem	Bankruptcy.Agent, Owe.Beneficiary COURT-TRUSTEE[20]
	LOCATION	Sem	PLACE
		Default	COURTROOM
	PRECONDITIONS	Sem	APPLY-FOR-1, EVENT
	HAS-PARTS	Sem	INFORM-1, EVENT
	EFFECTS	Default	CANCEL-OBLIGATION-1

[20] COURT-TRUSTEE is a putative new SOCIAL-ROLE concept.

That is, once a corporation has filed for bankruptcy, there will be a meeting of the corporations reprentatives, the creditors (i.e., the Owe.Beneficiary), and the court-appointed trustee.

Meeting.Has-Parts

INFORM-1

AGENT	Sem	Bankruptcy.Agent, CORPORATION
BENEFICIARY	Sem	Meeting.Accompanier, HUMAN
THEME	Domain	INFORMATION
	Range	Bankruptcy.Agent.Address, Bankruptcy.Agent.Liabilities, Bankruptcy.Agent.Assets

In other words, at this meeting of the bankrupt company, its creditors, and the court trustee, the corporation will be required to confirm information such as its address, liabilities, and assets. (Other information could be specified, but seems unnecessary here.)

Bankruptcy.Effects

CANCEL-OBLIGATION-1

AGENT	Sem	Apply-For-1.Beneficiary, COURT
BENEFICIARY	Sem	Bankruptcy.Agent, CORPORATION
THEME	Sem	Owe-1.Theme, OBJECT
EFFECTS	Sem	(Bankruptcy.Agent.Liabilities = 0)
	Meas-Unit	MONETARY-UNIT

Following are the complex events for the children concepts of BANKRUPTCY, BANKRUPTCY-CHAPTER-SEVEN and BANKRUPTY-CHAPTER-ELEVEN.

Frame: CHAPTER-SEVEN

Slot	Facet	Filler
DEFINITION	Value	"to be engaged in a Chapter 7 bankruptcy"

IS-A	Value	BANKRUPTCY
SUBCLASSES	Value	[None]
AGENT	Sem	HUMAN
	Relax-to	CORPORATION
LOCATION	Sem	PLACE
THEME	Sem	LIQUIDATE-1
PRECONDITIONS	Sem	RETAIN-1
	Sem	APPLY-FOR-1
HAS-PARTS	Sem	DESIGNATE-1
	Sem	POSTPONE-1
	Sem	LITIGATION-1
	Sem	MEETING
	Default	LIQUIDATE-1
	Default	PAY-2
EFFECTS	Sem	CANCEL-OBLIGATION-1
	Sem	EQUAL-TO
	Optional	DISSOLUTION
	Sem	EVENT

<u>Chapter-Seven.Preconditions</u>

RETAIN-1

AGENT	Sem	Chapter-Seven.Agent.CORPORATION
THEME	Sem	ATTORNEY, HUMAN

<u>Chapter-Seven.Preconditions</u>

APPLY-FOR-1

AGENT	Sem	Chapter-Seven.Agent, CORPORATION
	Relax-to	ATTORNEY
BENEFICIARY	Sem	COURT
THEME	Sem	Chapter-Seven.Theme, EVENT
PRECONDITIONS	Sem	GREATER-THAN-1
	Sem	RETAIN-1
HAS-PARTS	Sem	INFORM-1
	Sem	PAY-1
EFFECTS	Sem	DESIGNATE-1
	Default	POSTPONE-1
	Sem	MEETING-1
	Sem	LITIGATION-1

<u>Apply-For-1.Has-Parts</u>

INFORM-1

AGENT	Sem	Chapter-Seven.Agent, CORPORATION
	Relax-to	ATTORNEY
BENEFICIARY	Sem	Apply-For-1.Beneficiary, COURT
THEME	Domain	INFORMATION

	Range	Chapter-Seven.Agent.Address, Chapter-Seven.Agent. Liabilities,Chapter-Seven. Agent.Assets

<u>Apply-For-1.Has-Parts</u>

Pay-1

Agent	Sem	Chapter-Seven.Agent, Corporation
Beneficiary	Sem	Apply-For-1.Beneficiary, Court
Theme	Sem	(<100000)
	Meas-Unit	Monetary-Unit
Effects	Sem	Designate-1

That is, as part of the bankruptcy application process, the corporation is required to pay a fee, listed here only as below $100,000.

<u>Chapter-Seven.Has-Parts</u>

Designate-1

Agent	Sem	Apply-For-1.Beneficiary, Court
Theme	Sem	Court-Trustee
Preconditions	Sem	Apply-For-1, Event
		Pay-1, Event
Effects	Sem	Litigation-1, Event
	Sem	Meeting-1, Event

That is, following a bankruptcy application, the court appoints a trustee to oversee the bankruptcy proceedings.

<u>Chapter-Seven.Has-Parts</u>

Postpone-1

Agent	Sem	Apply-For-1.Beneficiary, Court
Beneficiary	Sem	Chapter-Seven.Agent, Object
Theme	Sem	Owe-1.Theme, Event

That is, the court postpones the bankrupt corporation’s responsibility to pay back debt.

<u>Chapter-Seven.Has-Parts</u>

LITIGATION-1			
	AGENT	Sem	Chapter-Seven.Agent, HUMAN
		Relax-to	ATTORNEY
	ACCOMPANIER	Sem	Owe-1.Beneficiary, HUMAN
		Relax-to	ATTORNEY
	THEME	Sem	Chapter-Seven.Theme, EVENT
	LOCATION	Sem	COURTROOM

<u>Chapter-Seven.Has-Parts</u>

MEETING			
	AGENT	Sem	Chapter-Seven.Agent, HUMAN
		Relax-to	ATTORNEY
	ACCOMPANIER	Sem	Owe-1.Beneficiary, HUMAN
		Relax-to	ATTORNEY
	LOCATION	Sem	COURTROOM
	PRECONDITIONS	Sem	APPLY-FOR-1, EVENT
		Sem	PAY-1, EVENT
		Sem	DESIGNATE-1, EVENT
	HAS-PARTS	Sem	NEGOTIATE-TRANSACTION-1
		Sem	PROPOSE-1
	EFFECTS	Default	LIQUIDATE-1
		Default	PAY-2

At this "creditor's meeting," it is decided how the bankrupt corporation will liquidate its assets in order to partially pay back its creditors.

<u>Meeting-1.Has-Parts</u>

NEGOTIATE-TRANSACTION-1			
	AGENT	Sem	Chapter-Seven.Agent, CORPORATION
		Relax-to	ATTORNEY
	ACCOMPANIER	Sem	Owe-1.Beneficiary, HUMAN
		Relax-to	ATTORNEY
	THEME	Sem	LIQUIDATE-1, EVENT
	EFFECTS	Default	PAY-2

<u>Meeting-1.Has-Parts</u>

PROPOSE-1			
	AGENT	Sem	Chapter-Seven.Agent, CORPORATION
		Relax-to	ATTORNEY
	ACCOMPANIER	Sem	Owe-1.Beneficiary, HUMAN
		Relax-to	ATTORNEY
	THEME	Sem	LIQUIDATE-1, EVENT

EFFECTS	Default	PAY-2

Chapter-Seven.Has-Parts

LIQUIDATE-1

AGENT	Sem	Chapter-Seven.Agent, CORPORATION
THEME	Sem	Chapter-Seven.Agent.Assets
EFFECTS	Sem	PAY-2

Chapter-Seven.Has-Parts

PAY-2

AGENT	Sem	Chapter-Seven.Agent, CORPORATION
BENEFICIARY	Sem	Owe-1.Beneficiary, HUMAN
THEME	Sem	Liquidate-1.Theme
PRECONDITIONS	Sem	LIQUIDATE-1
EFFECTS	Sem	CANCEL-OBLIGATION-1

That is, the bankrupt corporation liquidates its assets and then uses that money to pay off its creditors. This has the effect of cancelling its previous obligations to pay back those creditors.

Chapter-Seven.Effects

CANCEL-OBLIGATION-1

AGENT	Sem	Apply-For-1.Theme, COURT
BENEFICIARY	Sem	Chapter-Seven.Agent, CORPORATION
THEME	Sem	Owe-1.Theme, EVENT

Chapter-Seven.Effects

EQUAL-TO

DOMAIN	Sem	(Chapter-Seven.Agent. Liabilities = 0) and (Chapter-Seven.Agent. Assets =0)
RANGE	Meas-Unit	MONETARY-UNIT

Chapter-Seven.Effects

DISSOLUTION

AGENT	Sem	Chapter-Seven.Agent, CORPORATION
THEME	Sem	Chapter-Seven.Agent, CORPORATION

Chapter-Seven.Effects

EVENT		
AGENT	Sem	Chapter-Seven.Agent, CORPORATION
THEME	Sem	EVENT (Not BANKRUPTCY)

The complex event for CHAPTER-ELEVEN follows. While many of the preconditions for both CHAPTER-SEVEN and CHAPTER-ELEVEN are the same, there is a much richer structure of component subevents (PRECONDITIONS, HAS-PARTS, and EFFECTS) in CHAPTER-ELEVEN.

Frame: CHAPTER-ELEVEN

Slot	Facet	Filler
DEFINITION	Value	"to be engaged in a Chapter 11 bankruptcy"
IS-A	Value	BANKRUPTCY
SUBCLASSES	Value	[None]
AGENT	Sem	HUMAN
	Relax-to	CORPORATION
LOCATION	Sem	PLACE
THEME	Sem	REORGANIZATION, BUYOUT
PRECONDITIONS	Sem	OWE-1, EVENT
	Sem	LEND-1, EVENT
	Sem	RETAIN-1, EVENT
	Sem	APPLY-FOR-1, EVENT
	Optional	PETITION-1, EVENT
HAS-PARTS	Sem	DESIGNATE-1, EVENT
	Sem	POSTPONE-1, EVENT
	Sem	LITIGATION-1, EVENT
	Optional	PROPOSE-1, EVENT
	Optional	PROPOSE-2, EVENT
	Default	DECREASE
	Optional	REORGANIZATION-1, EVENT
	Optional	BUYOUT-1, EVENT
EFFECTS	Optional	APPLY-FOR-2, EVENT
	Sem	EVENT-1

<u>Chapter-Eleven.Preconditions</u>

OWE-1			
	AGENT	Sem	Chapter-Eleven.Agent, CORPORATION
	BENEFICIARY	Sem	BANK

Chapter-Eleven.Preconditions

LEND-1			
	AGENT	Sem	Owe-1.Beneficiary, BANK
	BENEFICIARY	Sem	Chapter.Eleven.Agent, CORPORATION
	THEME	Sem	(>0)
		MEAS-UNIT	MONETARY-UNIT

<u>Chapter-Eleven.Preconditions</u>

RETAIN-1			
	AGENT	Sem	Chapter-Eleven.Agent.Corporation
	THEME	Sem	ATTORNEY

<u>Chapter-Eleven.Preconditions</u>

APPLY-FOR-1			
	AGENT	Sem	Chapter-Eleven.Agent, CORPORATION
		Relax-to	ATTORNEY
	BENEFICIARY	Sem	COURT
	THEME	Sem	Chapter-Eleven.Theme, Event
	PRECONDITIONS	Sem	Greater-Than-1
		Sem	RETAIN-1
	HAS-PARTS	Sem	INFORM-1
		Sem	PAY-1
	EFFECTS	Sem	DESIGNATE-1
		Default	POSTPONE-1
		Sem	MEETING-1
		Sem	LITIGATION-1

<u>Apply-For-1.Has-Parts</u>

INFORM-1			
AGENT	Sem	Chapter-Eleven.Agent, CORPORATION	
	Relax-to	ATTORNEY	
BENEFICIARY	Sem	Apply-For-1.Beneficiary, COURT	
THEME	Domain	INFORMATION	

Range	Chapter-Eleven.Agent.Address, Chapter-Eleven.Agent. Liabilities,Chapter-Eleven. Agent.Assets	

Apply-For-1.Has-Parts

PAY-1		
AGENT	Sem	Chapter-Eleven.Agent, CORPORATION
BENEFICIARY	Sem	Apply-For-1.Beneficiary, COURT
THEME	Sem	(<100000)
	Meas-Unit	MONETARY-UNIT
EFFECTS	Sem	DESIGNATE-1

These are much the same preconditions as for CHAPTER-SEVEN, with the exception that the Theme of CHAPTER-ELEVEN is REORGANIZATION or BUYOUT, while it is CANCEL-OBLIGATION for CHAPTER-SEVEN. Although they might be "factored out" and shown only in BANKRUPTCY, the preconditions are repeated here for convenience.

Chapter-Eleven.Has-Parts

PETITION-1		
AGENT	Sem	Owe-1.Beneficiary, BANK
THEME	Sem	Chapter-Eleven.Theme, EVENT
EFFECTS	Sem	APPLY-FOR-1, EVENT

Since a creditor may force a corporation to file for bankruptcy.

Chapter-Eleven.Has-Parts

DESIGNATE-1		
AGENT	Sem	Apply-For-1.Beneficiary, COURT
THEME	Sem	COURT-TRUSTEE
PRECONDITIONS	Sem	APPLY-FOR-1, EVENT
		PAY-1, EVENT
EFFECTS	Sem	LITIGATION-1, EVENT
	Sem	MEETING-1, EVENT

Chapter-Eleven.Has-Parts

POSTPONE-1

AGENT	Sem	Apply-For-1.Beneficiary, COURT
BENEFICIARY	Sem	Chapter-Eleven.Agent, OBJECT
THEME	Sem	Owe-1.Theme, EVENT

<u>Chapter-Eleven.Has-Parts</u>

LITIGATION-1

AGENT	Sem	Chapter-Eleven.Agent, HUMAN
	Relax-to	ATTORNEY
ACCOMPANIER	Sem	Owe-1.Beneficiary, HUMAN
	Relax-to	ATTORNEY
THEME	Sem	Chapter-Eleven.Theme, EVENT
LOCATION	Sem	COURTROOM, PLACE

<u>Chapter-Eleven.Has-Parts</u>

PROPOSE-1

AGENT	Sem	Chapter-Eleven.Agent, HUMAN
	Relax-to	ATTORNEY
BENEFICIARY	Sem	COURT-TRUSTEE, Owe-1.Beneficiary
THEME	Sem	REORGANIZATION-1, EVENT
PRECONDITIONS	Sem	APPLY-FOR-1, EVENT
EFFECTS	Sem	VOTE-1, EVENT

<u>Propose-1.Effects</u>

VOTE-1

AGENT	Sem	JUDGE
ACCOMPANIER	Sem	Owe-1.Beneficiary, HUMAN
THEME	Sem	PROPOSE-1, EVENT
EFFECTS	Optional	REJECT-1, EVENT
	Optional	ACCEPT-1, EVENT

<u>Vote-1.Effects</u>

REJECT-1

AGENT	Sem	JUDGE
ACCOMPANIER	Sem	Owe-1.Beneficiary, HUMAN
THEME	Sem	PROPOSE-1, EVENT
EFFECTS	Optional	PROPOSE-1, EVENT

<u>Vote-1.Effects</u>

ACCEPT-1

AGENT	Sem	JUDGE
ACCOMPANIER	Sem	Owe-1.Beneficiary, HUMAN
THEME	Sem	PROPOSE-1, EVENT
EFFECTS	Sem	REORGANIZATION-1, EVENT

That is, if the bankrupt corporation proposes a reorganization, then its creditors vote on whether to accept the proposal, as does the bankruptcy judge. If they both reject, then the bankrupt corporation may make another proposal; if they both accept, then the reorganization proceeds.

<u>Chapter-Eleven.Has-Parts</u>

PROPOSE-2

AGENT	Sem	CORPORATION (Not Chapter-Eleven.Agent)
BENEFICIARY	Sem	COURT-TRUSTEE, Owe-1. Beneficiary, Chapter-Eleven. Agent, HUMAN
PRECONDITIONS	Sem	APPLY-FOR-1, EVENT
THEME	Sem	BUYOUT-1, EVENT
EFFECTS	Sem	VOTE-2, EVENT

<u>Propose-2.Effects</u>

VOTE-2

AGENT	Sem	JUDGE
ACCOMPANIER	Sem	Owe-1.Beneficiary, HUMAN
THEME	Sem	Propose-2, EVENT
EFFECTS	Optional	REJECT-2, EVENT
	Optional	ACCEPT-2, EVENT

<u>Vote-2.Effects</u>

REJECT-2

AGENT	Sem	JUDGE, HUMAN
ACCOMPANIER	Sem	Owe-1.Beneficiary, HUMAN
THEME	Sem	PROPOSE-2, EVENT
EFFECTS	Optional	PROPOSE-2, EVENT

<u>Vote-2.Effects</u>

ACCEPT-2

AGENT	Sem	JUDGE
ACCOMPANIER	Sem	Owe-1.Beneficiary, HUMAN
THEME	Sem	PROPOSE-2, EVENT
EFFECTS	Sem	BUYOUT-1, EVENT

Or, another corporation may offer to buyout the bankrupt corporation. Again, the creditors vote on whether to accept the proposal or wait for another proposal.

Chapter-Eleven.Has-Parts

DECREASE			
	THEME	Sem	Bankruptcy.Agent.Stock
	INITIAL-VALUE	Sem	(>0)
		Meas-Unit	MONETARY-UNIT
	FINAL-VALUE	Default	(0)
		Meas-Unit	MONETARY-UNIT

The bankrupt corporation's stock value decreases, usually becoming worthless.

Chapter-Eleven.Has-Parts

REORGANIZATION-1			
	AGENT	Sem	Bankruptcy.Agent, HUMAN
	THEME	Sem	Bankruptcy.Agent, HUMAN
	PRECONDITIONS	Sem	Accept-1, EVENT
	HAS-PARTS	Optional	Buy-Object-1, EVENT
		Optional	Recapitalize-1, EVENT
		Optional	MARKETING-EVENT-1
		Optional	SALES-ACTIVITY-1
	EFFECTS	Sem	CANCEL-OBLIGATION

A bankrupt corporation may undergo reorganization of various means: it may sell off subsidiaries (BUY-OBJECT-1), recapitalize (RECAPITALIZE-1), create a new marketing plan (MARKETING-EVENT), or offer new/different merchandise (SALES-ACTIVITY-1).

Reorganization-1.Has-Parts

BUY-OBJECT-1			
	AGENT	Sem	CORPORATION (Not Chapter-Eleven. Agent), HUMAN
	THEME	Sem	Bankruptcy.Agent.Subsidiary, OBJECT

Reorganization-1.Has-Parts

RECAPITALIZE-1			
	AGENT	Sem	Chapter-Eleven.Agent, HUMAN
	THEME	Sem	Owe-1.Theme, EVENT

Reorganization-1.Has-Parts

MARKETING-EVENT-1

AGENT	Sem	Chapter-Eleven.Agent, HUMAN
THEME	Sem	MARKETING, EVENT

Reorganization-1.Has-Parts

SALES-ACTIVITY

AGENT	Sem	Chapter-Eleven.Agent, HUMAN
THEME	Sem	SALES-EVENT, EVENT

Chapter-Eleven.Has-Parts

BUYOUT-1

AGENT	Sem	CORPORATION (Not Chapter-Eleven. Agent), HUMAN
THEME	Sem	Chapter-Eleven.Agent, OBJECT
PRECONDITIONS	Sem	ACCEPT-2, EVENT
EFFECTS	Default	OWE-2, EVENT
	Sem	CANCEL-OBLIGATION-1

Buyout-1.Effects

OWE-2

AGENT	Sem	Buyout-1.Agent, HUMAN
THEME	Sem	Owe-1.Theme, EVENT
BENEFICIARY	Sem	Owe-1.Beneficiary, HUMAN

Buyout-1.Effects

CANCEL-OBLIGATION-1

AGENT	Sem	COURT, HUMAN
THEME	Sem	OWE-1, EVENT
BENEFICIARY	Sem	Chapter-Eleven.Agent, HUMAN

Chapter-Eleven.Effects

APPLY-FOR-2

AGENT	Sem	Chapter-Eleven.Agent, CORPORATION
THEME	Sem	Chapter-Seven, EVENT
PRECONDITIONS	Optional	EVENT-1 (Not ACCEPT-1 and Not ACCEPT-2)
	Default	DEMAND-1, EVENT

Apply-For-2.Preconditions

DEMAND-1

AGENT	Sem	COURT-TRUSTEE, COURT
THEME	Sem	APPLY-FOR-2, EVENT

That is, in the event that no proposals for a buyout or reorganization have been accepted by both the creditors and the judge (within the given time frame), then the judge may order the bankrupt corporation to re-file for Chapter 7 bankruptcy, in which its assets will be sold off to pay creditors. Thus, it may in fact be the case that the CHAPTER-SEVEN complex event is nested within the CHAPTER-ELEVEN complex event.

Chapter-Eleven.Effects

EVENT-1			
	AGENT	Sem	Chapter-Eleven.Agent, CORPORATION
	THEME	Sem	EVENT (Not BANKRUPTCY)
	PRECONDITIONS	Optional	REORGANIZATION-1
		Optional	BUYOUT-1

That is, one of the effects of emerging from a bankruptcy is that the corporation is not allowed to file for bankruptcy again (for a period of years).

4.6 Conclusion

In this chapter, several complex events have been shown in order to demonstrate the feasibility of the formalism developed in this dissertation. Especially given that earlier systems have been criticized for not providing fine-grained-enough representations, this chapter reports the development of several complex events very close in conceptual nature: BUY-SERVICE, BUY-OBJECT, RENT, LEND, and AUCTION. The complex-event knowledge developed for each of these complex events discerns between each real-world event, as evidenced by their differing subevents and differing contingencies between these subevents. Thus, it is argued that complex events developed for an ontological-semantic natural language processing system can provide representations sensitive enough for language understanding.

Furthermore, the development of subevents of complex events as complex events themselves demonstrates the ability of an ontological-semantic NLP system to provide the conceptual linking necessary for robust meaning representation. Such linking is the hallmark of the frame-based and semantic-network-based architectures. The formalism developed in this dissertation fits well in the already-implemented Mikrokosmos system.

Finally, the development of the very complex event, BANKRUPTCY, demonstrates that a complex-event knowledge base for an ontological-semantic NLP system is capable of providing near-limitless depth of representation. Moreover, the BANKRUPTCY event provides evidence that the formalism is capable of reprenting specialized, expert knowledge, thereby extending the its usefulness.

Chapter Five provides a summary of the dissertation and suggests future directions for research into complex-event knowledge representation.

CHAPTER FIVE:

CONCLUSION

The use of natural language is believed to exhibit clear preferences for economy. Examples include: the "economy principle" (cf. Chomsky 1995) and verb-deletion in sytactic theory; the conversational maxims (cf. Grice 1975), in pragmatics, that direct speakers to "say only what is necessary"; syllable reduction (e.g., "did you eat yet?" > "jeet yet?"); anaphoric relations; and, metonymy (e.g., "She washed her car," not "She washed *the outside of* her car"). Such ellipses, or lacunae, on the part of a speaker are generally licensed as long as the hearer has the appropriate knowledge to fill in these gaps. As Schank and Abelson (1977) and others have noted, these gaps are present at the discourse-level—as "implied causal chains" or "pragmatic presuppositions" (cf. Levinson 1983)—and are recoverable based on a hearer's knowledge of the world.

But whereas the correspondence between, say, a phonologically reduced syllable and its "full" counterpart, or a pronoun and its antecedent, are rather well-defined and limited in scope, a text may draw on any aspect of a hearer's knowledge of the world. If I say that a toddler "looks like a linebacker," a hearer will know that I refer to the child's stocky build, with perhaps a short neck. If I say that I tried to use a wicker basket for a stool, a hearer will expect that I fell. We know also many things about how people act toward one another, how they achieve goals, what events are likely to follow, or to have occurred already, in a given situation. Any of these pieces of knowledge may be used to recover ellipsis in a text. In fact, we often add to our own world knowledge by generating abductive inferences about what the world must be like in order lead to the given result in a text—even when the text is playful or joking (cf. McDonough 1997). But while humans show impressive ability to draw the necessary correspondences between the text and the world referred to by the text, a computer system without knowledge of the world is unable to constrain the possible correspondences, inevitably leading to an "inferential explosion."

In the fields of artificial intelligence and natural language processing (NLP), Bar-Hillel (1960) and Hayes (1969) were among the first to recognize that a computer system must be equipped with knowledge of the physical world in order to constrain the possibilities in order to understand real, live natural language texts. Schank (Schank and Abelson 1977) and his colleagues (e.g., Cullingford 1978; Wilensky 1980) further asserted that an NLP system must know something about stereotypical event sequences ("scripts") and the things people are likely to do in given situations.

There is little doubt, in the field of NLP, that to approximate a human's ability to understand a text, a computer system will need to be programmed with knowledge of complex events—the events that are likely to occur or to have occurred given the mention of some other event. The goal of this dissertation is to develop a formalism in which large-scale acquisition of this knowledge can proceed.

5.1 Summary

Representing the entirety of our knowledge of complex events is, to be sure, an immense task—so immense that it is difficult even to hazard a guess. Thus, the complex-event knowledge described in this dissertation has not yet been implemented, nor tested. This is, instead, a feasibility study of sorts. Earlier programs designed to represent complex-event knowledge, although successful in demonstrating their ability to handle real texts, were built on a smaller scale. A review of the literature on these systems suggests that their limitations were three-fold: (i) meanings reprentations were not fine-grained enough to trigger complex-event inferences, (ii) complex-event knowledge was developed to be conceptually disparate and therefore likely to fire in contexts it should not, and (iii) complex-event knowledge was represented in modular chunks, making it difficult to facilitate inferencing relationships between events and difficult to maintain a large knowledge base.

An ontological-semantic natural language processing system, such as the Mikrokosmos system, is able to meet these requirements: meaning representation for this system has demonstrated 97% accuracy. Because the formalism for representing complex-event knowledge developed in this dissertation fits within the ontological-semantic system already in place, this accuracy should not be affected. Furthermore, the ontology of conceptual entities provides a framework for distinguishing fine-grained differences among these conceptual entities. Complex-event knowledge takes advantage of this by augmenting conceptual knowledge already in place, thus serving as a built-in heuristic to ensure that all complex-event knowledge is activated only in the appropriate contexts. Further evidence of the adequacy the ontology to represent complex-event knowledge is that each of the complex events developed in this dissertation makes use only of EVENT concepts already available in the ontology.

Linking between the frames in the ontology also allows conceptual knowledge to be shared and provides a path to generate inferences between two (or more) concepts. For example, since a "Pay" event may occur in numerous contexts, it makes sense to represent this EVENT concept just once and then provide the concept with links to other concepts of which it may play a part—e.g., "Buy," "Lend," and "Bankruptcy." The formalism developed in this dissertation allows complex events to comprise other complex events, by making use of conceptual linking built into the hierarchy.

While the ontological-semantic framework is well-suited to the needs of complex-event knowledge representation, some changes to this framework have been proposed in this dissertation in order to facilitate implementation. First, the competing needs to represent contingencies specific to subevents within a complex event and to maintain enough generality to represent information in a subsumption hierarchy gave rise to the variable-binding problem, for which a "special slot" BINDING-ROLE has been proposed. The BINDING-ROLE slot links a filler of a slot in its frame to a filler of a slot linked to that frame. This, for example, allows the PAY subevent, of BUY-OBJECT, not simply to refer to any human (as is the case in PAY's conceptual frame), but rather only to the AGENT of the BUY-OBJECT event.

Second, an Optional facet has been proposed to mark subevents that would not be assumed by the system, but would be available for assertion given confirming evidence. The subevents marked with the Optional facet thus serve the same role as "script deviations" in Schank and Abelson (1977). The usefulness of such information is obvious: in any complex event, certain subevents are more likely than others. Arming the computer system with likely—but not criterial—information increases its ability to draw inferences about the text. Furthermore, by specifying that slots marked with the Optional facet, like the Default facet, are not inherited, we avoid the the undesirable outcome that the filler parent concept slot marked with the Optional facet might be an excluded excluded filler (or slot) in a child concept. Although there are obvious advantages motivating the inclusion of the Optional facet, this marks a rather radical altering of the ontological principles on which the Mikrokosmos ontology is based. The ontology is a very strict "And-Graph" in which frames are conjunctions of their properties and in which a child frame is a conjunction of the slots of its parent; only occasional use is made of the (uninherited) Default slot and the *nothing* filler, which blocks inheritance of a slot. It will, of course, take implementation of the Optional facet to determine its effect on the system. Taking the suggestion in Mahesh (1996), this dissertation also recommends the specification of a SUBEVENT-OF slot, in which to provide information about which complex events a given EVENT concept is a component. However, it has also been suggested in this dissertation that adding this slot should wait until numerous complex events have been implemented, since the SUBEVENT-OF might bias the system toward favoring the complex events listed in this slot. Finally, although the PRECONDITIONS, HAS-PARTS, and EFFECTS slots are currently specified for the ontology, this dissertation provides them with new scope and provides clear principles for their use in EVENT concept frames. Furthermore, it has been demonstrated that these three slots may play vital roles in structuring contingencies between subevents in complex-event frames.

Several minor revisions to the ontological hierarchy have also been suggested in this dissertation—chief of these being the suggestion that some leaf-node concepts be further broken down into two or more child concepts. This seems to be an inevitable part of ontology development: first, the ontology's primary goal is to partition conceptual knowledge into manageable chunks—further refinements are a part of this process; and second, the amount of detail necessary to represent complex-event knowledge is somewhat at odds with the general, conceptual classifications for which the ontology is designed. However, it is also the case that most of the complex events developed thus far do not require that current concepts be further divided. Finally, throughout this dissertation several OBJECT frames and slots for already-existing concepts have been proposed. The need for making these small, local changes has arisen not because of the formalism for representing complex-events, but is rather the norm for all phases of knowledge acquisition.

In summary, a complex-event knowledge formalism, to be viable for large-scale acquisition, must meet three criteria:

- it must adequately encode information for the real-world events to which it refers;
- it must discern between similar conceptual entities; and,
- it must minimize duplication of knowledge.

The complex events developed in Chapter Four demonstrate that an ontological-semantic NLP system can meet these criteria. Each complex event is able to represent the contingencies of the real-world events to which they refer, including criteriality, optionality, conditionality, and branching. Furthermore, the LEND, RENT, and AUCTION complex events provide evidence that the formalism developed in this dissertation encodes minute differences between conceptual entities. And, given that only EVENT concepts already existing in the ontological hierarchy have been asserted as subevents and given that complex events themselves can be linked, as subevents, to other complex events, this formalism minimizes the duplication of knowledge that would cripple any attempt at large-scale complex-event knowledge acquisition.

5.2 Future work

While the goal of this dissertation is to develop principles for representing complex-event knowledge in an ontological-semantic NLP system, much needs to be done before the system can take advantage of the work here. The next steps are to implement complex-event knowledge in the Mikrokosmos system and then to assess its performance. Because this knowledge has been developed within the guidelines of the ontology, augmenting the frames should not present much of a problem. However, it is quite possible that inferencing mechanisms currently in use in the system will need to be modified in order to handle variable-binding between subevents and to handle the Optional subevents.

Once preliminary testing has determined the extent to which the inferencing mechanisms must be modified, large-scale acquisition of complex-event knowledge can begin. Although this dissertation makes clear that there is no clear preference for the order in which this knowledge is acquired, the Mikrokosmos ontology was originally designed to handle texts concerning corporate affairs. Thus, developing complex events that also address corporate affairs might help quickly, and firmly, solidify the reputation of the Mikrokosmos system as an exceptionally high-quality language understanding system.

However, as was discovered during the development of the Mikrokosmos system, even when one has a focus domain, the amount of knowledge needed to understand texts about this domain is great, since, of course, real-world texts do not strictly adhere to the subdomain. Thus, it is quite possible that, even if complex-event knowledge acquisition focuses on corporate affairs, perhaps two thousand complex events will need to be developed. Based on this dissertation, acquisition on this order of magnitude may still take less than a year.

LIST OF REFERENCES

LIST OF REFERENCES

Allen, J.F. (1984). "Towards a general theory of action and time." *Artificial intelligence*, 23, 123-154.

Allen, J.F. (1995). *Natural language understanding*. Redwood City, CA: Benjamin/Cummings.

Austin, J. (1975 [1962]). *How to do things with words*. 2nd ed. Cambridge, MA: Harvard UP.

Bar-Hillel, Y. (1960). "The present status of automatic translation of languages." In: *Advances in computers*, vol.1, F.L. Alt (ed.). New York: Academic Press.

Boguraev, B.K. (1989). "The definitional power of words." In: *Theoretical issues in natural language processing*, Y. Wilks (ed.). Hillsdale, NJ: Lawrence Erlbaum Associates.

Brachman, R.J. and Levesque, H.J. (eds.) (1985). *Readings in knowledge representation*. Los Altos, CA: Morgan Kaufmann.

Carbonell, J. (1981). "POLITICS." In: R.C. Schank and C.K. Riesbeck (eds.).

Carlson, L. and Nirenburg, S. (1990). "World knowledge in natural language processing." Center for Machine Translation, Carnegie Mellon University, Pittsburgh, PA. Technical Report 90-121.

Charniak, E. (1977). "Ms. Malaprop, a language comprehension program." *Proceedings of IJCAI-77.*

Charniak, E. (1979). "Ms. Malaprop, a language comprehension program." In: D. Metzing (ed.).

Charniak, E. (1986). "Jack and Jane in search of a theory of knowledge." In: B.J. Grosz, K.S. Jones, and B.L. Webber (eds.).

Chomsky, N. (1995). *The minimalist program*. Cambridge, MA: MIT Press.

Cullingford, R.E. (1978). Script application: computer understanding of newspaper stories. Ph.D. dissertation. Yale University, New Haven, CT.

Cullingford, R.E. (1981). "SAM." In: R.C. Schank and C.K. Riesbeck (eds.).

DeJong, G. (1979). Skimming newspaper stories in real time. Ph.D. dissertation. Technical Report 158, Department of Computer Science, Yale University, New Haven, CT.

DeJong, G. (1982). "An overview of the FRUMP system." In: *Strategies for natural language processing*, W.G. Lehnert and M.H. Ringle (eds.). Hillsdale, NJ: Lawrence Erlbaum Associates.

Dreyfus, H.L. (1985). "From micro-worlds to knowledge representation: AI at an impasse." In: R.J. Brachman and H.J. Levesque (eds.)

Fikes, R., Farquhar, A., and Rice, J. (1997). "Tools for assembling modular ontologies in Ontolingua." In: *Proceedings of AAAI*. Cambridge, MA.

Golden, R.M. and Rumelhart, D.E. (1993). "A parallel distributed processing model of story comprehension and recall." *Discourse processes*, 16, 203-237.

Grice, H.P. (1975). "Logic and conversation." In: *Speech acts*, P.Cole and J.L. Morgan (eds.). New York: Academic Press.

Grosz, B.J., Jones, K.S., and Webber, B.L. (1986). *Readings in natural language processing*. Los Altos, CA: Morgan Kaufman.

Hayes, P. (1979). "The logic of frames." In: D. Metzing (ed.).

Hayes, P. (1985). "The second naïve physics manifesto." In: *Formal theories of the commonsense world*, J.R. Hobbs and R.C. Moore (eds.). Norwood, NJ: Ablex.

Holland, J.H., Holyoak, K.J., Nisbett, R.E., and Thagard, P.R. (1986). *Induction*. Cambridge, MA: MIT Press.

Hudson, J.A., Fivush, R., and Kuebli, J. (1992). "Scripts and episodes: the development of event memory." *Applied cognitive psychology*, 6, 483-505.

Katz, J.J. and Fodor, J.A. (1963). "The structure of a semantic theory." *Language*, 39, 170-210.

Kempson, R. (1997). "Semantics, pragmatics, and natural-language interpretation." In: *The handbook of contemporary semantic theory*, S. Lappin (ed.). Oxford: Blackwell.

Kittredge, R.I. (1987). "The significance of sublanguage for automatic translation." In: *Machine translation: theoretical and methodological issues*, S. Nirenburg (ed.). Cambridge: Cambridge UP.

Lakoff, G. (1987). *Women, fire, and dangerous things*. Chicago: University of Chicago Press.

Lenat, D.B. and Guha, R.V. (1990). *Building large knowledge-based systems*. Reading, MA: Addison-Wesley.

Levinson, S. (1983). *Pragmatics*. Cambridge: Cambridge UP.

Mahesh, K. (1996). "Ontology development for machine translation: Ideology and methodology." Technical report MCCS-96-292, Computing Research Laboratory, New Mexico State University.

Mahesh, K. and Nirenburg, S. (1995). "A situated ontology for practical NLP." In: *Proceedings of the workshop on basic ontological issues in knowledge sharing*, IJCAI-95, Montreal, Canada.

McDonough, C.J. (1997). The information-conveying ability of humorous texts: an analysis of 'semantic objects.' M.A. thesis. Youngstown State University.

McKoon, G., Ratcliff, R., and Seifert, C. (1989). "Making the connection: generalized knowledge structures in story understanding." *Journal of memory and language*, 28, 711-734.

Metzing, D. (ed.) (1979). *Frame conceptions and text understanding*. Berlin: Walter de Gruyter.

Mikkulainen, R. (1993). *Subsymbolic natural language processing*. Cambridge, MA: MIT Press.

Mikkulainen, R. (1995). "Script-based inference and memory retrieval in subsymbolic story processing." *Applied intelligence*, 5, 137-163.

Minsky, M. (1968). "Introduction." In: *Semantic information processing*, M. Minsky (ed.). Cambridge, MA: MIT Press.

Minsky, M. (1975). "A framework for representing knowledge." In: *The psychology of computer vision*, P.H. Winston (ed.). New York: McGraw-Hill.

Nagao, M. (1990). *Knowledge and inference*. Boston: Academic Press.

Nirenburg, S. (1986). "Linguistics and artificial intelligence." In: *The real-world linguist: linguistic applications in the 1980s*, P. Bjarkman and V. Raskin (eds.),. Norwood, NJ: Ablex.

Nirenburg, S. and Raskin, V. (To appear). *Principles of ontological semantics*. Cambridge: Cambridge UP.

Nirenburg, S., Raskin, V., and Onyshkevych, B. (1995). "Apologiae ontologiae." In: *Proceedings of the conference on theoretical and methodological issues in machine translation*. Leuven, Belgium.

Onyshkevych, B. and Nirenburg, S. (1995). "A lexicon for knowledge-based MT." *Machine translation*, 10, 1-2.

Pereira, F.C.N. and Grosz, B.J. (eds.) (1994). *Natural language processing*. Cambridge, MA: MIT Press.

Raskin, V. (1971). K teorii yazykovykh podistem ("Towards a theory of sublanguages"). Moscow, USSR: Moscow State UP.

Raskin, V. (1985). "Linguistic and encyclopedic knowledge in text processing." *Quaderni di semantica* 6, pp. 92-102

Riesbeck, C. (1975). "Conceptual analysis." In: *Conceptual information processing*, R.C. Schank (ed.). Amsterdam: North Holland.

Rumelhart, D. (1994). "Towards an interactive model of reading." In: *Theoretical models of reading*, R.B. Ruddell, M.R. Ruddell, and H. Singer (eds.). Newark, DE: International Reading Association.

Schank, R.C. (1972). "Conceptual Dependency: a theory of natural language understanding." *Cognitive psychology*, 3, 552-631.

Schank, R.C. (1975). *Conceptual information processing*. Amsterdam: North Holland.

Schank, R.C. (1980). "Language and memory." *Cognitive science*, 4, 243-284.

Schank, R.C. (1983). *Dynamic memory*. Cambridge: Cambridge UP.

Schank, R.C. (1999). *Dynamic memory revisited*. Cambridge: Cambridge UP.

Schank, R.C. and Abelson, R. (1977). *Scripts, plans, goals and understanding*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Schank, R.C. and Riesbeck, C.K. (eds.) (1981). *Inside computer understanding*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Searle, J. (1972). "What is a speech act?" In: *Language and social context*, P.P. Giglioli (ed.). London: Penguin Books.

Viegas, E. (1999). "An overt semantics with a machine-guided approach for robust LKBs." In: *Proceedings of SIGLEX99, Standardizing Lexical Resources*, ACL99, University of Maryland.

Viegas, E., Mahesh, K., Nirenburg, S. and Beale, S. (1999). "Semantics in action." In: *Predicative forms in natural language and in lexical knowledge bases*, P. Saint-Dizier (ed). Dordrecht: Kluwer.

Wilensky, R. (1980). *Understanding goal-based stories*. New York: Garland. [Reprint of 1978 Ph.D. dissertation, Yale University, New Haven, CT.]

Wilks, Y. (1986). "An intelligent analyzer and understander of English." In: B.J. Grosz, K.S. Jones, and B.L. Webber (eds.).

Winograd, T. (1972). "Understanding natural language." *Cognitive psychology*, 3.

Winograd, T. (1976). *Understanding natural language*. New York: Academic Press. [Reprint of Winograd 1972.]

VITA

VITA

Craig J. McDonough was born in 1970 in York, Pennsylvania, and grew up in Boardman, Ohio, where he graduated from Boardman High School in 1989. After studying for a year at the University of Pittsburgh, he finished his undergraduate work, with a major in English and a minor in German, at Denison University in Ohio, in 1994.

As an M.A. student at Youngstown State University, he taught first-year composition and won the 1997 Dean's Quest Award for best presentation of graduate research, while studying under Dr. Salvatore Attardo, in the areas of semantics, pragmatics, and humor studies.

McDonough came to Purdue University in 1997 and, while there, taught first-year composition, ESL composition, and syntax and semantics. In addition to his teaching assistantship, he also served as a research assistant under Drs. Atallah and Raskin, in the areas of natural language processing and information security.

He is married to Wendy McDonough; they have two young and adorable daughters, Madeleine and Meghan.