Situation Viewpoints for Generation

Henry Hamburger[1] and Dan Tufis[2]

ABSTRACT: Representation systems are presented for the input and output of the first or deep phase of a language generation system. Actions and viewpoints are the key factors in determining what sentence is produced; viewpoints provide a wide range of ways to discuss actions, their states and the plans they compose. The language generator plays a key role in a two-medium conversational system for a naturalistic foreign language learning environment.

KEYWORDS: viewpoint, action-based natural language generation, two-medium, conversation

Overview

After an introduction to the nature and role of viewpoints, we motivate this work in terms of our two-medium system for conversational language learning. Since our version of generation is action-based, we then sketch actions. Finally, we return to a finer-grained look at viewpoints.

1 Viewpoints

In the natural use of natural language, a single event can be talked about in a variety of ways, taking a variety of viewpoints. Sueh variety is necessary across languages because of differences in how cultures prefer to express things (Delin et al., 1993) and because of differences in how languages make it possible to express things (Felshin, 1993). A selection of viewpoints is also needed within languages, both for coherence (Meteer, forthcoming) and for effective rhetoric (Hovy, 1988).

For us, varied viewpoints are a way to expose learners of a foreign language to a variety of linguistic constructions in the naturalistic, situation-based, two-medium (graphical as well as linguistic) conversations that take place in our foreign language learning environment called Fluent-2. To achieve this objective, we have been developing and implementing our notion of an abstract situation viewpoint, hereafter called simply a view.

A view specifies a way to operate on an action or a possible action in a situation to produce a language-independent conceptual structure that corresponds to a statement, command or question about an action or its results, purpose, participants, etc. This paper sketches an internal structure for views and indicates their range of expression. The choice of which view to use at a particular point can be made by the tutorial strategist, taking into account the student's limited knowledge of the language (Hamburger, in press). View processing is the deepest of three levels forming the NLG capability of the learning environment. The general idea of views can be seen from a few examples in three categories: action, state and plan views.

1. You picked up the pot.
   [description of an action]
2. The pot is in your hand.
   [description of a state]
3. Now fill the pot.
   [command to continue plan]
4. The water is not on.
   [unmet precondition]
5. What is (still) on the counter?
   [question on related object]
6. I asked you to pick up the cup,
   not the pot. [unheeded command]

These examples show differences not only in views but also in the type of conversational interaction: #5 is a question, while #3 and #6 show different aspects of a command-act interaction. Views differ in what actions they refer to, with #1 as the most straightforward case, describing a single action that just occurred. In contrast, #6 refers to two actions, one of which was created earlier in formulating a command that was never performed. Among state views, the most straightforward is to comment on the new value of an object's attribute, as in #2, but it is usually also quite possible to comment on the cessation of the corresponding previous value. Yet another state view is applicable if the new value is the same as the corresponding one for another object; one can then say, for example, that there are two cups on the table or that both cabinet doors are open.

Underlying sentence #3, above, is a plan view, in this case the notion of transition to the next action in the current plan. Plans can also refer to such things as the completion of a plan or subplan and the transition from one subplan to the next. Plans exist in the microworlds so that the successive actions will make sense, not only those chosen by the tutor to carry out itself, but also those the tutor tells the student to do, as in #3. The resulting situational continuity supports a language beginner by keeping it clear what is being talked about. For a more advanced student, plan views provide their own form of variety, including two-clause sentences like, "Now that the pot is full, put it on the stove," in which the first clause involves a state view, the second an action view and the whole sentence comes from a plan view, the transition from a just completed subplan to the next action whose goal is not already satisfied.

[1] Computer Science, George Mason Univ., Fairfax, VA
[2] Institute for Informatics, Bucharest, Rumania

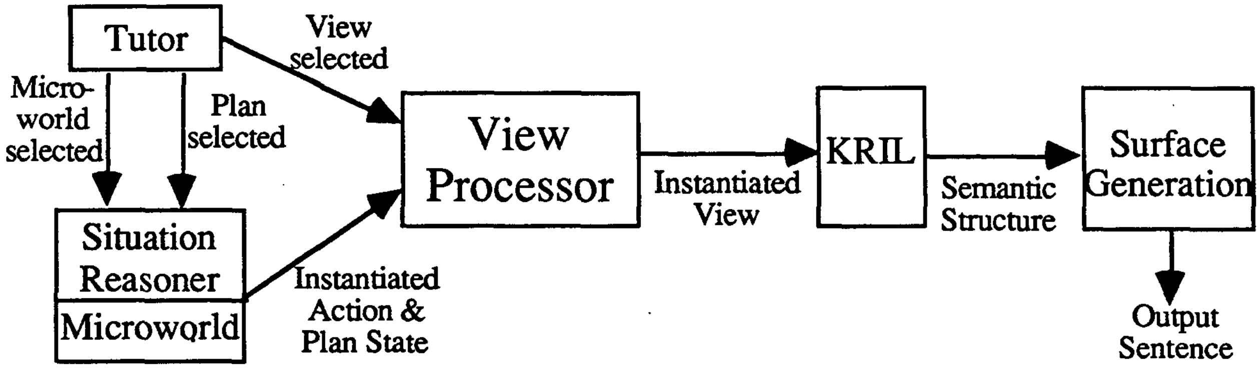

Figure 1. View Processing in context. Organization of modules relevant to language generation.

2 Two-Medium Language Learning

Fluent-2 is a two-medium tutorial system whose principal goal is to provide an essential form of foreign language learning experience: realistic conversation in the target language. Figure 1 shows key parts of the system.

Language interaction in Fluent-2 is tightly integrated with a visual second medium consisting of partially animated graphics under shared control of the student and the electronic tutor. Both the graphics and the language are the outward manifestations of an underlying microworld of objects, in a hierarchy of classes, taking part in actions that are structured into plans. The graphics and animation provide a realistic auxiliary source of information about what is being said. This independent channel helps the student pick up new vocabulary and language constructions in a clear situational context. This two-medium interaction capability, including the deep generation component sketched in this paper, should also be applicable to tutoring systems in other subject matter. Surface generation is done by a large natural language processing system, developed by Susan Felshin of the MIT Athena Language Learning Project (ALLP) and adapted for us. It is this system that takes semantic structures to syntactic structures and ultimately to sentences of English, Spanish and, to a lesser extent, other languages. The natural language processing, graphics, microworlds and tutorial reasoning are all in MCL2 Common Lisp with CLOS on a Mac-IIfx with 20MB of main memory.

The availability of the two media, along with situational continuity, can provide to adults the kind of redundancy that seems essential to children in their race to fluent use of their native tongue. This is not to say that adults learn in the same way as children. Nevertheless, Fluent-2, has been designed with careful attention to successful second language pedagogy and appropriate second language research. Language generation is especially important at the outset, since the learner must comprehend language before meaningfully producing it.

Second language research provides support for using simplified language in meaningful contexts. Three sources of such experience are foreigner talk (by native speakers, to foreigners), motherese (by parents and caregivers, to children) and teacher talk (by teachers, to students). We seek to replicate the benefits of these styles in a computational system, by identifying and adapting specific aspects that underlie their success. Such properties include: restricted vocabulary size; exaggerated intonation and stress; short grammatical sentences; use of concrete references; repetitions, expansions and rephrases; few pro-forms; few contractions; yes-no, choice and tag questions rather than Wh-questions; and so on. (See Hamburger, 1993 for a fuller account).

3 Representing Situations and Actions

The actions in a microworld are of special interest because they constitute an input to view processing, our central concern here. Both actions and plans are implemented as parametrized rules with constrained variables, as in the action rule example in Figure 2. Binding the parameters of an action (or plan) to microworld objects yields an instantiated action (or plan), which can then be carried out, with graphical and internal consequences, and/or forwarded to the generation process. Objects are of various types or classes, with individual properties, some inherited, and relationships to each other. Actions and objects, the bridge between the two media, are chosen partly on the basis of having consequences that are clearly realizable in graphics. Actions are organized into flexible, hierarchical plans that support coherent everyday activity

```
HEADER: (pick-up ($h hand) ($obj physical-object)
    ($from physical-object) ($mw microworld))
GOALS: ((oav $h grasp $obj))
PRECONDS: ((oav $h grasp nothing) ... )
K-RESULTS: ((modify-oav $h grasp $obj)
    (delete-among $from things-on-top $obj) ... )
```

Figure 2. Non-graphics parts of action rule, Pick-Up

Action rules are bi-directional: either the student or the tutor can activate them, depending on the type of interaction. The student does this in a graphically realistic manner, for example by dragging a hand to the faucet and clicking the mouse. The parameters have scope over the whole rule; binding originates in any slot. Information thus can flow among student, tutor and microworld, supporting the two-way, two-medium conversation.

An action rule's Header slot is a key to view processing. It contains the rule name or predicate as well as argument-constraint pairs, and is used in the straightforward view in Figure 3 for a simple description of the action. Also useful in view processing is the K-Results slot, containing object-attribute-value triples for updating the internal situation as a result of the action. State views can select among these results to report various changes. The Goal slot makes it possible, when executing a plan, to skip over any actions and subplans whose goals are already achieved. Besides permitting variety in student action sequences, the satisfied goal can form, via a view, the basis of a useful remark. Views for failed Preconds can also yield comments worth making. Two other action rule slots are for information passed to and from the graphics module; they are not used for views and are omitted from Figure 3..

To see the key role of views, suppose that the student has just made something happen and the system's role is now to make a relevant comment. A simple choice is to say what the student just did, using a representation of the student's preceding microworld action, consisting of an operator with its operands. Just such a representation is in the Header slot of the action rule just triggered by the student (via the graphics input slot). It can be transformed to a semantic structure that is an appropriate input for the surface generation module, which can output the resulting sentence. We do exactly that, but not deterministically. Into this action-to-semantics connection, we use views to insert the possibility of a wide-ranging choice of different approaches to constructing something to say.

4 Views, Levels and Instantiations

A view is an abstraction of what to say and how to say it, expressed as a structure. It guides the view processor in selecting parts of an instantiated action, to instantiate the view. The instantiated view is a language-independent intermediate representation which ultimately yields an output sentence.

The partial example of a view in Figure 3 shows the context level, event level and object level. (Object-level information is not shown.) The event level is central in that it corresponds roughly to the proposition expressed in the main (or only) clause of a sentence. The view type here is 'action', yielding a view that expresses the action itself, without reference to the plan or the resulting state. 'What-action' can be one that has actually occurred, has been talked about or has been constructed for generation. In Figure 3, this choice depends on the interaction type, which also controls the distinction between commands and declarative sentences and the choice of tense.

```
NAME: current-action
CONTEXT: (case interaction-type
                 ((movecaster tourguide) '(recent-past))
                 (antetourguide '(near-future))
                 (commander '(imperative)) )
EVENT: view-type: action
      what-action: (case interaction-type
                          (movecaster 'student-did)
                          (tourguide 'tutor-did)
                          ((antetourguide commander)
                                'tutor-thought))
OBJECT: ...
```

Figure 3. Part of the view, Current-Action.

These observations point to the key role of interaction types within views. Interaction types complement views by organizing the basic conversational move structure. An interaction is a short sequence of specified kinds of linguistic and spatial turns by the tutor and student. Choosing an interaction type determines whether it is the tutor or the student that momentarily takes the initiative. A pedagogically useful interaction type for language learning has at least one linguistic move (is not purely graphical). Either the tutor or the student can start with one of four move types: action, command, question or statement. Following each with its anticipated response yields the eight simplest interaction types.

In the Movecaster type, the student can make any possible move, and the tutor then comments; the tutor asks a question in Quizmaster; it gives a command that the student may act on in Commander; and these roles are reversed in Servant. Tourguide is an interaction type with three moves: an action by the tutor, a description of that action, and acknowledgement by the student. Tourguide can provide initial exposure to a new microworld. Variations of it allow the description to precede or follow the action, or both, giving a basis for variations in tense.

The second move in an interaction should be responsive to the first. Thus some kinds of questions call for a sentence in answer, others a phrase or "yes" or "no". Similarly, actions are expected to be responsive to commands. The tutor may comment about responsiveness to a command or lack of it, using a view-constrained interaction type.

It is now easier to see why, in Figure 3, Movecaster is associated with Student-Did, the student's action, whereas Commander calls for an action - Tutor-Thought - not yet carried out by anyone. What-Action takes four possible values: Student-Did or Tutor-Did for the most recent action executed by the student or tutor; and Tutor-Did or

Tutor-Thought for an action constructed by the tutor as the basis of something already said or about to be said.

State views need two slots at the event-level that are not in event views; see Figure 4. Since an action may result in more than one change in the values of object attributes, state views have an Aspect to specify how to select one of the changes. The selection method in Figure 4 simply takes the first one in the list of updates - reasonable if results are in order of importance. The Pre-Post slot tells whether to use the updated value or the prior one.

VIEW-TYPE:	state
WHAT-ACTION:	last-action
ASPECT:	(position 1)
PRE-POST:	pre

Figure 4. State view, event level: "The cup was in your hand"

Whereas a view tells where to get information, the instantiated view (IV) holds the information itself, which the view processor has for the most part extracted from the instantiated action. For an action view, this is principally the arguments, taken from the action header and placed in IV slots called Agent, Object1, Object2 and Modifier. Under the guidance of the object level of the view, the view processor associates each argument with the correct slot and puts in the contents. Designed for this purpose is the IV-O, or object level of an IV. Each IV slot can be filled by (i) an IV-O, (ii) a microworld object, (iii) a class, which is a language-independent meaning corresponding to a common noun, (iv) a list of items of the three foregoing kinds, or (v) another IV. The latter yields a subordinate clause, whereas each of the others underlies a noun phrase.

Object-level views determine how to express a particular microworld object to convey its relationship to other aspects of a conversation. With a black and a grey cup, for example, after moving the black one, the grey one can be referred to as "the other one," "the grey cup," "the second cup" or even "the cup that is still on the table." In each noun phrase the head noun corresponds by default to the class of the object, unless "one" is included in the specification for that object (giving, in English, the likes of "the red one"). The decision whether to include modifiers (adjectives, relative clauses, and prepositional phrases) may in some cases be expressed by code that includes a method that selects whatever properties are needed to distinguish an entity from others of its class.

The object level may also have information that affects decisions about determiners and possibly quantifiers or pronouns. The choice of determiner can not be specified in isolation by the view, since it must take into account the recent mentions of, and actions on, an entity, for example, "Pick up a (indefinite) box" and then, "Good! You picked it (definite pronoun) up."

Subslot	Possible Values
PRECISION	Top,Parent, Direct, Distinct
REFERENCE	Other, Pronoun, Nil
PROMINENCE	Topic, Wh, Nil

Figure 5. Possible values at Object level

Object-level subslots and their permitted values appear in Figure 5. First comes the degree of Precision with which the object is to be described. It can indicate whether the class for describing the object should be its direct class (e.g., girl, teaspoon), its parent class (e.g., child, spoon) or the highest class permitted by the type constraint for the particular argument of the action rule (e.g., person, thing). Another option is the highest level distinguishing the item from everything else in the current situation. If the item is not alone in the class named, the output needs a modifier or else an indefinite determiner.

If the Reference subslot in a view has the value Other, the item is to be described in terms of other items in its class, e.g., "the other X" or "the rest of the Xs", as opposed to the default case, a description of an object by its own properties. The Prominence subslot specifies whether its object should be made prominent or not, and if so, whether by topicalization - Topic - or by being questioned with a Wh word.

Acknowledgement. This work is supported under grant IRI-9020711 from the National Science Foundation.

References

Delin, J., Scott, D. and Hartley, T. (1993) Knowledge, intention, rhetoric. In O. Rambow (Ed.) Intentionality and Structure in Discourse Relations. Morristown, NJ: Association for Computational Linguistics.

Felshin, S. (1993) The Lingo Manual. Cambridge, MA: Lab. for Advanced Technology in the Humanities, MIT.

Hamburger, H. (in press) Tutorial tools for language learning by two-medium dialog. In M. Holland, J. Kaplan and M. Sams (Eds.) Hillsdale, NJ: Lawrence Erlbaum Associates.

Hamburger, H. (1993) SCIALogie and Fluent: Pedagogy and microworlds for language immersion and tutoring. In T. Chanier, D. Renie and C. Fouquere (Eds.) Actes du Colloque SCIAL. Clermont-Ferrand, France.

Hovy, E. (1988) Generating Natural Language under Pragmatic Constraints. Hillsdale, NJ: Lawrence Erlbaum Associates.

Meteer, M. (forthcoming) Text planning and text structuring. Computational Linguistics.