[Proceedings of the *Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Colgate University, Hamilton, New York, August 14-16, 1985]

Interlingua Design for TRANSLATOR

Sergei Nirenburg, Victor Raskin** and Allen B. Tucker**

*Department of Computer Science, Colgate University
**Department of English, Purdue University

ABSTRACT

The interlingua approach to machine translation (MT) is characterized by the following two stages: 1) translation of the source text into an intermediate representation, an artificial language *(interlingua)* which is designed to capture the various types of meaning of the source text and 2) translation from the interlingua into the target text. Over the years a number of MT projects tried to develop interlingua-based systems. In these projects the amount of linguistic and encyclopaedic knowledge used to produce intermediate representations was quite limited. However, even at that level difficulties connected with encoding knowledge seemed overwhelming. The TRANSLATOR project at Colgate University benefits from recent developments in knowledge representation techniques. The text of its interlingua text reflects syntactic, lexical, contextual, discourse (including speech situation) and pragmatic meaning of the input This paper discusses the lexicon and grammar of the interlingua used in TRANSLATOR, and touches upon the structure of the bilingual (source language to interlingua) dictionaries. The actual compilation of the interlingua dictionary and additional knowledge bases is an empirical process during which modifications to the original formulations are expected to occur. At all times in the design process the authors were guided by the desire to make decisions that are 'literate' from the point of view of linguistic theory and the experience of knowledge representation in artificial intelligence.

1. Introduction.

MT projects based on the interlingua (IL) approach perceive the translation process as consisting of two major stages: translation between source language (SL) and an interlingua; and translation between interlingua and target language (TL). Across the IL projects there has been no agreement as to the nature of IL. IL-based projects differed in the desired amount and nature of information contained in IL as well as the number of actual translation steps involved in the IL-oriented translation.

We understand the IL-based translation as, roughly, the following sequence of steps:

- apply the parsing group of programs to an SL text producing an IL text, a network of interconnected IL text, sentence, clause, object and event frame structures; this stage requires as a background a compendium of world knowledge (in the form of a monolingual IL dictionary), grammars of SL and IL, and an SL -- IL dictionary;
- apply the inspector/augmentor program to the IL representation received; the goal is to resolve anaphoric phenomena and, generally, to ensure that ail compulsory slots in the IL text frames are filled; the background for this is the monolingual IL dictionary; the result of this stage is an improved IL text; the inspector makes decisions concerning the extent of inference making in the system; the inspector can be considered TRANSLATOR's expert system, since it is planned to incorporate into it knowledge of human translators about their craft; the inspector also serves as troubleshooter by attempting to recover from unexpected difficulties; in order to be able to do this, the inspector draws on an additional knowledge base that includes, for instance,

knowledge about the translation process itself and situation knowledge which has been traditionally encoded in if-then rules, e.g. *IF it rains AND IF A is outside AND IF (A does not wear waterproof clothes OR A is under an umbrella) THEN A gets wet;*

- apply the generator group of programs to the IL text to produce a TL text; the background knowledge for this operation includes an IL -- TL dictionary and a grammar for TL†.

The background of the TRANSLATOR MT project at Colgate is presented in Tucker and Nirenburg (1984). The knowledge clusters of TRANSLATOR are the focus of this paper. They are identified as follows:

- IL dictionary
- SL - IL dictionary
- IL - TL dictionary
- SL grammar
- SL - IL translator (a group of parsers)
- IL grammar
- IL inspector/augmentor: TRANSLATOR'S expert system
- Inspector knowledge base
- TL grammar
- IL - TL translator

In this paper we offer a proposal for the types and organization of linguistic and world knowledge contained in the dictionary and the grammar of IL and used for producing IL texts. We also present a glimpse into the structure of the SL - IL bilingual dictionary (for reasons of clarity, English is chosen as the SL in this example). A sampling of the contents of an IL - TL dictionary will be reported elsewhere.

We do not deal here with the *processes* of analysis (parsing) and generation that connect the three texts involved in translation. Nor do we describe in any detail the system's augmentor/inspector, with its specialized knowledge.

A number of strategic decisions must be made in building an IL. First, the types of information to be included in IL should be chosen, so that it is sufficiently rich for achieving translation. It may be even more difficult to make sure that all types of information are actually necessary for that end. Second, the semantics of the IL frames and slots must be determined. We will define it extensionally by enumeration of value ranges for all frame and slot types suggested. Third, the choice of IL frame-owners (objects and events) is an important empirical question, connected with the problem of their semantic interpretation. The nature of our application (MT) suggests an approach to the solution of the last problem that is unavailable to conceptual universe builders in other applications. We maintain that the meaning of an IL entity is the set of its mappings to SL and TL correlates (the contents of the bilingual dictionaries in our system). By choosing this approach we avoid the pitfalls of compositional analysis of meaning, which tends to become an enormous burden in terms of effort and philosophical defensibility for anyone who attempts the construction of a complete NLP system.

We do not claim that the particular data set we include in IL is sufficient for translation. There must be a distinction between the ideal IL and the versions that can be immediately implemented. The evidence as to what is necessary and sufficient knowledge for MT will be found empirically. At this stage of the project we strive to develop what we conceive as an 'ideal' IL. Of course, our conception may prove inadequate. Problems in *achieving* this goal (typically, problems with semantic

† It is an open question whether the SL and TL grammars should be specialized (the former attuned to parsing; the latter, to generation) so that there will be a need to write two grammars for each language. This is the sphere of tradeoffs between generality (and parsimony of description) and the convenience of being able to tailor the design of a grammar to its particular application. Considerations of time efficiency of research effort as well as of the software system itself will also be taken into account.

representation, parsing and inferencing) will be highlighted in the implementation. We understand that in implementing TRANSLATOR we will inevitably have to resort to *ad hoc* measures to take care of certain 'real' text peculiarities (e.g., abbreviations, tables and equations, etc.). Also, as with all approaches to MT, a critical influence will be exerted by the *quality* of dictionaries and, to a lesser extent, grammars, irrespective of their theoretical basis.

It is important to keep in mind the difference between several kinds of potential inadequacies and obstacles. First, our IL may prove inadequate in some of its premises. Second, it may be argued that the interlingua approach *as such* is inadequate. Third, it may be argued that the IL approach is impractical because the difficulties in designing and implementing it may prove insurmountable. Incidentally, we believe that if the last claim can be proved to be true, this will simultaneously prove that MT is unattainable in any interesting sense.

A major problem at this stage of the project is to suggest what it means to 'understand' text in TRANSLATOR. The dominant approach to story understanding in the AI community is goal tracking. It has been developed mostly within the Yale school (cf. Schank & Abelson, 1977, Carbonell, 1979, Wilensky, 1983, etc.)

It is indeed very advantageous for input disambiguation to understand what *goals* the agent of an event pursues by causing this event to happen. Of special importance is the possibility to claim that once user goals are detected, the meaning of the text has been extracted, and, therefore, no further processing (inferencing) is necessary. This device kills two birds with one stone: first, it allows a definition of text meaning ('a network of goal and plan instances as pursued by all the protagonists in a story'); second, it provides the exit condition for the inference mechanism, which otherwise does not have a clearly cut stopping point.

Unfortunately, in MT one deals with texts that are typically not narratives (stories with protagonists that are causal agents). Among the most widely translated texts are textbooks, technical manuals, patents and other non-narrative literature. In texts such as the above one can hope to extract only goals of the participants in discourse situations (most often, the author of the text and the reader(s)), and not the protagonists of the story being told. Therefore, a different criterion for claiming 'understanding' of a text must be developed for MT†. Since a large number of text types does not lend itself directly and clearly to the goal-directed approach, we have to settle for the understanding criterion that requires the detection of the following types of knowledge:

- the conceptual context (the 'subworld') of the text; the use of this parameter for disambiguation is hardly novel; it goes back to Katz and Fodor (1963) and can be traced further back to the semantic field theory;
- the discourse content of the text; knowledge about logical, rhetorical, stylistic and other connections between propositions that constitute the 'body' of the text; elucidation of these connections allows much more freedom in the eventual expression of the same (both propositional and discourse) meaning in a different language, since it establishes equivalency relationships between, say, noun phrases and clauses, thus facilitating paraphrases with much less regard to the boundaries of the sentence and the clause than it was conceivable previously.

2. The structure of IL.

IL consists of the following types of entities:

- IL words: entity (object, event, process, state, etc.) types, described in the IL dictionary
- IL clauses: propositions (process and state tokens with their actants filled)
- IL sentences: utterances (propositions arranged with respect to the speech situation and constituting a quantum of discourse)

† Now it becomes quite clear why the stories understood by the goal-extraction school have always been 'action'-type narratives: eating at a restaurant; (newspaper reports of) terrorist activities; political arguments about (actual and projected) actions of certain countries and groups; reports about accidents, etc.

- IL texts: utterance clusters of paragraph length and more (utterances glued together by cohesion forces).

IL words belong to the IL dictionary. IL clauses, sentences and texts embody IL syntax. We therefore decided to use the term 'IL grammar' for the knowledge component where they are described.

The dictionary contains knowledge about the *types* of entities that IL deals with. It consists of a collection of frames that intuitively represent our generalized knowledge about the meaning of a conceptual entity. Most of the slots in the frames are instance variables, to be filled with a specific value when a token of the given type is instantiated (that is, in IL text). For example, the contents of the slot 'object' in the dictionary entry for the IL word *permit* is 'event', while in the IL translation of (1) the contents of the 'object' slot of the instance of *permit* will be C_1 which stands for the IL representation of the clause *John leaves,*

(1) John was allowed to leave.

In other words, the content of frame slots in the dictionary is not unlike the specification of the type of a variable among the declarations in a programming language. It is used for the similar purpose of validity checks. It is as if the word 'typical' were prepended to the slot names in the dictionary: 'typical-agent' instead of 'agent'.

The IL grammar describes the way in which actual IL texts are built. The latter consist of entity *tokens*. Propositions are represented in clause frames whose slot fillers are typically tokens of event object types†. An IL text reflects the knowledge as extracted from SL text and augmented, through inferencing, to reflect further disambiguation of input. While the dictionary is permanently stored in the system, the IL text is built by the SL - IL translator and the inspector/augmentor.

2.1. IL Dictionary.

The entities in the world of TRANSLATOR are organized in a multi-hierarchy* two fragments of which are illustrated in Figures 1 and 2. This structure serves as the IL monolingual dictionary. This shell serves as the starting point of compiling the IL dictionary. New concepts, often with additional slot types as well as different slot fillers are added as the work on the dictionary progresses. Several frame types of IL entities are illustrated below. A fragment of the IL dictionary see in Appendix 1.

The choice of the actual sets of values for the slot fillers in IL dictionary frame types (what can be called the IL data types) is one of the major problems referred to above. In what follows we describe the solutions suggested for TRANSLATOR. First, we present *generic* frames for the three top-level types of entities in TRANSLATOR's world and then comment on their slot contents. The slots in these frames stand for the types of information we invariably seek to know about such entities. Additional slots appear in frames that are descendants of the ones illustrated in the world quasi-hierarchy**.

† It is, of course, quite possible for a clause frame slot to be occupied by a reference to another clause (for example, in the cases of sentential subjects or objects).

* We do not guarantee a single parent to a node; although we organize the world according to the 'main' hierarchy of the isa relationship, we also include such orthogonal relationships as **made-of, part-of,** etc.

** It is an inherent property of a **physical-object** to have *color, size, shape* and *texture. But such slots will be inapplicable in a frame for* **object,** which, incidentally, is the contents of the 'isa' slot of the frame for **physical-object** in the system's world.

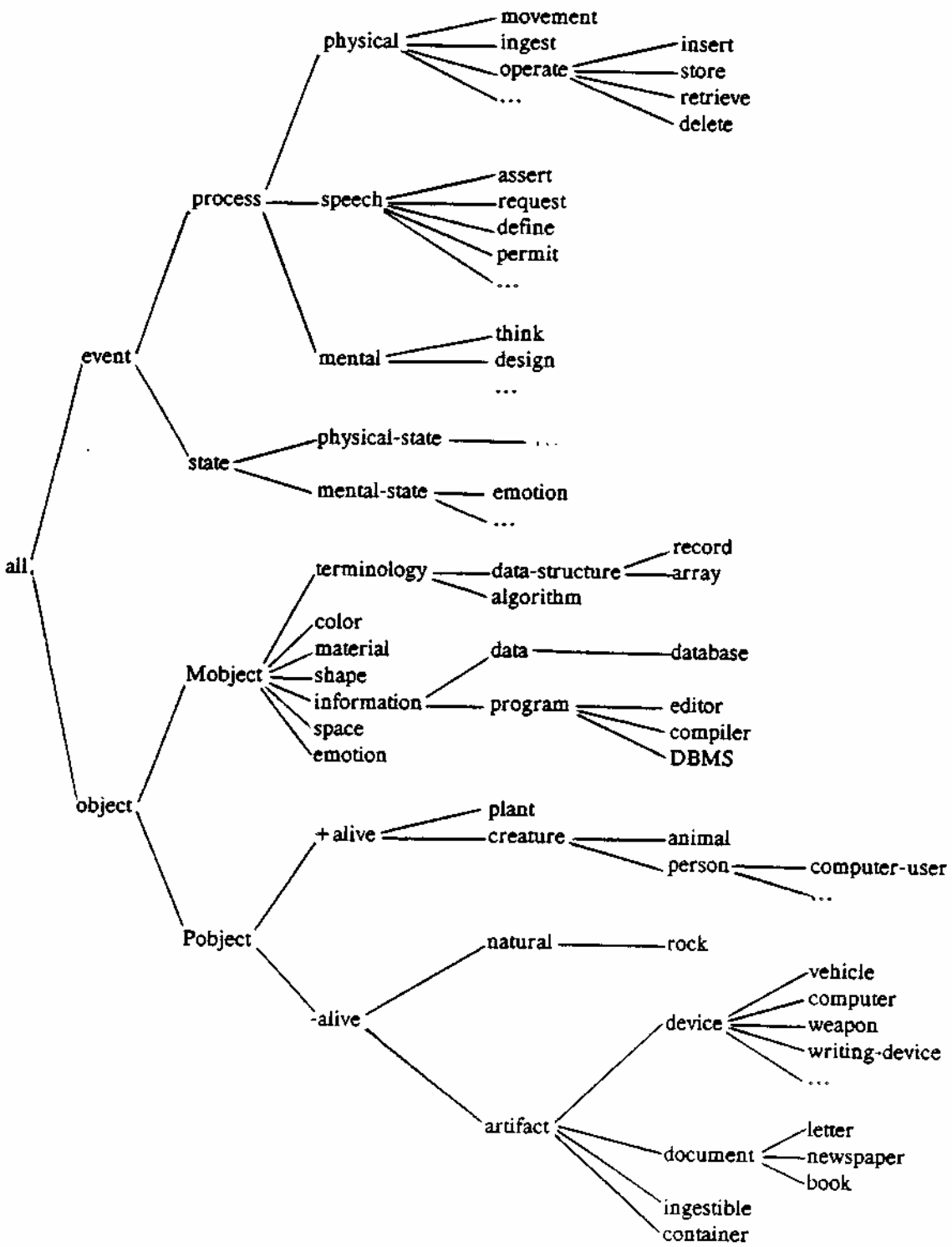

Figure 1. A fragment of the isa network

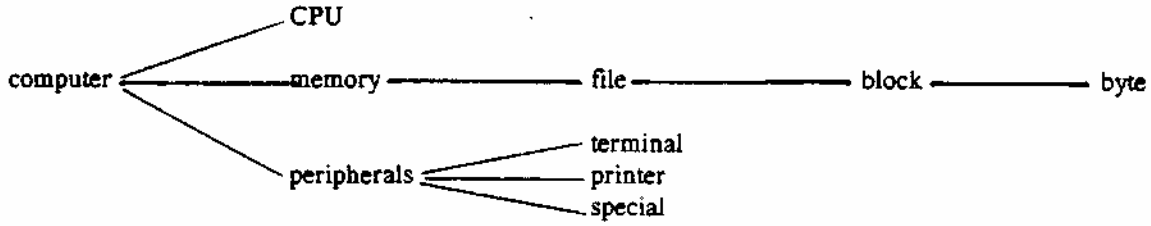

Figure 2. A fragment of the part-of network.

1. The object frame.

```
( < object-name >
  (isa < entity >)
 (subworlds < subworld-type > *)
 (agent-of < event-type >*)
 (object-of < event-type >*)
 (patient-of < event-type >*)
 (instrument-of < event-type > *)
 (source-of < event-type > *)
 (goal-of < event-type >*)
 (consists-of < object-type >*)
 (part-of < object-type >*)
 (described-by < clause >*)
 )
```

2. The state frame

```
(< state-name >
  (isa < entity >)
  (subworlds < subworld-type > *)
  (patient < object-type >*)
  (instrument < object-type >*)
  (time < time-marker >)
  (space < space-marker >)
  (preconditions < state-name > *)
  (effects < state-name >*)
  (described-by < clause >*) )
```

3. The process frame.

```
(< process-name >
  (isa < entity >)
  (subworlds < subworld-type >*)
  (agent < object-type >*)
  (object < object-type >*)
  (patient < object-type >*)
  (instrument < object-type >*)
  (source < object-type >*)
  (goal < object-type >*)
  (time < time-marker >)
  (space < space-marker >)
```

(part-of < event-type > *)
(goals < goal >*)
(preconditions < state-name > *)
(effects < state-name >*)
(is < EDL-expression >)
(described-by < clause >*))

'Entity' in the slot for the 'isa' relationship refers to the concept node in the world hierarchy that immediately dominates the node occupied by the given concept.

The world of TRANSLATOR is a union of a number of specialized *subworlds,* 'contexts'. Knowledge of the conceptual context is indispensable for disambiguation as well as for resolution of at least some anaphoric phenomena.

subworld-type ::= computer-world | office-world | business-world | everyday-world | ...

The conceptual actants we suggest for representing events (states and processes) include **Agent, Object, Patient, Source, Goal, Instrument, Time** and **Space.** The time and space slots are different from the rest in that they contain the *typical* time intervals and space environments for the event. Approximate scales are used: *instantaneous - seconds - minutes - hours - days - weeks - months - years - always* for time; *none - micro - hand - desk - room - house - town - everywhere* for space. This knowledge is used for disambiguation tests by the inference mechanism†. The *actual* time slots and actual spatial coordinates, or indirect references to them will be dealt with at the level of IL text, not in the dictionary.

The 'preconditions' and 'effects' clauses contain states of the world. The former describe those necessary for the current process to be successful or for the current state to set in. The latter describe new states of the world that are entailed from the event.

The event description in the 'is' slot of the process frame is either 'primitive' or a formula, written in modified EDL (Event Description Language, see Bates et al., 1981). This formula describes the sequence of processes that comprise a give (complex) process. A number of examples of EDL description of processes can be found in Nirenburg et al. (1985)

2.2. IL grammar.

In addition to instances (IL words) of entity types (IL lexemes), IL operates with three types of syntactic entities: propositions (IL clauses), utterances (IL sentences) and utterance clusters (IL paragraphs or texts). The types of information stored in the frames for the clause, the sentence and the text are described below.

1. The clause frame.
(< clause-id >
(constituent-of < sentence-id > OR < object-id >)
(main-clause? <y/n>)
(event < event-id > < negation >)
(agent (< object-id > < negation > < quantifier > *)
(object < object-id > < negation > < quantifier >*)
(patient < object-id >> < negation > < quantifier >*)
(instrument < object-id >> < negation > < quantifier >*)
(time < time-slot-value >)
(space < space-slot-value >
(source < object-id >> < negation > < quantifier >*)
(goal < object-id >> < negation > < quantifier >*)

† This knowledge will be used, for example, to deal with Bar Hillel's famous example of semantic ambiguity *The box is in the pen.* The comparison of the 'space' fields of the concepts corresponding to the two meanings of the English word *pen* will lead to immediate disambiguation.

(modality < modality-value >)
(subworld < subworld-name > *)
(given/new
(given < object-id > OR < event-id > OR < clause-id >)
(new < object-id > OR < event-id > OR < clause-id >))))

2. The sentence frame.

(< sentence-id >
(interclausal-structure < interclausal-structure-formula >)
(subworld < subworld-name >)
(modality < modality-value >)
(given/new
(given < object-id > OR < event-id > OR < clause-id >)
(new < object-id > OR < event-id > OR < clause-id >)
)
(speech-act
(type <sa-type>
(direct? (y/n))
(speaker < object-id >)
(hearer < object-id >*)
(perlocution < state >*)
(sa-time < time-slot-value >)
(sa-space < space-slot-value >))))

3. The text frame.

(< text-id >
(discourse-structure < discourse-structure-formula >)
)

Instances of entity types contain pointer fields to additional clauses that have these instances as slot fillers. This device allows, first, to accommodate nominal and verbal modifiers that do not fit into the predefined role structure, and, second, to retain the perspective of the input sentence as regards the relationship between its clauses (main and subordinate). If the secondary clauses are not 'anchored' in representations of entity instances, the IL representation will not be cohesive enough to be called a text. It will then be a 'memory state', as is, for example, the output of such semantic parsers as PLUM (Lehnert and Rosenberg, 1985),

Clause frame slot fillers are not merely object instances ('object-id's'). Additional information also includes the quantifiers and the negation operator (if present) whose scope is the object or event token. We decided to represent IL numerals as slot quantifiers. Therefore,

quantifier :: = all | most | many | some | certain | several | few | no | little | 1 | 2 | 3 | ...

We allow quantifiers on event instances (e.g. *almost* in (2)). Their semantic interpretation is that they represent the extent to which the event (process or state) has taken place.

event-quantifier :: = hardly | half | almost | barely | completely ...

The negation slot is a yes/no flag. The types of slot fillers that are not references to instances of objects or events are described below.

(2) John almost completed his weekly assignment.
produced by conceptual parsers.

† The COUNSELOR system at the University of Massachusetts (cf. Pustejovsky, 1985) seems to be a pertinent example. It is not an MT system. but rather a general reasoning system with a natural language front end. There is
[remainder of note absent in published version; but see page 236 where the footnote is repeated]

TIME slot values		
VALUE	**MEANING**	**EXAMPLE**
EQUAL (t_1, t_2)	time t_1 is the same as time t_2	*At 7 o'clock I was in the library*
BEFORE(t_1, t_2)	time t_1 precedes time t_2 on the time axis	*The rescue party combed through the forest and then searched the marshes.*
DURING (P, t_1, t_2)	event P starts after t_1 and ends before t_2	*During the day the tigers sleep*
AROUND (P, t)	event P takes place in the neighbourhood of t	*Let's meet threeish*
ALWAYS (P)	event P takes place at all times	*The Danube flows into the Black sea*

Table 1. The values of time slots in IL text.

Table 1 contains information about the values for time slots. The representation owes much to Allen (1983) as well as to Nirenburg (1981), though does not follow these approaches directly, due to the differences in the application areas (planning for Allen; semantic description of prepositional constructs for Nirenburg). The time points can be either mentioned directly (cf. (3)), by reference to an event (cf. (4)) or by reference to a time period before or after an event (cf. (5) and (6)). Time points or intervals can be left unspecified altogether (cf. (7) and (8)). There are also statements that cannot in principle refer to a time specification. Thus, one can pinpoint the time of the speech act in (9) but not the time at which the state described there is true.

(3) The proceedings began *at two o'clock.*
(4) We started the game only *when John returned.*
(5) The tickets did not go on sale until *two hours prior to the departure.*
(6) We'll know for sure *on the first Tuesday after a Monday in November.*
(7) These reviews take *an awful lot of time.*
(8) *God only knows when* these papers will arrive.
(9) Data collections such as the above we term databases.

Table 2 contains the possible values for space slots in event frames. The specification of points, intervals and regions in space is similar to the way time points and intervals are treated. The difference lies in the dimensionality: time has one dimension, while for space representation one has to describe three.

Points A, B and C in Table 2 can be overtly specified, as in (10). But more often, the system will have to derive the coordinates, typically, approximate or fuzzy, from reference to positions of certain objects or regions, as in the examples in the table. Points of space can also be referenced by the position at which a certain event takes place, as in (11). A special case is making reference to the position of one of the participants in the speech situation, as in (12).

(10) By 1400 hours the ship was *at 60° 34' North by 11° 20' West.*
(11) He could be seen *wherever people were singing.*
(12) Go straight ahead for three blocks and then turn *left.*

A very important generalization is the use of space parameters when manipulating mental objects. Natural languages are very strongly attuned to this type of non-poetic metaphor. Thus, the space marker will be allowed for **mental-objects**, such as, for example, the marker *IN* for the token

SPACE slot values		
VALUE	**MEANING**	**EXAMPLE**
EQ (A, B)	spatial coordinates of points A and B are the same (relative to the scale value)	*The expedition has reached the South Pole.*
LEFT-OF (A, B, C)	point A is to the left of point B looking from point C	*The only exit is to the right of the lemon tree.*
BETWEEN (A, B, C)	point A is between points B and C in space	*The Khyber Pass connects Afghanistan with Pakistan.*
IN (A, R)	point A is inside region R	*Timbuktu is somewhere in Africa.*
ON (A, O)	point A is on the upper surface of object O	*The book is on the table.*
ABOVE (A, B)	point A is above point B	*Clouds hung low over the treetops.*
NEAR (A, B)	point A is in the neighbourhood of point B	*Hamilton is just an hour from Syracuse.*
NONE	expected to be the default in a type of texts (e.g. textbooks) these can be called *spaceless* events; is distinct from *unknown* space slot value!	*Linguistic structures have been represented using a variety of formalisms corresponding to different levels of structure.*

Table 2. The values of **space** slots in IL text.

of the IL lexeme *set* in (13). Note that this phenomenon is not restricted to just spatial markers.

(13) Our set of options includes debugging this program, rewriting it or complete redesign of the system.

The values of modality include: *real* (cf. (14)), *unreal* (cf. (15)), *possible* (cf. (16)), *impossible* (cf. (17)), *necessary* (cf. (18)), *desirable* (cf. (19)) or *undesirable* (from the vantage point of a specified actant, cf. (20)).

(14) There are four chairs in this room.
(15) There could be four chairs in this room.
(16) It may well be that there are four chairs in this room.
(17) There can't be four chairs in this room.
(18) You must go.
(19) It would be nice if there were four chairs in this room.
(20) It's not good for you!

The 'interclausal structure formula' in the sentence frame represents the way the clauses connect into a sentence. The 'discourse structure formula' of a text represents the way the sentences connect in a text. This information makes explicit the cohesion forces in the text and provides a taxonomy for clause and sentence types. It is also used for defining the scope of anaphora resolution procedures. Note that we propose to use a single apparatus for accounting for the interclausal and discourse structure.

Our approach to the discourse structure problem has been influenced by the work of Sidner (e.g., 1985), Allen, Cohen and Perrault (e.g. Cohen and Perrault, 1979; Litman and Allen, 1984), Reichman-Adar (1984) and Pustejovsky (1985).

During the analysis of input the first priority is to recognize and fill the event and actant slots in clauses. If a clause is found not to fit any of the actant slots in another clause then a discourse structure relation must be established between it and another clause or an object in the sentence. Table 3 gives a sampling of discourse structure types in TRANSLATOR.

The speech act information adds a new dimension to MT-related representations. Together with the discourse data, it provides semantic analysis of IL sentences and utterances. The use of discourse and speech situation knowledge kindles the hope of avoiding the reference to the analysis of SL during the TL generation stages of the translation process.

We classify speech acts first into **assertions** and **requests**. Assertions are further subdivided into **definitions, opinions, facts, promises, threats,** and *advice.* The difference between definitions, opinions and facts is illustrated in (21). Whenever possible, these speech acts are recognized through lexical clues (e.g. 'it is my opinion that P'). When the speech acts are indirect, then **definitions** are the only ones that introduce new *types* of entities to the hearer. All the rest deal with entity tokens. **Facts** are those that speak of information about objects or events known to the speaker and the hearer and obtained or believed to be obtained through direct observation or pure logical inference. Finally, **opinions** speak of judgemental information about objects known to the speaker and the hearer.

(21) a) A collection of objects is called a *set* **(definition)**
b) I prefer Chopin to Brahms **(opinion)**
c) The Court House is at Court and Pleasant (**fact**)

Requests are further subdivided into **questions** (request-information) and **commands** (request-action). Questions classify into **yes/no** and **wh.** Commands include **orders, suggestions** and **pleas.**

3. An SL - IL dictionary.

The structure of the SL - IL dictionary is not in the focus of this paper. We would like to include a sampling of entries in such a dictionary in order to make the ideas behind the design of the interlingua more explicit. English will be used as an instance of the SL. There is no reason to require direct correspondence between the words of the source language text and 'words' (entity frames) of IL Some of the SL words are translated as IL words (e.g. nouns and verbs), others, as modifiers of IL 'words' (i.e., components of IL word frames), still others, as markers of text cohesion (components of the sentence and text frames of IL). A partial list of correspondences of SL (English) and IL entities can be seen in Table 4. A fragment of the English -- IL dictionary can be seen in Appendix 2.

Types of Inter-Clausal Relationships (Discourse Cohesion Markers)		
TYPE	**EXAMPLE**	**COMMENT**
SIMPLE (C)	*John sleeps*	a one-clause utterance
TEMP (C_1, C_2)	*I got up at 7 a.m. AND was at work by 8.*	There is always a universal weak suspicion of causality here
CAUSE (C_1, C_2)	*It rained, SO John stayed home.*	A problem for analysis is to distinguish this from ENABLE, cf. below
ENABLE (C_1, C_2)	*It stopped raining, SO John went outside.*	Compare with CAUSE
CHOICE (C_1, C_2, ..., C_n)	*EITHER we eat out, OR Melinda will have to cook.*	
EXAMPLE (C_1, C_2)	*Then we will do something nice LIKE going to the movies.*	
COMPAR (C_1, C_2)	*today the weather is much worse than yesterday*	comparisons along all possible parameter values of objects and events; the comparison is recorded in the corresponding parameter slots of all entities involved, thus allowing a *relative* rather than *absolute* values
EQUIV (C_1, C_2)	*This is tantamount to complete failure*	a special case of COMPAR; separated because of its wide usage and semantic peculiarity; definitions fall under this category, too
EXPAN (C_1, C_2)	*The man WHOM I met in the park went to Cork.*	more types of NL clauses can belong to this class than just relative clauses
CONDI (C_1, C_2)	*I'll take the exam IF you do too.*	
(+/-) SIMIL (C_1, C_2)	*with '+': John goes to the university AND Mary works in the bank;* with '-': *John goes to college BUT Mary works in the bank.*	there is always a suspicion of the existence of a stronger connecuon; therefore, this is a potential inference trigger

Table 3. Discourse structure parameters (cohesion types).

SL (English) Category	IL Representation
NOUN	object frame
ADJECTIVE	slot in object frame; state frame
VERB	action or state frame
MODAL	modality marker in clause and sentence frames
ADVERB	slot in action or state frame (time, space, ...); state marker ('he moved nonchalantly'): may introduce a separate clause frame
DETERMINER	marker in given/new; directs generation of new instances of objects vs. reference to existing instances...
CONJUNCTION	marker of sentence type; marker of cohesion
PREPOSITION	case marker; state marker: may introduce a separate clause frame
DEMONSTRATIVE	marker of deixis
NUMERAL	a quantifier operator on IL NPs
PRONOUN	marker of deixis; reference to object
NP	marks boundaries of case slot values
VP	helps mark boundaries
PP	helps mark boundaries
CLAUSE	fills clause frame
S	fills sentence frame
PARENTHETICALS	discourse markers; connector clues for sentences

Table 4. Selected categories of English with their IL counterparts.

4. An IL Sentence and an IL Text.

This paper does not describe the peculiarities of the process of translating between SL and IL or IL and TL. The emphasis is on providing an adequate output of the SL - IL translation and an adequate input for the IL - TL translation. These are traditionally the parsing and the generation stages. Experience shows that the gap between the results of parsing and the input for generation is quite large. Special reasoning subsystems are typically employed to bridge this gap†. In TRANSLATOR this is the augmentor/inspector. We would like to narrow the augmentor/inspector mandate as much as possible by making IL a language capable of holding more types of information than is typically produced by conceptual parsers.

† The COUNSELOR system at the University of Massachusetts (cf. Pustejovsky, 1985) seems to be a pertinent example. It is not an MT system, but rather a general reasoning system with a natural language front end. There is a very significant distance between the output of both the conceptual parser and the reasoning system itself on the one hand, and the information necessary for adequate generation on the other. The language of the former representations is simply not powerful enough.

4.1. A Sentence.

In what follows we display the IL translation of a sentence from the subworld we would like to apply TRANSLATOR to first, the computer-world. The sentence is from Ullman, 1982, p.1.:

(22) Data, such as the above, that is stored more or less permanently in a computer we term a *database.*

Intuitively, we would like to be able to extract the following information from (22): the author gives a definition for the term *database* by stating that it is a kind of data; the data is stored in the computer; the period of storage is typically long; an example of data that is defined as database is given previously in the text, most probably, in the paragraph immediately preceding the one in which the sentence appears. In what follows the IL representation for this sentence is given, with comments.

```
(sentence
  (id S1)
  (discourse-structure SIMPLE: clause C1)
  (subworld computer-world)
  (modality real)
  (given/new
   (given (data ... in a computer))  ;cf. clause C1 below
   (new   database)
  )
  (speech-act
   (type definition)
   (direct? yes)
   (speaker AUTHOR)
   (hearer READER)
   (perlocution < READER knows definition of database >)
   (sa-time < time of reading >)
   (sa-space none)
)))

(clause                          ;'we term NP database'
  (id C1)
  (constituent-of S1)
  (main-clause? y)
  (event (process (define))
  (agent AUTHOR)
  (object database)
  (patient data1)        ;head of the NP above
  (modality real)
  (subworld computer-world)
  (given/new
(given (data ... in a computer))
(new   database)
)))

(data1                           ;'data'
  (isa mental-object)
  (subworlds computer-world business-world)
  (adjunct-clauses (+SIMIL C2, C3))
)

((be- example- of 1
```

(isa mental-state)
(subworlds science-world)
(patient *data*$_1$) ;this is a two-place predicate whose
(patient *data*$_2$) ;arguments are of equal status
(typical-time always)
)

(*computer - memory*$_1$;' memory'
(isa device-component)
(subworlds computer-world)
(source-of information)
(goal-of (process (store))
(consists-of file)
(part-of *computer*$_1$)
)

(computer$_1$;' computer'
(isa device)
(subworlds computer-world business-world college-world)
(object-of use)
(instrument-of solve analyze)
(source-of information)
(consists-of CPU *memory*$_1$ peripherals)
)

(speaker$_1$;'we'
(isa human)
(subworlds everyday-world)
(agent-of process)
(object-of process)
(patient-of state)
)

(database$_1$;database token here same as type
(isa)
(subworlds computer-world)
(object-of use consult create modify)
(source-of information)
(consists-of file record DBMS)
)

(store$_1$ *;'* store'
(isa physical-process)
(subworlds everyday-world)
(agent human)
(object data)
(time always)
(space *computer*$_1$)
(goals maintain-database)
(preconditions (Thereexists object)(Thereexists space))
(effects IN (object, space))
)

(define ;'term'
  (isa speech-process)
  (subworlds science-world)
  (agent AUTHOR)
  (object *database*$_1$)
  (patient *OBJECT*$_1$)
  (time (time of writing))
)

(clause ;'data is stored more or less permanently in a computer'
  (id C_2)
  (constituent-of *data*$_1$)
  (main-clause? n)
  (event (process (store))
  (agent unknown)
  (object *data*$_1$)
  (space (IN store computer))
  (time < almost > ALWAYS)
  (modality real)
  (subworld computer-world)
  (given/new
      (given 'data is stored more or less permanently')
      (new 'in the computer')
)))

(clause ;data is such as the above
  (id C_3)
  (constituent-of *data*$_1$)
  (main-clause? n)
  (event (state-equivalency))
  (object *data*$_1$)
  (modality real)
  (subworld computer-world)
  (given/new
      (given ′ *data*$_1$ is an example of')
      (new *data*$_2$)
)))

(*data*$_2$;data above
  (isa mental-object)
  (subworlds airline-world)
)

4.2. A text.

In what follows we show the representation in IL of a 'text', actually, the paragraph from which the sentence (20) was taken. First, we present the text, as a sequence of sentences, with (20) repeated here as S1.

S1 Data, such as the above, that is stored more-or-less permanently in a computer we term a *data-base.*

S2 The software that allows one or many persons to use and/or modify this data is a *database management system* (DBMS).

S3 A major role of the DBMS is to allow the user to deal with the data in abstract terms, rather than as the computer stores the data.

S4 In this sense, the DBMS acts as an interpreter for a high-level programming language, ideally allowing the user to specify what must be done, with little or no attention on the user's part to the detailed algorithms or data representation used by the system.

S5 However, in the case of a DBMS, there may be far less relationship between the data as seen by the user and as stored in the computer, than between, say, arrays as defined in a typical programming language and the representation of those arrays in memory.

The sentences S1 through S5 are connected in the following way:

(T1
(discourse-structure-formula(+ SIMIL(S2,S1) +SIMIL(S3,S2) +SIMIL(S4,S3) -SIMIL (S5, S4))))

A possible problem with this representation is that may not be sufficiently expressive. Unlike the dialog, to which most of the discourse approaches were attuned, the text, especially the non-narrative text, seems to have significantly fewer syntactic markers of cohesion. Experimentation will prove whether it is necessary to augment the formalism needed for representing the discourse structure of the text.

5. Conclusion.

A major (possibly, the major) reason for, at best, limited success of the research toward fully-automated MT is, in our opinion, the widespread desire to perform translation without extracting sufficient knowledge from the input text. We believe that a discussion of the amount of such knowledge necessary and sufficient for successful MT should precede the discussion of the best ways to extract and represent this knowledge. In other words, in this sense it is less important at this point to ponder a choice of syntactic formalisms for parsing SL texts than a) to understand what specific uses will the information obtained through such analysis be put to and b) to detect the types of information necessary and sufficient for MT. Proof and/or verification procedures must be developed for this purpose.

One point to remember in this respect is that theory of translation in general and of MT in particular both belong to the class of application theories that must be clearly distinct from basic (linguistic or otherwise) theories. It is an open question to decide *to what extent* it is appropriate to import general methods, even from a closely related theoretical field, such as theoretical linguistics for MT, to solve the peculiar problems of an application (see discussion of the problem of application theories in Raskin, 1985).

We do not share the opinion that a 'deep understanding' of an SL text is either impossible or unnecessary for MT. It remains to be seen whether it is impossible. Also, if this proves to be the real state of affairs, this would mean that fully automated MT as such is impossible. Our efforts, we hope, will help decide this puzzle. As to the necessity of knowledge-intensive analysis in MT, the only way to bypass it is to shift the main bulk of knowledge from grammars into dictionaries, especially, the bilingual SL - TL dictionaries in hope to aid disambiguation of senses. In the inevitable cases of failure or indecision heavy post-editing is to be used†. In any case, MT without involved semantic and pragmatic analysis of text can be possible only if the lexicographers employed by such projects compile dictionaries that attempt to preempt all disambiguation problems and record them manually. Such an approach may even be in principle feasible, but the nature of work in it will be such that few or no available methods for automatic knowledge handling will be used.

---

† We do not claim that post-editing should be completely eliminated from the design of an MT system. Our desire is to see MT post-editing resemble editing human translations. It is the *types* of errors in translation, not so much their quantity that we are concerned with here.

The TRANSLATOR interlingua sketched above will be tested extensively and will obviously undergo changes in the process. The next step in the project will be writing a cluster of parser programs that will transform SL text into its IL representation. The design of the parser cluster will be reported elsewhere. On the generating side, the next step is to build a preprocessor for generation that will translate the information in the IL frames into the form required by the generator to assure correct TL realization.

Acknowledgements. The authors would like to thank Irene Nirenburg and James Pustejovsky for many fruitful discussions of the subject matter of this paper.

Bibliography.

Allen, J.F., 1983. Maintaining knowledge about temporal intervals. *Communications of ACM,* vol. 26, No. 11, pp. 832-843.

Bates, P., J. Wileden and V. Lesser, 1981. A language to support debugging in distributed systems. University of Massachusetts COINS Technical Report 81-17.

Carbonell, J., 1979. Computer models of human personality traits. Proceedings of IJCAI-79. 121 - 123,

Cohen, P.R. and C.R. Perrault, 1979. Elements of a plan-based theory of speech acts. *Cognitive Science,* 3, 177-212.

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

Lehnert, W. and S. Rosenberg, 1985. The PLUM parser user's guide. COUNSELOR Project Technical Report. University of Massachusetts.

Litman, D. and J.Allen, 1984, A plan recognition model for clarification subdialogues. Proceedings of COLING-84. Stanford, 302 - 310.

Nirenburg, S., 1981. A method of formal semantic description of prepositional phrases in Russian. Discussion paper 51, Department of Computer Science, The Hebrew University of Jerusalem.

Nirenburg, S., I. Nirenburg and J. H. Reynolds, 1985. Control knowledge in POPLAR, a personality-oriented planner. In: Proceedings of Conference on Intelligent Machines. Oakland University, Rochester, Michigan.

Pustejovsky, J., 1985. The level hypothesis in discourse analysis. This volume.

Raskin, V., 1985. Linguistics and natural language processing. This volume.

Reichman-Adar, R. Extended person-machine interface. *Artificial Intelligence, 23,* 157 - 218.

Schank, R. C. and R. Abelson, 1977. **Scripts, Plans, Goals and Understanding.** Hillsdale, NJ: Erlbaum.

Sidner, C. L., 1985. Plan parsing for intended response recognition in discourse. *Computational Intelligence,* 1.

Tucker, A. B. and S. Nirenburg, 1984. Machine translation: a contemporary view. In: **Annual Review of Information Science and Technology,** 19, White Plains, NY: Knowledge Industry Publications, pp. 129 - 160.

Ullman, J.D., 1982, **Principles of Database** Systems. Rockville, MD: Computer Science Press.

Wilensky, Robert, 1983. **Planning and Understanding.** Reading, MA: Addison-Wesley.

Appendix 1. Examples of entries in the IL monolingual dictionary.

Note that in TRANSLATOR, due to the existence of bilingual dictionaries, only the leave nodes in the quasi-hierarchy will have instances. The superstructure will be maintained to provide a kind of a

checklist for knowledge to be detected about a given terminal node occupant.

```
(physical-object
  (isa object)
  (subworlds *all*)
  (object-of change-position-of)
  (shape < shape-value >)
  (mass < integer >)
  (color < color-value>)
)

(mental-object
  (isa object)
  (subworlds *all*)
)

(information
  (isa Mental-Object)
  (object-of send-info receive-info process-info)
  (consists-of message)
)

(program
  (isa Information)
  (subworlds computer-world)
  (object-of run)
  (consists-of procedure-comp function-comp)
  (part-of system-comp)

(algorithm
  (isa terminology)
  (subworlds computer-world)
  (object-of create use)
  (instrument-of solve)
  (consists-of procedure function statement command)
)

(array
  (isa data-structure)
  (subworlds computer-world)
  (goal-of store)
  (consists-of data)
)

(permit
  (isa speech-process)
  (subworlds everyday-world)
  (subject human)
  (object event)
  (patient human)
  (time instantaneous)
  (space none)
)

(computer-user
```

```
  (isa person)
  (subworlds computer-world)
  (agent-of operate)
  (space (chair))
  (consists-of head hand)
)

(operate
  (isa physical-process)
  (subworlds computer-world)
  (parent-of (store insert delete retrieve))
  (agent user)
  (object computer)
  (time always)))

(data
  (isa information)
 (subworlds compute-world)
 (object-of store retrieve modify)
 (consists-of file record byte)
(part-of database)
)

(store
  (isa operate)
  (subworlds computer-world)
  (agent user)
  (object data)
  (time instantaneous)
)

(computer
  (isa device)
  (subworlds computer-world)
  (instrument-of compute operate)
  (consists-of CPU memory peripherals)
)

 (database
 (isa data)
(subworlds computer-world)
(object-of design implement)
(consists-of data)
)

(DBMS
  (isa program)
  (subworlds computer-world)
  (instrument-of (operate
                 (object data))
)
```

Appendix 2. A Sampling of entries in the SL (English) -- IL dictionary.

English lexeme	IL correlate
above	1. space marker: (ABOVE A, B) 2. deictic clue ('the above statement' has as referent a previously instantiated proposition).
algorithm	'algorithm' (cf. Appendix 1)
allow	1. 'permit' (cf. Appendix 1) 2. Cohesion marker ENABLE
array	'array' (cf. Appendix 1)
as	1. Cohesion marker CAUSE 2. Cohesion marker EQUIV
computer	'computer' (See Appendix 1)
data	'data' (see Appendix 1)
database	'db' (see Appendix 1)
databaseManagementSystem	'dbms' (see Appendix 1)
DBMS	'dbms' (see Appendix 1)
for	1. marker of the 'beneficiary' slot in process frames 2. cohesion marker CAUSE.
however	cohesion marker -SIMIL
lessThan	cohesion marker COMPAR
inThisSense	cohesion marker + SIMIL
little	1. (relative) value of slot 'size' of objects 2. event-quantifier (meaning defined by position on the scale)
many	quantifier (meaning defined by position on the scale)
moreOrLess	event-quantifier (meaning defined by position on the scale)
ratherThan	cohesion marker -SIMIL
store	'store' (cf. Appendix 1)
suchAs	cohesion marker EXAMPLE
the	1. existence marker of an object instance 2. deixis marker to an earlier introduced object
this	1. deixis marker to an earlier introduced object or event 2. cataphora marker (as in *let me tell you this: ...)*