"Do Nothing Till You Hear from Me": Composing Processes with Termination Conditions

Bonnie Webber
Department of Computer & Information Science
University of Pennsylvania
Philadelphia PA 19104-6389
bonnie@central.cis.upenn.edu*

Introduction

Embodied agents act in a world and derive perceptions from it. One thing that perceiving the world enables such an agent to do is decide when s/he can stop what s/he is doing and get on to something else.

In terms of Natural Language (NL) instructions that specify action, the termination condition an agent is meant to be checking for may be derivable from the verb and its arguments (a.k.a. "logical form" or "predicate-argument structure"), as in

(1) Close the door.
⇒ The agent can stop moving the door when s/he has seen/heard/felt it is closed.

(1) Put the books in the freezer.
⇒ The agent can stop picking up book(s) and placing them in the freezer when s/he sees they are all inside.

Often in NL instructions, however, the termination condition an agent must check for can only be derived from some clausal adjunct, as in the case of instructions containing "until" clauses:

(Do) α **until** κ

These instruct an agent to engage in the *process* specified by α (where a process is an action with no *intrinsic* termination condition), terminating α when the condition specified by κ is determined to hold. Perception is critical to that determination. For example,

(1) Lay the first brick on the mortar bed. Press it down **until** the mortar is about 3/8 inch thick.

This requires the agent to monitor the amount of mortar underneath the brick she's pressing down, stopping both pressing and monitoring only when she determines that only 3/8 inch of mortar remains under the brick.

(This does not exhaust the source of termination conditions in NL instructions. As we show in [6], specifying the *purpose* of some action may indirectly convey the conditions for terminating it (i.e., when that purpose is achieved). The termination condition for an action can also be conveyed directly or indirectly in a separate sentence – e.g., "Saute the onions over a low flame. When they have wilted, add the chopped garlic." Specifying the additional discourse-level machinery needed to handle such examples is quite beyond the scope of this short paper.)

A group of faculty and students associated with Penn's Center for Human Modeling and Simulation is exploring the use of Natural Language and other high-level task specifications to create realistic animated simulations of virtual human agents carrying out tasks. The value in using a high-level task specification is that the same specification can be used to produce appropriate agent behavior in a variety of different environments and/or different conditions, without additional animator or programmer intervention. Such simulations can thus afford a relatively inexpensive way to carry out human factors studies in computer-aided design, multi-agent training, or even studies of Natural Language understanding.

Using NL instructions to guide the behavior of animated agents requires parsing and mapping instructions to a logical form, and then mapping that structure into a representation that can drive the behavioral simulation that forms the basis for animation. While instructions with "until" clauses present no major problem for parsing, mapping them to a representation that will produce realistic behavior requires understanding the behavioral contributions of both α and κ and how they are commonly integrated. That is, a realistic rendition of "(Do) α until κ is not simply a matter of doing α for some time until κ becomes true: separate actions needed to asses κ may have to be interleaved at appropriate points during the performance of α, which may in turn depend on the relationship between α and κ. My point here is that composing a process with a termination condition requires reasoning. My hope here is to elucidate some of that reasoning.

The data on which this paper is based are drawn from six chapters of two volumes of home repair instructions scanned in by Joseph Rosenzweig, a graduate student at the University of Pennsylvania: Dorling Kindersley's *Home Repair Encyclopedia* [8] and the *Reader's Digest*

*The author would like to thank Joseph Rosenzweig and Mark Steedman, for comments on earlier drafts of the paper. Title courtesy of Duke Ellington. This research is partially supported by ARO DAAH04-94-G-0426 DMSO DAAH04-94-G-0402 and ARPA.

New Complete Do-It-Yourself Manual [11]. The data consist of 80 instructions containing "until clauses", which provide clear examples of activities contingent on monitored conditions.

Some of the instructions concern repair jobs (e.g. fixing broken china, repairing cracked parquet, etc.), and the others concern construction of concrete, asphalt, and/or masonry structures. (The chapters were chosen randomly, not because of their subject matter, and all sentences containing "until" clauses were extracted from them.) While these instructions do not come from face-to-face engagements, and while there are well-known differences between written and spoken language, I believe they do reflect agents' physical presence in on-going activities and an awareness of the perceptual stimuli that should be available to them. I would therefore claim that these instructions represent suggestive evidence of language facilitating extended agent interaction with the world, which is the subject of the workshop.

The agent architecture I am assuming contains at the very least:

- one or more low-level Sense-Control-Act (S-C-A) loops, that can be invoked, modified and/or killed by
- a process-based (as opposed to state-space) control structure (cf. [10]). The one we have begun to use in much of our animated simulation work is called a parallel transition network (PaT-Net) [1, 2]. Nodes in a PaT-Net correspond to processes. Arc transitions occur when (1) a particular broadcast signal is received; (2) a specific feature of the environment is found to have a particular value when polled; or (3) an instrinsic culmination point is reached.

With respect to such an architecture, instructions with "until" clauses would be interpreted as process-based representational structures that set the S-C-A loops and interpret both their success and error conditions.

κ's Contribution to Agent Behavior

The first thing to note is that perception alone may be insufficient to determine whether a condition holds: one or more actions may first be necessary to bring the world into a state in which an appropriate observation can be made. Such actions Kirsh and Maglio [7] have called *epistemic*.[1] While no epistemic action may be required in

(2) Squeeze riveter handles **until** rivet stem breaks off.

to be able to observe the rivet stem breaking off, to determine whether condition κ holds in

(3) Wait for the filler to set and rub it down, first with a needle file and then with glasspaper, **until** it lies flush with the surface.

[1] Kirsh and Maglio use the term *pragmatic* action for ones whose purpose is to bring an agent closer to her goal.

the agent must use some form of tactile perception. This will likely force her to stop rubbing the filler with glasspaper and feel the filler-surface area with her fingertip(s).

Where the agent is assumed to know how to detect κ, no explicit guidance is given. Where the agent is not assumed to know how to detect κ, she may be told how to do so, in terms of relevant epistemic actions and directly preceivable condition. For example, the water-changing instructions below provide guidance in determining whether water contains salt.

(4) Change the water daily **until** all the salts have gone. To test this, hold a spoonful of the water over a flame so that the water evaporates. There should be no salts left.

One reason for providing explicit guidance is that the epistemic and perceptual actions can have undesirable side effects. To avoid them, an alternative procedure may be specified in the instructions, although perhaps not necessarily in the clearest way – e.g.

(5) Leave this glaze for a short time **until** it becomes "tacky" (a test strip on an old tile will indicate when it is ready).

In the end though, it is possible that the specified condition cannot be directly perceived and that no procedure for determining it is provided – e.g.

(6) Mix the powders a little at a time **until** the proportions look right, ...

The agent is then left to her own devices.

An interesting case is where the condition to be tested for is the agent's ability to perform the next action in the sequence. While the condition may be tested several times and found not to hold, when it *is* found to hold, the next action has effectively been performed – e.g.

(7) Chip brick with chisel **until** it can be removed.

(8) After loosening stone with pick and shovel, pry it out with one 2x4, then with the other, **until** you can use one of the levers as a ramp to get stone out of hole.

There may, of course, be several ways to assess a condition, and with further experience, an agent may change which one she uses. So in the earlier brick-laying example (repeated here)

(9) Press it down **until** the mortar is about 3/8 inch thick.

an inexperienced agent may have to interrupt her pressing to measure with a ruler the amount of the mortar still remaining beneath the brick. With experience, the agent may learn to simply eyeball thickness. In creating realistic animations, we can have our agents' skills reflect any degree of experience, as long as it is clear what they are supposed to represent.

α's Contribution to Agent Behavior

As noted earlier, in instructions of the form (Do) α **until** κ, α must be interpretable as a *process* in Moens

and Steedman's terminology [9] – that is, a temporally-extended action with no intrinsic culmination point. If α cannot be directly interpreted as a process, it must be coerced into such an interpretation. Moens and Steedman, for example, note how "for phrases" such as "for five minutes", can coerce what they term an *culminated process* – i.e., a temporally-extended action with a culmination point – into a process either through iteration of the basic action or through loss of its intrinsic culmination, as in:

(10) Play the Moonlight Sonata for 1 minute.

(11) Play the Moonlight Sonata for 1 day.

In the first case, the intrinsic culmination point is lost (one stops after a minute, not when one reaches the end of the piece), and in the second, playing the sonata must be repeated until it fills the whole day.

The first thing to note in interpreting instructions with "until" clauses, is that coercions such as the above can help to determine what the agent is supposed to be doing and what its relationship is to the condition to be assessed. In the most straightforward case, α is the process that affects the world either *cumulatively* until κ is the case

(12) Squeeze riveter handles **until** rivet stem breaks off.

or *nondeterministically* until κ is the case

(13) Try sample specks on the piece **until** you get a get a good match, wiping them away each time **until** you find the right colour.

As the condition-effecting process, α may either be a *simple process* such as in the "squeeze" example above or in

(14) Rotate the plate **until** the guide fingers touch the rod lightly

or what Moens and Steedman call an *iterated process*

(15) Strike set with fat end of hammer **until** rivet head is rounded off.

(16) Fill in low spots and strike off again **until** concrete is level with the top of the form.

On the other hand, when it is an *independent* process that affects the world either cumulatively or nondeterministically, the agent may not be responsible for doing anything other than actions needed to assess the specified condition κ.

(17) Let poultice stand **until** it dries.

(18) Stop work and wait **until** the water evaporates and the concrete stiffens slightly.

The independent process that produces the specified condition is usually one that has been initiated by a previous action taken by the agent. If the process is *cumulative*, the condition to be assessed may either be its *end stage*, as in example 17, or an *intermediate stage*, as in example 18, where the process must be interrupted, lest the concrete harden completely. (I speculate that this independent process could be non-deterministic, but I do not have any examples as evidence.)

An independent process may also be involved in producing the specified condition when the agent is herself engaged in a non-wait process – e.g.,

(19) Place the article in a plastic container and add distilled water .... Change the water daily **until** all the salts have gone.

(20) Heat larger pieces first with a broad flame, otherwise they may distort. Heat the joint in the centre **until** it is red hot.

The existence of an independent process can also affect what will happen if the agent stops her non-wait process – say to check whether the condition holds.

In example 19, the agent's action of changing the water *enables* the process of drawing salt out of the article to continue. If the amount of salt in the water and on the surface of the article are in equilibrium, the process will stop on its own accord. Thus, if the agent fails to act, the specified condition "all the salts have gone" will never be achieved. The agent's action provides, in a sense, the resources needed for the process to continue.

In example 20, on the other hand, the agent is not providing additional resources through her action but rather *maintaining* the existing situation, which in turns *enables* the heating process to continue. If the agent stops her maintenance action – e.g., to check whether the center joint is red hot – the joint will start to cool.

I noted above two forms of coercion from an activity with a culmination point to the *process* against which an "until" clause can be interpreted. I noted such coercions help to determine what the agent is meant to be doing. Here I want to suggest a third type of coercion. While I pose it as an alternative to the analysis given by Moens and Steedman in [9], it adheres to their basic event ontology and thus provides additional evidence for it. The suggestion is motivated by the following example:

(21) If solder gets runny or if iron smokes, turn off iron **until** it cools a bit.

I think it is obvious that what the agent is *meant* to do is to turn the soldering iron off (at which point it will start to cool) and then wait some amount of time until the iron is cooler and has stopped smoking. The question is what that interpretation derives from.[2]

Turning off an appliance is a *culmination* in Moens and Steedman's terminology, an activity that gives rise to a change in the world but that a speaker views as happening instantaneously. Moens and Steedman note that a "for" adverbial (which, like an "until" clause, requires a process) in combination with a culmination seems to denote a time period *following* the culmination. For example

(22) John left the room for a few minutes.

[2] I thank Joseph Rosenzweig for suggesting the following analysis.

But they deny that such a durative interpretation is correct, claiming that the phrase expresses *intention* rather than duration, since the following utterance would be true even if John is only out of the room for an instance:

(23) John left the room for a half hour, but returned immediately to get his umbrella.

I do not believe that the "until" clause in Example 21 has this property. Consider the related sentence

(24) John turned off the microphone until his hiccups disappeared.

One cannot deny the inference that the microphone stayed off for the full period until John's hiccups disappeared.[3] For a reading of *intention*, a modal would be needed:

(25) John turned off the microphone until his hiccups would disappear, but he had to turn it on again immediately to get the audience's attention.

I would argue then that the coercion that *seems* to be the case – that the process in question is a coercion of the *consequent state* that takes hold at the culmination point of "turn off" and continues until the agent intervenes – is *actually* the case. I believe that such a coercion is only possible if the culmination initiates an independent process, but this needs additional evidence to either support or deny.

There is one more point I want to make about how an agent derives the action she is meant to carry out, given an instruction of the form

(Do) α **until** κ.

That is that being told what condition κ to check can also convey information as to *how* one is must act in order to check it. As such, perceptual conditions can function just like purpose clauses [3, 4, 5, 6] in guiding an agent to the action she is intended to carry out as well as conveying what perceivable condition should lead her to stop it. The following two examples have led me to that conclusion:

(26) Have your helper move the tape side ways **until** the 4-foot mark on the tape coincides with the 5-foot mark on the rule.

(27) To make sure that all corners are square, meaure diagonals AD and BC, and move stake D **until** the diagonals are equal.

[3] It is a separate issue whether the microphone is turned back on afterwards, but one very relevant to instruction-following. It appears that what a agent is meant to do *after* a condition holds is strongly influenced by purpose and/or common expectations:

i. Slow down until you are out of the school zone.
(⇒ You can then speed up.)

ii. Slow down until you see my house.
(⇒ You should then stop.)

iii. Slow down until you reach the end of the cul-de-sac.
(⇒ You can then turn around.)

Without the "until" clause , the "move" verb phrases above are underspecified: they do not tell the agent (or her helper) what *direction* to move in. The "until" clauses, by indicating the condition to be achieved, conveys direction by implication – whatever direction will most directly lead to the condition becoming true.

Integrating Behaviors from α and κ

To create a realistic animated simulation, one needs to figure out how the behaviors motivated by α and κ should be integrated. There are two interesting points about this issue:

- Since all actions require resources, the agent must determine whether the behaviors motivated by α and κ can be carried out in parallel, or whether they must be interleaved.
- Even if they can be carried out in parallel, checking a condition has a cost and often undesirable side effects as well, so the agent may prefer to do it as little as possible, without preventing her pragmatic actions from coming to a successful conclusion. This means recognizing *when to start* checking for the specified condition and *how often* to do so.

The impression I get from the instructions I have looked at so far is that lexical semantics can only contribute to the solution of the first problem, in terms of what can be derived from aspectual type and aspectual coercion. For example, when a *culminated process* is coerced to a process through iteration of the basic action, the perceptual condition can be checked at the end of each iteration, as in:

(28) Strike set with fat end of hammer **until** rivet head is rounded off.

(29) Fill in low spots and strike off again **until** concrete is level with the top of the form.

On the other hand, I do think the instructions themselves help suggest answers to the questions of *when to start* checking for the specified condition and *how often* to do so. Here I am returning to the notions of *cumulative* effects and *non-deterministic* effects I introduced earlier. First consider a condition that results cumulatively from an on-going process. If the cumulative effect is perceivable, then based on the expected rate of the process, an agent can delay checking the condition until the point that the effect is likely to take hold. For example, in

(30) Chip brick with chisel **until** it can be removed.

it is not worth the agent's effort to start checking her ability to remove the brick each time she's dislodged another chip. If the cumulative effect is not perceivable, then it is as if the condition were a non-deterministic result of the process. In the case of conditions that arise non-deterministically, then the existence of a reliable probabilistic model of the process might be incorporated into an efficient perceptual strategy.[4]

[4] This suggestion is due to Joseph Rosenzweig.

The examples so far only address the cost of checking conditions and therefore the desirability of a policy that delays them as long as possible and does them as infrequently as possible. I also want to call attention to the danger of starting to check a cumulative condition too soon, a danger that can be avoided by delaying checking:

(31) Let cement dry **until** kraft paper won't stick to either surface.

Checking too soon can result in kraft paper stuck to the surface.

Conclusion

Questions about language and perception cannot be investigated outside the context of embodied agents. Animated simulation can provide us with embodied simulated human agents that allow us to better understand language, and by doing so, allow us to better employ such agents for our benefit.

References

[1] Badler, N., Webber, B., Becket, W., Douville, B., Geib, C., Moore, M., Pelachaud, C., Reich, B., Stone, M. Planning for Animation. To appear in N. Maginet-Thalmann and D. Thalmann (eds.), *Computer Animation*, Englewood Cliffs, NJ: Prentice Hall.

[2] Becket, W. The *Jack* Lisp api. Technical Report MS-CIS-94-01/Graphics Lab 59, University of Pennsylvania, Philadelphia, PA 19104-6389, 1994.

[3] Di Eugenio, B. *Understanding Natural Language Instructions: a Computational Approach to Purpose Clauses.* PhD Thesis, University of Pennsylvania, December 1993. (Technical Report MS-CIS-93-93.)

[4] Di Eugenio, B. Action Representation for Interpreting Purpose Clauses in Natural Language Instructions. *Proc. 4th Int'l Conference on Principles of Knowledge Representation and Reasoning (KR '94)*, Bonn Germany, May 1994.

[5] Di Eugenio, B. and Webber, B. Plan Recognition in Understanding Instructions. *Proc. 1st. Int'l Conference on Artificial Intelligence Planning Systems*, College Park MD, June 1992.

[6] Di Eugenio, B. and Webber, B. Pragmatic Overloading in Natural Language Instructions. Submitted to *Int'l J. of Expert Systems.*

[7] Kirsh, D. and Maglio, P. On Distinguishing Epistemic from Pragmatic Actions. *Cognitive Science* 18(4), 1994, pp. 513-549.

[8] McGowan, J. and DuBern, R. (eds.) *Home Repair.* London: Dorling Kindersley Ltd., 1991.

[9] Moens, M. and Steedman, M. Temporal Ontology and Temporal Reference. *Computational Linguistics*, 14(2):15-28, 1988.

[10] Pym, D., Pryor, L. and Murphy, D. Actions as Processes: A position on planning. Working Notes, *AAAI Spring Symposium on Extending Theories of Action.* Stanford CA, March 1995, pp. 169-173.

[11] *Reader's Digest New Complete Do-It-Yourself Manual.* Pleasantville NY: The Reader's Digest Association, 1991.