FREE ADJUNCTS IN NATURAL LANGUAGE INSTRUCTIONS*

Bonnie Lynn Webber   Barbara Di Eugenio
Department of Computer and Information Science
University of Pennsylvania
Philadelphia PA 19104-6389

ABSTRACT

In this paper, we give a brief account of our project *Animation from Instructions*, the view of instructions it reflects, and the semantics of one construction – the *free adjunct* – that is common in Natural Language instructions.

1 Introduction

Over the past few years, we have been developing a system for creating *animated simulations* from Natural Language instructions. When the system is complete, these animated simulations will combine:

- *animated agents* which demonstrate the instructions being carried out in a specified environment;
- Natural Language *narration* which explains what is being done and why.

Such narrated simulations can then be used in instructing agents of varying capacities in how to perform tasks with varying demands in workplaces of varying layout.

In [2], we argue that the only way to create such narrated simulations is to drive *both* animation and narration from a *common representation* that embodies the same conceptualization of tasks and actions as Natural Language itself.[1] We also argue the difficulty of hand-tooling such a representation for each task to be demonstrated and explained. Instead, we argue for enabling a system to create these representations for itself, from *Natural Language Instructions*. In fact, we make the stronger claim that creating task animation from anything but direct graphical manipulation *forces* one to Natural Language as the only instruction source accessible to users other than manually skilled (or programming-wise) animators.

Creating task animations from Natural Language instructions forces us to understand instructions computationally. Instructions as a type of text have not been studied as much as narratives as a way of describing tasks, but it is clear that they differ: when a *narrative* describes a task, it tells what happened when the task was performed in a particular circumstance. *Instructions*, on the other hand, commonly specify how to perform the task in a wide range of circumstances that may change during the course of performance in quite different ways. This has at least two consequences: (1) to understand instructions, one has to understand how instructions relate to intended behavior, and (2) in processing instructions, one has to deal with constructions that either only rarely appear in narrative or play different roles than they do in narrative.

In this paper, we start by presenting what we take to be the relationship between instructions and behavior, and then explore one construction often found in instructions – *free adjuncts* – explaining them in light of this relationship.

2 Instructions

Our view of *instructions* derives from a view of *plans* variously advocated in Pollack [7, 8], Suchman [11], and Agre and Chapman [1].

Pollack contrasts two views of plan: *plan as data structure* and *plan as mental phenomenon*. (The former appears to be the same view of plans that Agre and Chapman have called *plan as program*.) Plans produced by Sacerdoti's NOAH system [9] are a clear example of this *plan as data structure* view. Given a goal to achieve (i.e., a partial state description), NOAH uses its knowledge of actions to create a data structure (a directed acyclic graph) whose nodes represent goals or actions and whose arcs represent temporal ordering, elaboration, or entailment relations between nodes. This data structure represents NOAH's *plan* to achieve the given goal.

As Suchman points out [11], NOAH's original intent was to provide support for novice human agents in carrying out their tasks. Given a goal that an apprentice was tasked with achieving, NOAH was meant to form a plan and then use it to direct the apprentice in what to do next. To do this, it was meant to generate a Natural Language instruction corresponding to the action associated with the "current" node of the graph. If the apprentice indicated that he didn't understand the

*We thank Mark Steedman, Hans Karlgren and Breck Baldwin for comments and advice. They are not to blame for any errors in the translation of their advice into the present form. The research was supported by DARPA grant no. N0014-85-K0018, and ARO grant no. DAAL03-89-C0031.

[1]This is not to suggest that animation can be driven *solely* from that common representation: other types of knowledge are clearly needed as well – including knowledge of motor skills and other performance characteristics.

instruction or couldn't perform the prescribed action, NOAH was meant to "move down" the graph to direct the apprentice through the more basic actions whose performance would entail that of the original. The result is a sequence of *instructions* that corresponds directly to the sequence of nodes encountered on a particular graph traversal.

Pollack contrasts the above with a *plan as mental phenomenon* view, in which having a plan to do some action β corresponds roughly to

- a constellation of beliefs about actions and their relationships;
- beliefs that their performance, possibly in some constrained order, both entails the performance of β and plays some role in its performance;
- an *intention* on the part of the agent to act in accordance with those beliefs in order to perform β.

With respect to such beliefs, Pollack draws a three-way distinction between *act-types*, *actions* (or acts) and *occurrences*. *Act-types* are, intuitively, types of actions like playing a chord, playing a D-major chord, playing a chord on a guitar, etc. Act-types, as these examples show, can be more or less abstract. *Actions* can be thought of as triples of *act-types*, *agents*, and *times* (relative or absolute intervals) like Mark playing a D-major chord last Sunday afternoon on his Epiphone. Because it is useful to distinguish an action from its occurrence in order to talk about intentions to act that may never be realized, Pollack introduces a separate ontological type *occurrence* that corresponds to the realization of an action. (Pollack represents an *occurrence* as OCCUR(β), where β is an *action*. Thus an *occurrence* inherits its time from the associated time of its argument.)

Agents can hold beliefs about entities of *any* of these three types:

- *act-types* – An agent may believe that playing a D-major chord involves playing three notes (D,F♯ and A) simultaneously, or that s/he does not know how to perform the act-type playing a D-major chord on a guitar, etc. Any or all of these beliefs can, of course, be wrong.
- *actions* – An agent may believe that some action α_1 must be performed before some other action α_2 in order to do action β_1 or that α_2 must be performed before α_1 in order to do β_2. Here too, the agent's beliefs can be wrong. (It was to allow for such errors in beliefs and the Natural Language questions they could lead to that led Pollack to this *Plan as Mental Phenomenon* approach.)
- *occurrences* – An agent may believe that what put the cat to sleep last Sunday afternoon was an overdose of catnip. S/he may also have misconceptions about what has happened.

Therefore one can take the view that instructions are given to an agent in order that s/he develops appropriate beliefs, which s/he may then draw upon in attempting to "do β". Depending on the evolving circumstances, different beliefs may become salient. This appears to be involved in what Agre and Chapman [1] and what Suchman [11] mean by *using plans as a resource*. Beliefs are a resource an agent can draw upon in deciding what to do next.

Given this view of *plan as mental phenomenon*, we can now consider possible relationships between *instructions* and *behavior*. At one extreme is a *direct* relationship, as in the game "Simon Says", where each command ("Simon says put your hands on your ears") is meant to evoke particular behavior on the part of the player. That is,

Instruction ⇒ Behavior

The fact that such instructions are given in Natural Language is almost irrelevant. We have already demonstrated [4] that they can be used to drive animated simulations. Key frames from such a demonstration of two agents (John and Jane) at a control panel following instructions that begin

> John, look at switch twf-1.
> John, turn twf-1 to state 4.
> Jane, look at twf-3.
> Jane, look at tglJ-1.
> Jane, turn tglJ-1 on.

are shown in Figure 1.

In contrast, instructions can depart from this simple direct relation in many ways:

1. *Multiple clauses may be involved in specifying the scope or manner of an intended action.* For example, the intended *culmination* of an action may not be what is intrinsic to that action, but rather what is taken to be the start of the action prescribed next.[2] Consider the following instructions that Agre [1] gave to several friends for getting to the Washington Street Subway Station.

> Left out the door, down to the end of the street, cross straight over Essex then left up the hill, take the first right and it'll be on your left.

While the action description "[go] left up the hill" has an intrinsic culmination (i.e., when the agent gets to the top of the hill), it is not the intended termination of the action in the context of these instructions. Its intended termination is the point at which the action of "taking the first right" commences – that is, when the agent recognizes that s/he has reached the first right. In Section 3, we will provide many more examples of this feature of instructions.

2. *Instructions may describe a range of behavior appropriate under different circumstances.* The agent is

[2] This is not the case in "Simon Says" type instructions, where each action description contains an intrinsic culmination [6].

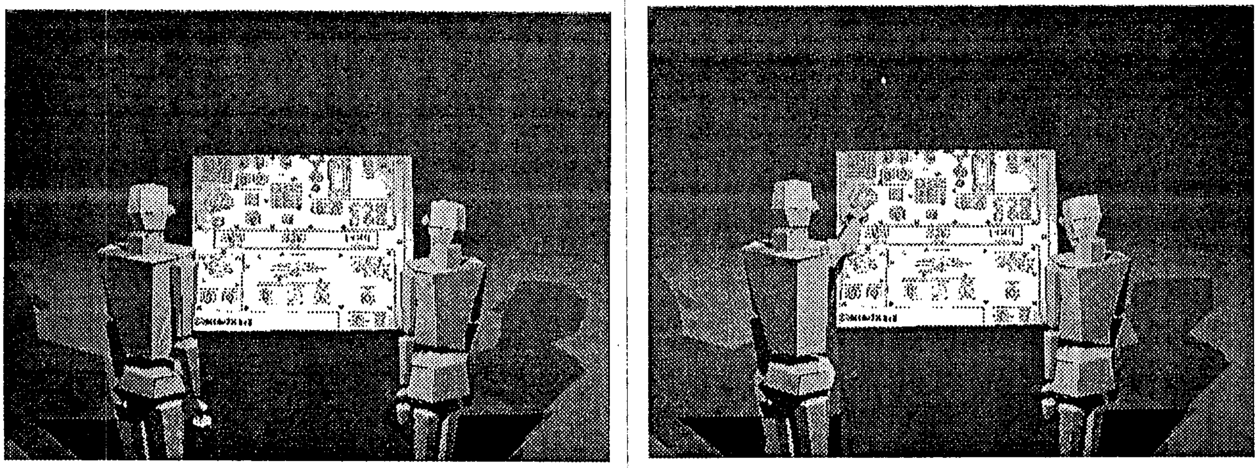

Figure 1: Control Panel Animation

only meant to do that which s/he recognizes the situation as demanding during its performance. For example, the following are part of instructions for installing a diverter spout:

> Diverter spout is provided with insert for 1/2" pipe threads. If supply pipe is larger (3/4"), unscrew insert and use spout without it.

Here, the relevant situational features can be determined prior to installing the spout. In other cases, they may only be evident during performance. For example, the following are part of instructions for filling holes in plaster over wood lath:

> If a third coat is necessary, use prepared joint compound from a hardware store.

Here, the agent will not know if a third coat is necessary until s/he sees whether the first two coats have produced a smooth level surface.

3. As in the *plan as data structure* model, *instructions may delineate actions at several levels of detail or in several ways.* For example, the following are part of instructions for filling holes in plaster where the lath has disintegrated as well as the plaster:

> Clear away loose plaster. Make a new lath backing with metal lath, hardware cloth, or, for small holes, screen. Cut the mesh in a rectangle or square larger than the hole. Thread a 4- to 5- inch length of heavy twine through the center of the mesh. Knot the ends together. Slip the new lath patch into the hole ...

Here the second utterance prescribes an action at a gross level, with subsequent utterances specifying it in more detail.

4. *Instructions may only provide circumstantial constraints on behavior* but not specify *when* those circumstances will arise. For example, the following comes from instructions for installing wood paneling:

> When you have to cut a sheet [of paneling], try to produce as smooth an edge as possible. If you're using a handsaw, saw from the face side; if you're using a power saw, saw from the back side. Otherwise you'll produce ragged edges on the face because a handsaw cuts down and a power saw cuts up.

Such cases as these illustrate an *indirect* relation between instructions and behavior through the intermediary of an agent's beliefs and evolving plan. That is,

Instructions ⇒ Beliefs ⇔ Plan ⇔ Behavior

3 Free Adjuncts

In the previous section, we noted that multiple clauses may be involved in specifying an intended action, using this as evidence for our view of an indirect relationship between instructions and behavior. Here, we discuss one multiple-clause construct in more detail – the **free adjunct** – since it also provides evidence for our claim that the representation driving *narrated animations* should embody the same conceptualization of tasks, actions and events as Natural Language itself.

A *free adjunct* is defined as *a nonfinite predicative phrase with the function of an adverbial subordinate clause* [10]. It may be headed by a noun, adjective, prepositional phrase, or verb[3]. Here we focus on free adjuncts headed by progressive gerundives, as they are quite common in instructions – e.g., the underlined clause in Ex. 1:

Ex. 1 *Pour mixture over cheese in casserole, <u>spreading evenly</u>.*

Stump notes of free adjuncts that *their logical connection with the clause they modify is not overtly specified* [10][4]. Here we argue that (1) instructions exploit three

[3] Constructions headed by subordinating conjunctions and containing a nonfinite verb, such as *while fighting in France, he was taken prisoner* are not considered to be free adjuncts by Stump [10], who calls them *augmented adjuncts.*

[4] Free adjuncts are just one kind of a larger class of syntactic forms, *absolute constructions*, that have this property: for a more thorough discussion, see [10].

logical connections between a gerundive adjunct and its matrix clause; and (2) to represent these relations requires a representation with a temporal ontology at least as rich as that proposed in [6], as well as support for *generation* relations [5] (defined below) and *abstraction*. We conclude by showing that the role adjuncts play in instructions differs from the role they play in narratives.

3.1 Data Analysis

We collected 97 consecutive instances of gerundive adjuncts (here called simply "adjuncts") in instructions.[5] The syntactic structure of sentences containing these adjuncts is generally limited to a main clause, preceded and/or followed by an adjunct. The main clause describes an action, which we call α_{main}; μ_{adj} will refer to the semantic content of the adjunct. We found that our corpus divided into three classes, depending on the logical connection between the adjunct and α_{main}:

1. it may *augment* the description of α_{main};
2. it may describe a second action α_{adj} that *generates* or is generated by α_{main};
3. it may describe an independent action α_{adj} that should be performed simultaneously with α_{main}.

It is important to remember, in the following discussion, that (following Pollack [7, 8]) an *action*, like an *act-type*, is a *description*, not something in the world.

3.1.1 Augmentation

About half the adjuncts in our corpus supply features of α_{main}, such as its starting point; necessary tool(s) or material(s); objects that α_{main} may create, etc. Thus, α_{main} is a less specific version (i.e., an *abstraction*) of the intended action α that results from combining α_{main} and μ_{adj}. For example, in Ex 2, the adjunct specifies the tool to use:

Ex. 2 *Using a coping or back saw, carefully cut all pieces to the dimensions given in the materials list.*

Alternatively, the adjunct can provide features of the world that have to either remain or become true after executing α_{main}.

Ex. 3 *Sew the head front to back, leaving the neck edge open.*

The adjunct can alternatively specify a constraint on the execution of α_{main}, including:

- a *manner constraint*, that α_{main} be executed in such a way that a state is brought about which continues to hold during its execution. In the following example, while the agent executes the cutting action, s/he has to *stay to the outside of the line*:

Ex. 4 *Transfer pattern to top back board A and using a jig or a scroll saw carefully cut out pattern staying to the outside of the line.*

- a *side-effect constraint*, that a possible side effect of α_{main} should be avoided. Verbs like *take care*, *be careful*, *make sure* etc. followed by *not to ...*, are often used:

Ex. 5 *Cut under eaves of cabin with chisel, being careful not to chip roof.*

The need to represent the result of augmentation and the relation between α_{main} and α is one reason for requiring our system to have a representational capacity at least rich enough to represent partial descriptions of actions and an *abstraction* relation between them.

Partial description is not meant to imply partial with respect to some fully specified description. On the contrary, we do not assume that there is an *a priori* fixed set of features belonging to an action. To say that an adjunct conveys additional features of α_{main}, does not mean that one can specify beforehand what all those features might be.

To a first approximation, the relation between descriptions could be stated in terms of the amount of information that a description conveys. Note that this does not have to be new information: in Ex 2, the information conveyed to an expert carpenter by the adjunct is probably redundant, given that he knows what kinds of saws to use.

3.1.2 Generation

Goldman [5] defines *generation* as that relation between actions informally conveyed by the preposition by in expressions such as "agent G does β by doing γ" – e.g., "John turns on the light by flipping the switch". Free adjuncts can specify a generation relation between actions α_{main} and α_{adj} in either direction, without an overt by – for example,

Ex. 6 *As you work, clean the surface thoroughly each time you change grits, vacuuming off all the dust and wiping the wood with a rag dampened with turpentine or paint thinner.*

[α_{adj} GEN α_{main}]

Ex. 7 *Cut one 7x7-inch square from foil. Fold corners to center of square; cut in half on the diagonal creating two triangles.*

[α_{main} GEN α_{adj}]

Ex. 8 *Sew bottom canvas bag to bottom of front and back, making a long rectangle.*

[α_{main} GEN α_{adj}]

[5] Data were collected from five magazines – two of which describe wood projects, and the other three, "crafts" – and one chapter of a "how to" book on installing wall coverings.

In the case of generation, only one action is executed *per se*, generating the other as a result.

One natural question to ask is why two different descriptions are given of the same action. The reasons are the same as in any text: to make explicit the *purpose* of an action or a *salient feature*. For example, in Ex. 6, *clean* provides a unifying description for the two actions expressed in the adjuncts, and by doing so, indicates their purpose. In Ex. 7, the result of α_{main} (the two triangles) is mentioned explicitly, in order to introduce these new referents in the agent's discourse model. In Ex. 8, the description *a long rectangle* provides a visual clue to the result to be achieved. (This may be an additional purpose for the generate relation in Ex. 7 as well.)

Again, Ex. 6 shows the need for abstraction in our representation, in the form of one or more abstraction hierarchies of action descriptions: to understand this example, we need to know that both *vacuum* and *wipe* are specializations of *clean*.

3.1.3 Simultaneity

If the adjunct describes an action α_{adj} that is independent of α_{main}, it is meant that both are to be executed simultaneously:

Ex. 9 *Soak paper in water for 1 hour; remove paper, then smooth onto board, squeezing out excess water and air bubbles. Staple paper to board along the edges. Mix rose madder and water; pour onto wet paper, tilting board to spread color.*

Ex. 10 *Unroll each strip onto the wall, smoothing the foil into place vertically (not side to side) to avoid warping and curling at the edges.*

3.2 Aspect and Event Structure

Earlier, we claimed that the representation driving *narrated animations* should embody the same conceptualization of tasks, actions and events as Natural Language itself. We take the conceptualization of actions and events to be the tripartite event structure described by Moens and Steedman (hereafter, M&S) in [6].

The goal in [6] is to provide a single explanation of aspectual profiles, of changes in aspectual profile related to the use of adverbial and prepositional modifiers, and of the purported temporal "ambiguity" of when-clauses. The explanation makes use of a tripartite event structure which M&S call a *nucleus*. A nucleus consists of a *preparatory process*, a *culmination* and a *consequent state*. Within this framework, an event description interpreted as a PROCESS corresponds simply to a preparatory process, while a CULMINATED PROCESS corresponds to an entire nucleus. CULMINATIONS (Vendler's *achievements* [12]) correspond to a culmination followed by its consequent state.

Within this framework, M&S attribute changes in aspectual profile brought about by modifiers (viewed as *functions* from event description to event description) to two factors: (1) The modifier, viewed as a function, may have a different output type than its input type. The modified form will thus have the same aspectual type as the function's output. (2) When a function demands a different aspectual type for its input than it has been given, a mechanism called *coercion* maps the input to the needed type. This may change semantic features of the input, *before* function application.

What we shall show here (rather briefly) is that this same tripartite *nucleus* can ground the possible interpretations of *augmentation* (Section 3.1.1) and *simultaneity* (Section 3.1.3), and in fact, account for ambiguities in interpretation. We start with the following minimal pair:

Ex. 11 *Starting with this mark, make another mark, leaving exactly 2 inches between marks.*

Ex. 12 *Starting with this mark, make a series of marks, leaving exactly 2 inches between marks.*

In M&S's framework, making a (single) mark (Example 11) could be considered a CULMINATION. The plural "series of marks" in Example 12 would then map this singular interpretation to a CULMINATED PROCESS through *iterating* mark-making. (Iterated mark-making culminates when there is no more room to make marks.) The augmentation in Example 11 constrains the distance between the single pair of marks, that in Example 12, the distance between each pair of marks produced during the iteration.

Now consider the following example of *simultaneity*:

Ex. 13 *Wire vines together at one end. Twine vines into an 8-inch diameter wreath, fastening with wire to hold.*

The second sentence mentions two independent actions - twining the vines into a wreath (α_{main}) and fastening (α_{adj}). In M&S's framework, the action α_{main} can be taken to be a CULMINATED PROCESS in two different ways: a single homogeneous twining process, which culminates when one has used up all the vines, or (as above) an iteration of individual twinings, culminating for the same reason. In the first case, fastening happens at the single point of culmination – its purpose being to prevent the two ends of the wreath from coming apart. In the second, fastening happens at the end of each iteration – its purpose being to keep the strands together. To capture both these interpretations (and decide between them) requires a representation such as M&S's rich enough to capture the required event structure.

3.3 Relation to Previous Work

The most extensive analysis of the semantics of free adjuncts (in English) that we are aware of is that done by Greg Stump [10]. However, all his examples come from narrative text, and as a result, he focusses on their truth-conditional properties. For example, he draws a distinction between *strong* and *weak* adjuncts:

Ex. 14 *a) Having unusually long arms, John can touch the ceiling.*
b) Standing on the chair, John can touch the ceiling.

Ex. 15 *a) Being a businessman, Bill smokes cigars.*
b) Lying on the beach, Bill smokes cigars.

Stump calls the adjuncts in both *a* sentences *strong*, because their actual truth is uniformly entailed. He calls those in the *b* sentences *weak*, because their actual truth can fail to be entailed.

Related to this, Stump also notes a causal flavor in strong adjuncts. Consider the adjuncts in the *a* sentences in both Exs. 14 and 15. The sense is that in both cases, the main clause assertion is true *because* the adjunct is. Weak adjuncts, on the other hand, have a conditional sense: it is (only) when the condition described in the adjunct is true that the main clause assertion is true.

While these observations appear to be both correct and relevant in narratives, this strong/weak distinction appears irrelevant for instructions, which do not concern themselves with truth conditions in the same way as narratives. The only thing in instructions that comes close to the conditional sense of weak adjuncts is the perfective gerundive adjunct, as in

Ex. 16 *Having basted the seams, check again for fit.*

Such adjuncts do convey a similar sense that it (only) when the action described in the adjunct is complete that the main clause command is relevant.

In Section 3.1, we essentially tried to show that in instructions, gerundive adjuncts play a role in further specifying intended action. They may do this through *augmenting* α_{main}, through providing an alternative description of α_{main} through *generation*, or through specifying another (independent) action that must be performed *simultaneously* with α_{main}in some way. Thus we conclude that gerundive adjuncts (if not all free adjuncts) play a different role in instructions than they do in narrative text. This emphasizes the importance of analysing constructions *in situ*, rather than assuming that conclusions based on narrative text will hold equally of instructions.

4 Summary

In this paper, we have given a brief account of our project *Animation from Instructions*, the view of instructions it reflects, and the semantics of one particular construction that occurs often in the type of instructions we will be handling. The project is proceeding on several fronts, including the following: (1) Similar analyses are being done of other constructions that commonly occur in instructions [3]; (2) we are starting to develop a representation that embodies both the temporal ontology [6] that grounds the semantics of these constructions and an abstraction mechanism - notice that when we talk about abstraction we do not limit ourselves to abstraction hierarchies: we intend abstraction as a general relation between more and less specific descriptions of actions; and (3) translation processes are being expanded for mapping that representation into forms that our simulation system [4] can deal with. More detailed description of the system as a whole is given in [2].

References

[1] Phillip Agre and David Chapman. What are Plans For? A.I. Memo 1050a, Artificial Intelligence Laboratory, MIT, October 1989.

[2] Norman Badler, Bonnie Webber, Jeff Esakov and Jugal Kalita. Animation from Instructions. *Making Them Move: Mechanics, Control and Animation of Articulated Figures.* Morgan-Kaufmann, 1990.

[3] Barbara Di Eugenio and Bonnie Webber. Action Specifications in Natural Language Instructions. Technical Report, Dept. of Computer & Information Science, University of Pennsylvania, Philadelphia PA. Forthcoming.

[4] Jeffrey Esakov and Norman I. Badler. An Architecture for Human Task Animation Control. In *Knowledge-Based Simulation: Methodology and Applications* P.A. Fishwick and R.S. Modjeski (eds.), Springer Verlag, New York, 1989.

[5] Alvin Goldman. *A Theory of Human Action.* New York: Prentice-Hall, 1970.

[6] Marc Moens and Mark Steedman. Temporal Ontology and Temporal Reference. *Computational Linguistics.* 14(2), 1988, pp. 15-28.

[7] Martha Pollack. *Inferring Domain Plans in Question-Answering.* PhD Thesis, Dept. of Computer and Information Science, University of Pennsylvania, Philadelphia PA. (Available as Technical Report MS-CIS-86-40, University of Pennsylvania, May 1986.)

[8] Martha Pollack. Plans as complex mental attitudes. In *Intentions in Communication*, J. M. P. Cohen and M. Pollack, Eds., MIT Press, 1990.

[9] Earl Sacerdoti. *A Structure for Plans and Behavior* Elsevier, New York, 1977.

[10] Greg Stump. *The Semantic Variability of Absolute Constructions.* Dordrecht: D. Reidel, 1985.

[11] Lucy Suchman. *Plans and Situated Actions: The problem of human machine communication.* Cambridge University Press, 1987.

[12] Zeno Vendler. *Linguistics and Philosophy.* Ithaca NY: Cornell University Press, 1967.