Communicating with Language: Mapping from a Concepual Model to Lingustic Form

Marie Meteer
BBN Systems and Technologies
Cambridge, MA 02138
MMETEER@BBN.COM

ABSTRACT

Communication among agents in a natural language places certain requirements on the conceptual model, and language itself places constraints on what can be communicated fluently, without circumlocution. In this paper, I explore the relationship between conceptual modeling and natural language generation, focusing on the problem of representing and describing events. A text generator producing the linguistic representation of actions and objects, either as part of the communication among agents or a description of their actions, has its own set of requirements for what information needs to be represented in order to map from the model to linguistic form. I describe six kinds of information needed for the expression of events: linear time, event type, temporal modifiers, event structure, and agency and discuss whether that information best reside in the underlying model or in the linguistic component. The discussion is grounded in a particular system, SAGE, which includes an event based simulator, knowledge base, and text generator. However, the issues are the same whether the underlying model drives a simulator, is the output of vision system or some other program.

1. INTRODUCTION

In any natural language generation (NLG) task, the essential problem is communicating the information represented in an underlying program to a particular audience. Central to this is the problem of spanning the distance between the representation used in a conceptual model and language. In a multiagent simulation, this model must support the actions of the agents and their effects on objects in the environment. A text generator producing the linguistic representation of these actions and objects has its own set of requirements for what information needs to be represented for the mapping from the model to linguistic form. For example, in a simulation, the sex of the agents may be immaterial, and thus not represented explicitly. However, in order for a text generator to be able to use pronouns ("he", "she") the information must be there.

For narration, temporal and aspectual information must be available in the underlying model in order to describe events. While most studies of events and their description are done within the realm of linguistics, where the focus in on the expression of event descriptions, it is clear that the way events are modeled is also an essential element. Bach (1988) describes "how certain metaphysical assumptions are essential to an understanding of English tenses and aspects. These assumptions have to do with the way reality—or our experience—is structured."

From a generation perspective, there are three basic questions to be answered. First, what information is needed in order to produce the distinctions available in language; secondly, which distinctions are facts of language (and thus should be in the generator) and which are better represented at the model level, and thirdly, what are the alternatives available to the speaker given the constraints of the model and the constraints of language.

In this paper, I look at what information is needed to generate events through an analysis of events and a review of the linguistic literature. I look most closely at two problems: the granularity of events and event types. The discussion is grounded in an actual system, SAGE, a "Simulation and Generation Environment", which provides components for both conceptual modeling and text production.

2. EVENTS

In this section, I address the problem of representing and describing events. The goal is to identify the information that needs to be represented in order to take advantage of all the resources a language provides for describing events (which involves determining what distinctions language supports) and determining at what level the information should be represented and the decisions made to make those distinctions.

We first outline six different kinds of information needed for the expression of events: linear time, event type, temporal modifiers, event structure, argument structure, and agency.

2.1 Linear time

First, in order to generate events, there needs to be a model of linear time. Most of the current work on tenses is based on a Reichenbach-style analysis, which involves

three temporal notions, point of event (E), point of speech (S), and point of reference (R), are needed in order to adequately account for the English tense system, as shown in the following examples:

1. *Peter drove to work. (E = R < S)*
2. *Peter had driven to work. (E < R < S)*

Two of the points, E and S, are facts of the model, when the event took place and the time the speaker is producing the utterance. However, the third point, the reference time, is a fact of the discourse, a choice to be made by the speaker. (1) and (2) above are distinguished by the reference time, but otherwise could describe the same event and be spoken at the same time.

2.2 Event type

Another well recognized distinction is that of event types, such as state, process, transition, exemplified by the following examples:

3. *The mouse is under the table. (state)*
4. *Fluffy ran. (process)*
5. *Peter found his keys. (transition—achievement)*
6. *Helga wrote a letter. (transition—accomplishment)*

While verbs have an intrinsic type (e.g. wait is a process and catch is a transition), these types also apply to whole phrases since tense, aspect, adjuncts and arguments can compose with the type of the lexical head to form a new type:

7. *Fluffy ran into the kitchen. (process —> transition)*
8. *Helga is writing a letter. (transition —> process)*
9. *The mouse is caught. (transition —> state)*
10. *Roscoe builds houses. (transition —> iteration)*

2.3 Temporal adverbials

Four kinds of temporal adverbials can be distinguished and are linked to the event types. *Duration* modifies processes, as in example (11a), but not transitions (11b); *frame* adverbials modify accomplishments, as in (12a), but not processes (12b); *point* adverbials modify achievements, as in (13); and *frequency* adverbials modify iterative events, as in (14).

11. a) *Peter waited in the lobby for an hour.*
    b) ** Helga wrote the letter for an hour.*
12. a) *Helga wrote the letter in an hour.*
    b) ** Peter waited in the lobby in an hour.*
13. *Hank found the pen at four o'clock.*
14. *Martha writes letters frequently.*

2.4 Event structure

It is also clear that events are not undifferentiated masses, but rather have subparts that can be picked out by the choice of phrase type or the addition of adverbial phrases. Moens & Steedman (1988) identify three constituents to an event nucleus: a preparatory process, culmination, and consequent state, whereas Nakhimovsky (1988) identifies five: preparatory, initial, body, final, result, exemplified by the following:[1]

15. *When the children crossed the road,*
    a) *they waited for the teacher to give a signal.*
    b) *they stepped onto its concrete surface as if it were about to swallow them up.*
    c) *they were nearly hit by a car.*
    d) *they reached the other side stricken with fear.*
    e) *they found themselves surrounded by strangers.*

Pustejovsky (1991) offers a much more compositional notion of event structure, where a transition is the composition of a process and a state. This analysis is more closely tied to the lexicon than Moens and Steedman's or Nakhimovsky's (and is offered in the context of a generative theory of lexical semantics). It not only accounts for the semantics of verbs, but also their compositions with adjuncts to form new types, as in (7) above.

2.5 Argument structure

The participants of an event are those entities that act in or are acted upon in the event. The argument structure is the set of participants in the event that are grammaticized with respect to a particular lexicalization of the event, such as the agent, theme, source, and goal. For some event types (especially those that appear as examples in linguistics papers), the distinction between what is an argument and what is an adjunct is clear. For example, in "Fluffy ate a bone in the dining room yesterday", "Fluffy" (the agent) and "a bone" (the theme) are arguments, whereas the location and time are adjuncts. For other verbs, however, the distinction is not so clear, as in "Mickey slid into home plate", where the location is a necessary participant to the meaning, yet as a location it would be treated as an adjunct in most analyses.

2.6 Agency

Agency in an event is an aspect of the argument structure, but since there are some important generalizations over this participant that is not true of others, we treat it separately. One of the most widely discussed syntactic variations is active/ passive, which vary on the placement/inclusion of the agent. There are really many different motivations for what is often characterized as a single "switch" in generators. The degree of explicitness of the agent in different syntactic constructions can be seen in the following set of examples, from the explicit inclusion of the agent in the subject position in (a), to the movement of the agent to the by-phrase in (b), to the deletion of the agent in (c), to an adjectival construction in (d) using the past participle form of the verb, to a result construction in (e) that includes no indication of agency. Notice that the explicitness of the event's tense diminishes along with the agency.

[1] Nakhimovsky, 1988, p.31.

18. a) *Peter tore the shirt.*
b) *The shirt was torn by Peter.*
c) *The shirt was torn yesterday.*
d) *Peter wore the torn shirt yesterday.*
e) *No one noticed the tear in the shirt. (cf. No one noticed the missing button.)*

Another argument that agency should be treated specially is made by Pustejovsky (1991) in his work in generative lexical semantics and event structure. Pustejovsky argues that some distinctions usually characterized by event type or argument structure are actually rooted in agency, such as the difference between verbs that are lexically transitions but have unaccusative and causative variants ("The door closed" vs. "Thelma closed the door"). Furthermore, the difference between the two types of transitions, accomplishments vs. achievements, is based on an agentive/non-agentive distinction. According to Pustejovsky, accomplishments (such as build, draw, and leave) include both the act and the causation in their semantics, whereas in achievements (such as win, find, and arrive) agency is not an intrinsic part of the semantics of the verb, but is rather based on something else, such as the configuration of elements (someone wins when they are at the front in some competition at a particular moment, given some particular evaluation function). This is substantiated by the interaction with "deliberately" and these verbs, shown in the examples below:

19. a. *Helga deliberately drew a picture*
b. **Helga deliberately found the pen.*
20. a. *Peter deliberately left the party.*
b. **Peter deliberately arrived at the party.*

3. REPRESENTING THE INFORMATION

Having identified the information necessary for the description of events, the next step in the research is to determine which levels should be responsible for the representation of the information. In particular, what aspects of the event description are

- dependent on the event itself (a fact of the world/model);
- dependent on the discourse context;
- dependent on what linguistic resources are available (e.g. lexicon and syntax) and constraints on their composition.

In addition, of course the description is dependent on the goals and intentions of the speaker. For the purpose of this discussion, we will assume that the goal is to simply describe the event, for example as part of a narration.

In this section we focus on two issues that are of particular importance in mapping from a conceptual representation of events to a linguistic one: the granularity of events and event type. We first describe the SAGE system, in which we will be grounding the discussion.

3.1. SAGE

SAGE is a package of integrated tools to aid in exploring the relationship between simulated events in a multi-agent environment, the narration of those events by a text generator, and the animation of the events with simple graphics. There are three main components to SAGE:

- The speaker's intensional world is modeled in an "underlying program" built using the knowledge representation language VSFL and the event based simulator SCORE;[2]
- The text generator is SPOKESMAN, with the linguistic component MUMBLE-86 and the text planner Ravel;
- The graphics component is built with the graphics package in Macintosh Common Lisp and Mac Quickdraw.

We have seen that in SAGE what are generally treated as a single phenomenon stretch across multiple levels:

- Event time and speech time are facts of the simulator, whereas reference time is part of the discourse model in the generator.
- Events have an intrinsic type in the model, but the speaker can make explicit only a portion of the event or can compose it with other information and express it as a different event type. What subconstituents of an event are available to be made explicit are defined by the procedures of the underlying program (in this case, the simulator), but the ways they can be made explicit are constrained by the resources of language.
- Whether an action is caused by an agent is part of the definition of the action, but whether that agent is expressed is a choice by the speaker.

In all of these cases, the information must be represented at both the model level and in the generator in order to capture the full expressiveness of event descriptions in English. Using SAGE as an environment in which to model both conceptual and linguistic information has let us experiment with the best division of the information across its components.

The simulation begins with each of the agents located at a position on a map of a house. Fluffy the dog is assigned a goal of catching a mouse and Jake the mouse is assigned the goal of getting some cheese, which is located in the kitchen. The following simple paragraph, generated by Spokesman, describes each of their goals and actions and is produced incrementally as they are executed by the simulator:

---

[2] VSFL ("Very Simple Frame Language") and SCORE ("Sproket Core") were developed at BBN Systems and Technologies by Glenn Abrett and Jeff Palmucci, with assistance from Mark Burstien, and Stephen Deutsch. See Abrett, et al. (1989) for a more detailed description of these systems.

Fluffy wants to catch a mouse. He is looking for her.
The mouse wants to get cheese. She is leaving a mouse-house. She is going toward it.
Fluffy is chasing the mouse. He is going toward her. He caught her.
The mouse didn't get the cheese.

Details of the connection from the underlying program to the text generator are described in Meteer 1994.

3.2 Granularity of events

In our system, events are represented as goals and procedures in the simulator and are also linked to the knowledge base through their types, which are concepts in the knowledge base. This provides a classification of events into the three main event types: state, process, and transition. The parameters to those goals/procedures are the roles on the concept, defining the participants in the event, as well as associated information, such as location.

The simulator provides a language for declaratively representing the plans of agents, where a *plan* is a partial ordering of procedures and subgoals for accomplishing goals and handling contingencies. *Goals* define the intentions of agents (goals succeed or fail) and *procedures* define a sequence of actions and decision points (procedures complete or are interrupted). The primitives in this system are *actions*, which are simply Lisp functions.

The hierarchical structure of the plans, with procedures defined in terms of subprocedures and actions, defines the structure of events, in the sense of Nakhimovsky, described above. The procedure for cross-the-road, for example, would be defined in terms of prepare-to-cross (look both ways, wait for traffic, wait for teacher's signal, etc.,) step onto the road, walk across, step on to the other side, with a consequent change in that agent's state (more specifically, his location) from one side of the street to another.

Note that in these terms, the constituents of an event are a fact of the model, i.e. what level of granularity is represented, and not a linguistic issue. We can describe the event as a single action "cross the road", but with an animation component, each of the steps must be modeled as well (depending, of course, on the granularity of the animation, since if the "road" is a single line, then a single action might be adequate to move the agent across it).

The options available for expressing a single "event" (whether compositional or not), are dependent on what lexical resources are available in the language. For example, we have a single lexical item "walk" for taking steps, but for the entire action of getting from one side of the road to the other, we need a verb and argument ("cross the road"). We can pick out the beginning portion of the crossing with the verb "start" as in "start to cross the road", but cannot easily get at walking three quarters the way across (note that "almost crossed the road" is available, but ambiguous).

The question here is not whether something is expressible *at all* in language, but rather what the linguistic resources of the language make it easy to express and how to best make use of those resources. The choice of what level of granularity at which to describe an event is clearly dependent on the purpose of the utterance and its context. If the context is setting the stage for a description of nearly getting run over by a car, then the larger event needs to be broken up, however if it is a description of getting to school, then treating the event as a single action "cross the road" may be more appropriate.

3.3 Event type

Another kind of information needed for generation is the event type. Note that in SAGE there is not a single notion of "event type", but rather two: one for the underlying knowledge base and the other for Ravel, the text planner. This reflects the difference between:

- a concept's intrinsic type in the domain, which includes what objects it is related to (e.g. its parents, what slots it has), and how it functions in the underlying program (e.g. what methods it has or inherits), and
- a concept's "expression type" in the text planner, which reflects the fact that the speaker can alter an object's expression type through lexical choice (e.g. nominalization) and the choice of tense, aspect and adjuncts.

Portions of these two types of classification hierarchies are shown in Figure 1.

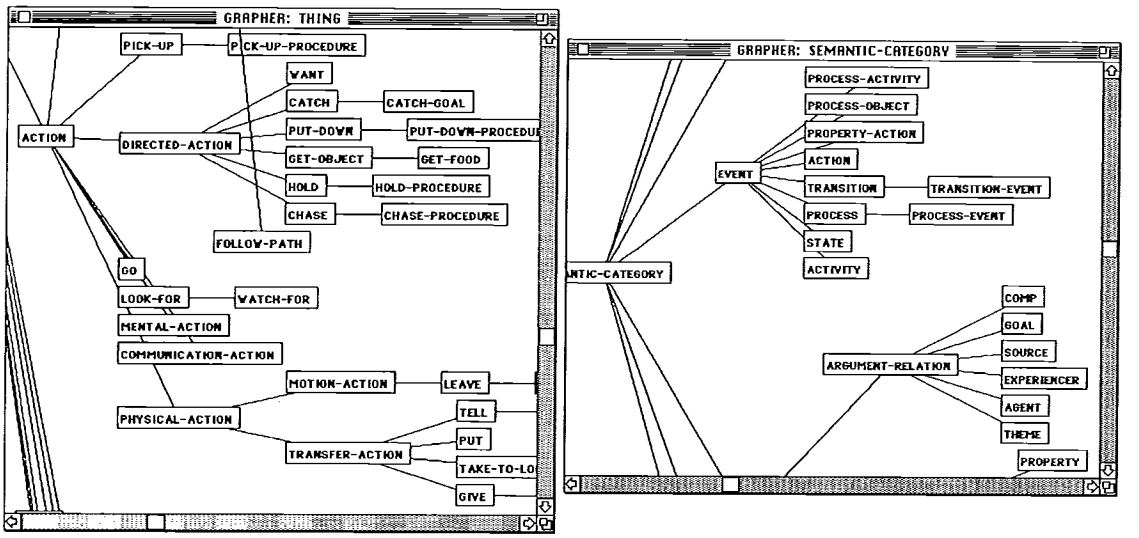

Figure 1: Event types in the Domain Model and Text Planner

Event types combined with other information, such as arguments, tense, and aspect to pick out different views on an event and thus allow different kinds of information to be expressed, such as different temporal modifiers, as we discussed above.

For example, for the same event in an underlying situation, a variety of expressions of that event are allowed, as exemplified in Figure 2. Note that the verb "catch" can entail either the entire action set (wait, watch, see, chase catch) when it occurs in a sentence like "Fluffy wants to catch a mouse" or just refer to the final act as in "Fluffy caught the mouse at 10 PM".

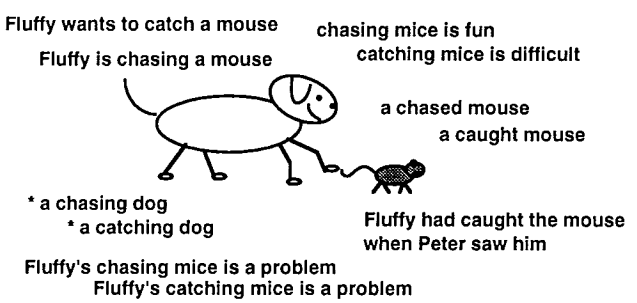

Figure 2: Various descriptions for the same underlying actions

Since "catch" and "chase" are intrinsically different event types ("catch" is a transition and "chase" is a process) they take different modifiers (*Fluffy chased a mouse for an hour* vs. **Fluffy caught a mouse for an hour*). Note that this is a fact about language and not a fact about the conceptual model. Therefore this constraint is captured in the text planner in a representation level called the Text Structure in Spokesman.

It is at this level that the content lexical items are selected and the semantic category of the constituents is determined. Events and their composition are handled in the style of Pustejovsky (described above). For example, a RUN-TO-LOCATION procedure in the simulator (which has a type of transition) is mapped to the composition of the lexical head "run" (with the agent from the WHO role of the procedure), which is lexically a process, with a goal locative adjunct (e.g. "to the kitchen"), which produces a transition as shown in the Text Structure tree in Figure 3. Constraints on the transition type indicate that only a frame adverbial (e.g. "in two minutes"), can be added, and not a duration (e.g. "for two minutes").

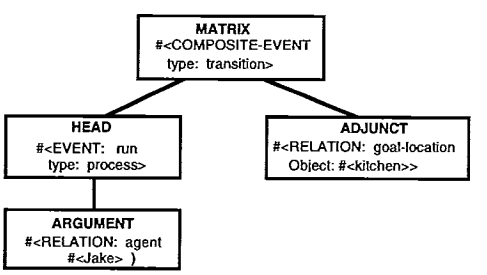

Figure 3: Text Structure Tree

The speaker could also choose not to express the entire transition as a kernel unit, but rather pick out only the process portion, as in "Jake ran", in which case the composition would also be of type process, which constrains the temporal adjuncts to be of type duration, rather than frame. (See Meteer, 1992, for a more complete description of the vocabulary of terms in the text structure and its role in the incremental composition of the text plan.)

5. CONCLUSION

The goal of this paper was to demonstrate the kind of information needed in a conceptual model in order to support natural language. The focus here is on events,

however, the same careful study is needed to determine the information required for generating text or speech and to make a judicial division into that which is a part of language and should be in the linguistic components and which are a part of the world and should be part of the conceptual model.

The greatest area of research is how to mitigate among the choices available to the speaker and to balance between realizing the goals of the speaker and making maximum use of the linguistic resources available.

References

Abrett, G., Deutsch, S. and Downes-Martin, S. (1989), "AI Languages for Simulation", BBN Technical Report, BBN Systems and Technologies, Cambridge, Massachusetts.

Bach, Emmon (1981) "On Time, Tense, and Aspect: An Essay in English Metaphysics" in , Academic Press.

McDonald, David D. (1984) Description Directed Control. *Computers and Mathematics 9(1)* Reprinted in Grosz, et al. (eds.) *Readings in Natural Language Processing*, Morgan Kaufman Publishers, California, 1986, pp.519-538.

McDonald, David D. (1991) "On the Place of Words in the Generation Process" in *Natural Language Generation in Artificial Intelligence and Computational Linguistics.* Eds. Cecile Paris, William Swartout and William Mann, p.229-247.

Meteer, Marie W. (1994) "Generating Event Descriptions with SAGE: a Simulation and Generation Environment" 7th International Workshop on Natural Language Generation, June 21-24, 1994, Kennebunkport, ME.

Meteer, Marie W. (1992) *Expressibility and the Problem of Efficient Text Planning.* Pinter Publishers, London, England.

Meteer, Marie W., David D. McDonald, Scott Anderson, David Forster, Linda Gay, Alison Huettner, Penelope Sibun (1987) *Mumble-86: Design and Implementation*, UMass Technical Report 87-87, 173 pgs.

Moens, Marc & Steedman, Mark (1988) "Temporal Ontology and Temporal Reference" *Computational Linguistics*, Vol.14, No. 2, p.15-28.

Nakhimovsky, Alexander (1988) "Aspect, Aspectual Class, and the Temporal Structure of Narrative" *Computational Linguistics*, Vol.14, No. 2, p.29-43.

Pustejovsky, James (1992) "The Syntax of Event Structure" *Cognition*, Vol. 41, 47-81.

Webber, Bonnie (1988) "Tense as Discourse Anaphor" Computational Linguistics, Vol.14, No. 2, p.61-73.