Cascaded Acts: Conscious Sequential Acting for Embodied Agents

CSE Technical Report 99-10

Haythem O. Ismail and Stuart C. Shapiro
Department of Computer Science and Engineering
and Center for Cognitive Science
State University of New York at Buffalo
226 Bell Hall
Buffalo, NY 14260-2000
E-mail: {hismail|shapiro}@cse.buffalo.edu
April 28, 2000

Abstract

A cognitive agent is expected to use awareness of the environment and its own body to direct its actions. Such awareness is required for the simplest and most common type of composite acts: sequences. To perform a sequence of acts, a cognitive agent should start performing one step when, and only when, it comes to believe that the previous one has been completed. We present a model for interleaving action, reasoning, and perception in order to allow for the conscious performance of sequences of acts. To define such a model, the notion of act completion needs to be characterized. This is achieved by developing a formal system in which acts are classified along the dimensions of telicity and primitiveness. The proposed system includes and go beyond the traditional binary telic/atelic distinction.

1 Introduction

Cognitive agents should act consciously. When carrying out a sequence of acts, an agent should be aware of the progression of its acts, and should reason, using its perception of the environment and awareness of its own body, to decide when to move to the next step in the sequence. Most acting systems in artificial intelligence do not explicitly follow this strategy. They either do not focus on the execution of (but rather on reasoning about) actions or they somehow overlook and avoid the problem completely.

We present a model for acting that makes use of reasoning, perception, and bodily feedback to direct the agent's actions. We present the model in the framework of the SNePS knowledge representation and reasoning system (Shapiro and Rapaport, 1987; Shapiro and Rapaport, 1992; Shapiro and the SNePS Implementation Group, 1999) and apply it to an agent designed using the GLAIR architecture (Hexmoor et al., 1993; Hexmoor and Shapiro, 1997). Section 2 outlines the problem in detail and discusses related work. Section 3 provides an analysis of the structure of acts using the linguistic notion of telicity. We argue that different types of telicity (we go beyond the traditional binary telic/atelic distinction) correspond to different *intentions* the agent should have while performing its actions. In section 4, we discuss primitive and composite acts and present our approach to characterizing both types. Section 5 explains in some detail the architecture of our agent, discusses the structure of various types of acts, defines the crucial technical notion of *goals*, and presents the concept of *cascaded acts* which we use to model conscious performance of sequences of acts. Finally, in Section 6, we present some examples of our theory in action.

2 Reasoning, Bodily Feedback, and Sequential Acting

2.1 The Problem

Before getting into a detailed discussion of what we are trying to do in this paper, let us first briefly point out what it is that we are *not* doing. First, we do not address planning problems. We assume that our agent has a prestored library of plans (or recipes, à la (De Eugenio, 1998)) that it uses to achieve its goals. Second, we do not address problems that may arise from various concurrent processes communicating with each other to affect the behavior of the agent. Third, in this paper, we do not consider errors and interrupts that may happen during acting. Nevertheless, our approach has been designed with these considerations firmly in mind. The problem that we are addressing here is, we believe, more fundamental.

Consider an agent that is performing a sequence of acts $\langle\alpha_1 \ldots \alpha_n\rangle$. The agent may be involved in such an activity either because it was instructed to do so, or because it was asked to achieve some goal for which it needs to perform that sequence of acts as indicated by some recipe. What does performing a *sequence* of acts mean? Intuitively, to perform α_1 and then α_2 and then α_3, etc. The simple concept of performing one act and *then* another seems to be a very fundamental one. However, when considering what it requires to actually behave in such a manner, it turns out that this concept is indeed fundamental but far from simple, or at least not as simple as it may seem. For example, consider the following instructions:

(1) Pick up the block and then walk to the table and then put the block on the table.

(2) Run to the store and then buy a bottle of milk and then run back here.

(3) Stick a stamp on the envelope and then bring the secretary here and then give her the envelope.

In the three cases, the English word *then* stands for an important constraint on how to behave according to the given instruction. In particular, in (1) for instance, the agent should first *start* picking up the block and it should start walking to the table when, and only when, it is holding the block. Similarly, it should start putting the block on the table when and only when it is near the table. That is, the above instructions could be more explicitly represented as follows:

(1') Start picking up the block; and when you are holding the block, start walking to the table; and when you are near the table, start putting the block on the table.

(2') Start running to the store; and when you are in the store, start buying a bottle of milk; and when you have bought a bottle of milk, start running back here.

(3') Start sticking a stamp on the envelope; and when there is a stamp on the envelope, start bringing the secretary here; and when the secretary is here, start giving her the envelope.

As should be obvious, the above versions of (1), (2), and (3) are linguistically awkward. They, however, represent a more detailed account of what the agent would do in order to behave correctly. There are two main differences between the two versions of the instructions. First, note the use of *start*. Indeed, the only thing that an agent is guaranteed to do is to start performing some act.[1] Whether it will actually perform the act or not depends on many factors including the occurrence of errors and various environmental conditions. Second, the agent moves on to the next step in a sequence of acts when and only when it has completed the previous one. The important point here is that the agent does not start the first act and then start the second and then the third etc.; it must start the i^{th} act only when the $(i-1)^{st}$ act has been successfully completed.

In this paper, we propose a theory for the performance of sequences of acts. Given instructions as those in (1)-(3) (either in a formal or a natural language), we need our theory to ensure that the agent would actually behave along the lines of (1')-(3'). One important feature of our theory (and, for that matter, any theory that would address the same issue) is that perception, bodily feedback, and reasoning should all be integrated into the acting system in such a way that they actually direct the execution of acts. In particular, knowing when an act is complete is a reasoning process that in many cases (see the above examples) is initiated by perception and/or bodily feedback. It is this conscious awareness of what it has done that directs the agent in its execution of a sequence of acts.

2.2 Related Work

As far as we know, nobody else has explicitly addressed this issue before. To start with, research oriented towards developing languages for representing and reasoning about actions and plans does not say anything about the correct execution of sequences of acts in real time in the real world.[2] For example, see (Traverso and Spalazzi, 1995), (Artale and Franconi, 1998), or (Chen and De Giacomo, 1999) for recent proposals with various concerns. In these systems, there is some construct in the formal language denoting sequences of acts.[3] The semantics of these constructs either specify temporal constraints on the component acts or implicitly use variants of the English *then*. No mention is made of acts being complete and agents being aware of it. We believe that such issues should be accounted for in the semantics of

[1] Where *start* could be interpreted as *intend*.

[2] In the sense of (1')-(3').

[3] Indirectly in the case of (Traverso and Spalazzi, 1995).

the action language.

Research within the GOLOG family (including standard GOLOG (Levesque et al., 1997), its concurrent version: CONGOLOG (De Giacomo et al., 1997), and its temporal version (Reiter, 1998)) may be divided into those versions of GOLOG with an off-line interpreter and those with an on-line interpreter. With an off-line interpreter, the output of the system is a sequence of actions that would lead to the goal sought should they be executed in order. However, execution is assumed to be carried out by a different component of the system and there is no mention of how sequences of acts would actually be performed in real time in the world.

On-line interpreters of GOLOG (De Giacomo et al., 1998; De Giacomo and Levesque, 1998) account for the actual execution of actions and recovery from errors that may affect their outcome.

> The robot is executing a program on-line. By this, we mean that it is physically performing the actions in sequence, as these are specified by the program. *After* each execution of a primitive action or of a program test action, the execution monitor observes whether an exogenous action has occurred." (De Giacomo et al., 1998, pp. 453-454; our emphasis)

Nevertheless, nothing is mentioned about how the monitor knows that the action has been successfully completed. The implicit assumption is that when the agent is acting, it is acting, and control returns to the main execute-monitor loop when the action is finished (see the sample Prolog implementations presented in (De Giacomo and Levesque, 1998) and (De Giacomo et al., 1998)). This would work so long as the action is merely simulated. However, if the action initiates certain activities in a hardware robot, the monitor should *wait* for these activities to terminate not merely for their initiation to be over. The mail-delivery GOLOG agent described in (Lespérance et al., 1998) seems to satisfy this requirement. The following outlines the gist of the recipe presented therein for trying to serve a customer.

1. `Start going to the customer.`
2. `Wait until you are not moving.`
3. `If you have reached the customer, then serve the customer.`
4. `Else if you are stuck, then handle failure.`

This looks very much like the sequences in (1')-(3') and indeed results in the appropriate behavior. Nevertheless, it does not provide any insights into the general issue of sequential acting; the correct way to act is explicitly represented in the recipe rather than being implemented in the semantics of the GOLOG sequencing operator. New sequences would also have to be explicitly expanded in the same way, thereby missing important generalizations that could be made regarding the structure of sequential acting. In addition, such representations are also far removed from natural linguistic instructions (note the awkwardness of (1')-(3')), a consequence that we would like to avoid in building cognitive agents.

The only place where the completion of a component act is explicitly stated in the semantics of sequences is (Davis, 1992, p. 41). Informally, he states that a sequence of two acts is active if the first is active and not yet completed or the second is active and the first is completed. This suggests (but does not guarantee) that the second act may only be activated when it is known that the first one has been completed. Davis (1992) also mentions that for some acts (what he calls *finite acts*), the reasoning system is notified that they are complete:

> The command to execute [finite acts] is shipped off to a black box: an effector control unit, or a separate module, or even a lower-level planner. It is assumed that the black box knows what it means to "begin", "continue", "interrupt", "resume", and "complete" such an act, and can report to the plan interpreter when the act has been completed."(Davis, 1992, p. 41)

We may interpret this as an indication of some sort of bodily feedback (where the body is Davis's black box). Nevertheless, this issue is not explicitly related to the execution of sequences and there is no mention of how feedback from the body allows the agent to reason about when to move on to the next step.

Research on interleaving sensing, planning, and execution (Georgeff and Lansky, 1987; Ambros-Ingerson and Steel, 1988; Shanahan, 1998, for example) addresses issues that are related to our problem here. In these systems, planning and execution are interleaved (or performed concurrently) in a producer-consumer kind of model. When the planner reaches a primitive act, the executive performs it. The sensory system updates some knowledge base with changes in the environment that are taken into consideration by the planner. It is not clear, however, how the planner knows when to send a new primitive act to the executive, essentially the problem of when to move to the next step in a sequence. For example, Shanahan discusses the *sense-plan-act cycle*:

> The robot's sensors are read, then a bounded amount of time is allotted to processing sensor data before the robot moves on to planning. Then a bounded amount of time is given over to planning before the robot proceeds to act. *Having acted*, the robot consults its sensors again, and the cycle is repeated. (Shanahan, 1998, p. 8; our emphasis)

How does the robot know that it *has acted*? It is not clear how to answer this question in light of Shanahan's discussion. It is, however, clear that the sense-plan-act cycle is not consciously controlled by the agent, i.e., it is not encoded in Shanahan's logic. Apparently, some *subconscious* mechanism controls the cycle.

Reviewing previous work, we obviously needed to *read between the lines*, in an attempt to come up with answers to the question of how the discussed systems address the problem of sequential acting that we outlined in Section 2.1. This shows that the problem has not been explicitly addressed. Even though the discussed systems certainly have their means of overcoming, or overlooking, the problem, there is yet no generalized theory to be found.

One might think that there is nothing fundamentally deep about this problem. In particular, we may suggest that concurrent processing can totally eliminate any problems with sequential acting. For example, consider the following system. There are two concurrent processes, p_1 and p_2. p_1 carries out reasoning and possibly interacting with another agent (typically, a human user). p_2 controls the execution of sequences of acts and has direct access to the status of the body. Suppose that the agent is instructed to perform a sequence of acts. Having received the instruction, p_1 sends it to p_2 which starts executing the sequence. p_2 initiates one act, sleeps until the body finishes execution, and then initiates the next act. This keeps on repeating until the whole sequence has been performed. In the meantime, p_1 is active, available for interacting with the user and can, at any time, interrupt p_2 causing it to stop the execution of the sequence.

Although this seems to solve the problem, it actually does not. The main problem is how p_2 operates. Note that p_2 initiates an act when the current activity of the body terminates. However, just termination is no guarantee that the act has actually been performed successfully. Something might have gone wrong during execution and it may not be appropriate to assume that the act has been completed. Suppose for the sake of the argument, however, that p_2 can somehow tell whether the act actually succeeded. That is, p_2 initiates an act only when the body has actually carried out the previous one. In this case, such a system might be sufficient for *some* sequences. For example, it may correctly perform the sequence in (1). However, consider (2). The act "buy a bottle of milk" does not seem to be primitive. That is, it is not one continuous bodily activity; the agent would need to reason and follow a plan in order to buy a bottle of milk. Such a plan may be another sequence of acts including walking to the dairy section, picking up a bottle of milk, walking to the cashier, paying the money, and possibly interacting with other agents. Merely monitoring the state of the body is obviously not sufficient for p_2 to tell when to initiate the act following that of buying the milk. The agent has to *know*, using both bodily feedback and reasoning, that the *goal* of buying a bottle of milk has been achieved; according to (2'), that it has bought a bottle of milk. Only then could the agent move on to the next step. Even more interesting is the sequence in (3). The act "bring the secretary here" may be performed by the agent calling the secretary, for instance. However, once it does that, its body is no longer active doing anything. This, by no means, should make the agent move on to the next step; the secretary has to actually arrive, an event that can happen any time and that is not under the control of the agent.

Waiting till bodily activities terminate is obviously not sufficient to initiate the next step in a sequence of acts. We need a theory of an agent that is conscious of what it is doing, aware of the outcome of its activities, and whether they actually achieve their intended goals. In the rest of the paper, we propose such a theory.

3 Telicity

Consider the following English instructions:

(4) Run.

(5) Run to the store.

(6) Run toward the store.

Upon hearing the above instructions, an agent would engage in certain activities in order to carry out what it was asked to do. For an outsider, someone who did not hear the instructions, the agent's responses to (4), (5), and (6) may look exactly the same: the agent is running. Nevertheless, the three instructions are intuitively different. The difference does not correspond to different motor programs the agent executes, but rather to different intentions or beliefs the agent has.

One dimension along which linguists characterize acts is that of telicity (Comrie, 1976; Dahl, 1981, for instance). An act is telic if it is described (or conceived of) as having a built-in *definite* ending state. Otherwise, the act is considered atelic. Accordingly, (5) is telic, where the definite ending state is the agent's being *at* the store.[4] On the other hand, (4) and (6) are atelic; there is no specific state whose achievement represents a (right) boundary for the activity of running.

As pointed out by many authors (see (Dowty, 1977), (Parsons, 1989), and, in a slightly different sense, (Comrie, 1976, p. 47)), part of the meaning of each of the above instructions is the running activity (in the case of (4), it may be all there is). This corresponds to the single motor program that the agent executes. The conceptual telic/atelic distinction only comes into play in deciding the agent's intentions. In particular, its intentions regarding when to stop running. In what follows, we shall analyze acts as having a component $\mathcal{P}$ representing the process that the agent has to carry out in order to perform the act. Some acts will also have another component, $\mathcal{S}$, representing the state whose achievement signals the completion of the act.

In some cases, (5) for example, the completion of the act is almost simultaneous with the agent's ending the process $\mathcal{P}$. In such cases, telicity implies a strict condition on when the agent should end the process that it is carrying out. For our agent to truthfully say that it has run to the store, it has to stop running when, and only when, it is at the store. There are cases, however, where such a relation between $\mathcal{S}$ and $\mathcal{P}$ fails to hold. Consider the following examples.

(7) Push the rock down the hill into the river.

(8) Slide the pen across the table to John.

Note that, in the case of (7), the process $\mathcal{P}$ ceases once the agent pushes the rock. Nevertheless, a period of time elapses before the state $\mathcal{S}$ (the rock being in the river) starts to hold. During this period, the agent is not, and typically cannot, do anything to help achieve $\mathcal{S}$. Note that it is not appropriate for the agent to utter (9) after having pushed the rock.[5]

(9) I am pushing the rock down the hill into the river.

In examples like (5), the achievement of $\mathcal{S}$ is totally dependent on the agent's behavior and the cessation of $\mathcal{P}$ takes place when and only when the

[4] Of course the definiteness here is plagued by the vagueness of *at*.

[5] Note that saying that it has pushed the rock down the hill into the river is not appropriate either.

act is complete. In examples like (7) and (8), on the other hand, achieving the state $\mathcal{S}$ is only partially dependent on the agent's behavior. The agent merely initiates a sequence of events that (may) result in achieving the intended goal and hence completing the act. These two cases correspond to Talmy's (in press) sentences that involve what he calls *extended causation* and *onset causation*, respectively. Telic acts may therefore be categorized into two types: *telic with extended causation* (denoted $\overset{\rightarrow\bullet}{\text{telic}}$), where the cessation of the process is almost simultaneous with the completion of the act; and *telic with onset causation* (denoted $\overset{\rightarrow\ldots\bullet}{\text{telic}}$), where a temporal gap separates the cessation of the process from the completion of the act.

To make the analysis more concrete, let $<$ denote temporal precedence and let $\mathcal{T}_{\mathcal{O}}(x)$ and $\mathcal{T}_{\mathcal{C}}(x)$ be (respectively) the times of the *onset* and *cessation* of the state/process x[6]. The two types of telic acts may be formally characterized as follows, where $\texttt{complete}(\alpha)$ is the state of the act α being complete.

$\overset{\rightarrow\ldots\bullet}{\text{telic}}$: $\mathcal{T}_{\mathcal{C}}(\mathcal{P}) < \mathcal{T}_{\mathcal{O}}(\texttt{complete}(\alpha))$.

$\overset{\rightarrow\bullet}{\text{telic}}$: $\mathcal{T}_{\mathcal{C}}(\mathcal{P}) \not< \mathcal{T}_{\mathcal{O}}(\texttt{complete}(\alpha))$ and $\mathcal{T}_{\mathcal{C}}(\mathcal{P}) \not> \mathcal{T}_{\mathcal{O}}(\texttt{complete}(\alpha))$.

The conjunction characterizing $\overset{\rightarrow\bullet}{\text{telic}}$ acts means that $\mathcal{P}$ ceases *around* the time at which $\mathcal{S}$ starts to hold. If we think of times as points, then this simply means that the two times are identical. However, if we think of times as intervals, then it only means that the two times overlap. For convenience, we shall refer to this relation (i.e., the conjunction of $\not<$ and $\not>$) by the symbol "$\doteq$".

In general, the difference between telic acts (whether $\overset{\rightarrow\ldots\bullet}{\text{telic}}$ or $\overset{\rightarrow\bullet}{\text{telic}}$) and atelic acts, hinges on the existence of certain temporal constraints on when the act is considered complete. For telic acts, there is a state ($\mathcal{S}$) that the act cannot be considered complete before, nor could it extend after, it starts. For atelic acts, no state so constrains the completion of the act. Going to a higher level of abstraction, let us define the two following predicates for a general act, α, and state, $\mathcal{S}$.

- $\mathrm{R}(\alpha, \mathcal{S}) \equiv \mathcal{T}_{\mathcal{O}}(\texttt{complete}(\alpha)) > \mathcal{T}_{\mathcal{O}}(\mathcal{S})$.
- $\mathrm{L}(\alpha, \mathcal{S}) \equiv \mathcal{T}_{\mathcal{O}}(\texttt{complete}(\alpha)) < \mathcal{T}_{\mathcal{O}}(\mathcal{S})$.

Thus, $\mathrm{R}(\alpha, \mathcal{S})$ ($\mathrm{L}(\alpha, \mathcal{S})$) holds if the act α starts to be complete after (before) the state $\mathcal{S}$ starts to hold. In general, for any act, α, the following holds:

[6]We follow (Galton, 1984) in assuming that processes form a subcategory of states.

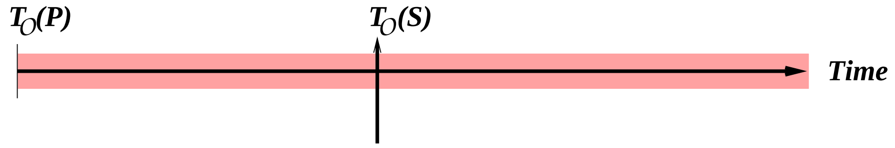

Figure 1: The structure of telic acts; $\mathcal{T}_{\mathcal{O}}(\texttt{complete}(\alpha))$ is *around* $\mathcal{T}_{\mathcal{O}}(\mathcal{S})$.

LIN. $\forall \mathcal{S}$ ($\mathcal{T}_{\mathcal{O}}(\texttt{complete}(\alpha)) \doteq \mathcal{T}_{\mathcal{O}}(\mathcal{S})$) $\bigvee$ R(α, $\mathcal{S}$) $\bigvee$ L(α, $\mathcal{S}$).[7]

The above axiom should be understood as follows. First, the disjunction is to be taken as exclusive. That is, only one of the three relations may hold between a given act and a given state. Second, **LIN** merely outlines the different temporal relations that are, by default, possible between two events. Characterizing different types of acts is achieved by asserting the existence of states that constrain **LIN**. For example, telic acts are characterized by the existence of some state, $\mathcal{S}$, for which only the first disjunct in **LIN** is possible. Figure 1 depicts the constraints on the temporal structure of a general telic act, α. Here, we are only interested in those times starting with and following $\mathcal{T}_{\mathcal{O}}(\mathcal{P})$. The shaded part of the time line represents times at which the act does *not* start to be complete, i.e., those that cannot be the value of $\mathcal{T}_{\mathcal{O}}(\texttt{complete}(\alpha))$. The vertical arrow marks the only time at which $\texttt{complete}(\alpha)$ may start to hold: the same time at which $\mathcal{S}$ starts to hold.[8]

In this paper, we are interested in acts that may be characterized by constraints of the form:

$\exists \mathcal{S}$ ($\mathcal{T}_{\mathcal{O}}(\texttt{complete}(\alpha)) \doteq \mathcal{T}_{\mathcal{O}}(\mathcal{S})$) $\bigvee$ C(α, $\mathcal{S}$).

Where C(α, $\mathcal{S}$) is R(α, $\mathcal{S}$), L(α, $\mathcal{S}$), their disjunction, or F (for falsehood). The third case corresponds to no constraints on **LIN**. The last case reduces to the constraint on telic acts.

Definition 3.1 *An act α is said to have the* **+R** *(*+**L***) feature if C(α, $\mathcal{S}$) includes R(α, $\mathcal{S}$) (L(α, $\mathcal{S}$)) as a disjunct. Otherwise, α has the* **-R** *(*-**L***) feature.*

Given the above definition, it seems that we have four possible types of acts corresponding to the four possible combinations of the **R** and **L** features.

[7] Note that is the familiar linearity axiom adopted in temporal logics. See (van Benthem, 1983).

[8] Here we are adopting a coarse-grained perspective, considering times as points. However, the formal analysis is general enough to also cover an interval-based ontology of time.

This enables us to make distinctions that go beyond the traditional binary telic/atelic classification.

Definition 3.2 *Let α be an act.*

- *α is telic if it is $\langle$**-R**, **-L**$\rangle$:*
  $\exists \mathcal{S}\ \mathcal{T}_{\mathcal{O}}(\texttt{complete}(\alpha)) \doteq \mathcal{T}_{\mathcal{O}}(\mathcal{S})$.

- *α is right-atelic (denoted $\overrightarrow{\text{atelic}}$) if it is $\langle$**+R**, **-L**$\rangle$:*
  $\exists \mathcal{S}\ (\mathcal{T}_{\mathcal{O}}(\texttt{complete}(\alpha)) \doteq \mathcal{T}_{\mathcal{O}}(\mathcal{S}))\ \bigvee\ R(\alpha, \mathcal{S})$.

- *α is left-atelic (denoted $\overleftarrow{\text{atelic}}$) if it is $\langle$**-R**, **+L**$\rangle$:*
  $\exists \mathcal{S}\ (\mathcal{T}_{\mathcal{O}}(\texttt{complete}(\alpha)) \doteq \mathcal{T}_{\mathcal{O}}(\mathcal{S}))\ \bigvee\ L(\alpha, \mathcal{S})$.

- *α is left-right-atelic (denoted $\overleftrightarrow{\text{atelic}}$) if it is $\langle$**+R**, **+L**$\rangle$:*
  $\forall \mathcal{S}\ (\mathcal{T}_{\mathcal{O}}(\texttt{complete}(\alpha)) \doteq \mathcal{T}_{\mathcal{O}}(\mathcal{S}))\ \bigvee\ R(\alpha, \mathcal{S})\ \bigvee\ L(\alpha, \mathcal{S})$.

For example, (4) is $\overleftrightarrow{\text{atelic}}$; there is no state at/before/after which the agent must stop running (see Figure 2). Classical examples of atelicity are mostly of $\overleftrightarrow{\text{atelic}}$ acts. Sentence (6) represents a $\overleftarrow{\text{atelic}}$ act. The agent may stop running at any time before reaching the store. However, once at the store, the agent must stop running since continuing to run would be away from, not toward, the store (see Figure 3). The class of $\overleftarrow{\text{atelic}}$ acts also explains certain cases that (Dahl, 1981) discusses. For example, consider the following sentences.[9]

(10) John is trying to build a house.

(11) The submarine moved toward the north pole.

According to (Dahl, 1981, p. 86), the existence of some state beyond which the process cannot continue rules out the possibility of the above sentences being atelic. Accordingly, Dahl treats them as telic. Such a move proves to be problematic as Dahl himself notices. However, given the proposed analysis, the presence of such a state beyond which the process cannot continue only means that the sentences are **-R**. In that case, they could be either telic or $\overleftarrow{\text{atelic}}$, and according to our analysis they indeed are $\overleftarrow{\text{atelic}}$ (since they are **+L**). This resolves the problems discussed by Dahl and at the same time

[9]These are Dahl's (1981) (18) and (22), respectively.

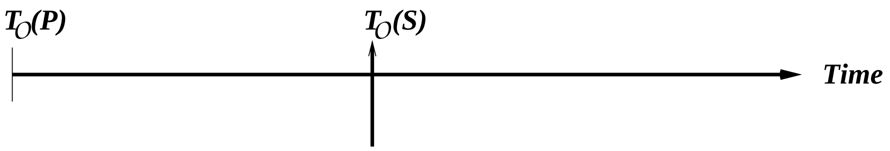

Figure 2: The structure of $\overleftrightarrow{\text{atelic}}$ acts; $\mathcal{T}_{\mathcal{O}}(\texttt{complete}(\alpha))$ is not constrained.

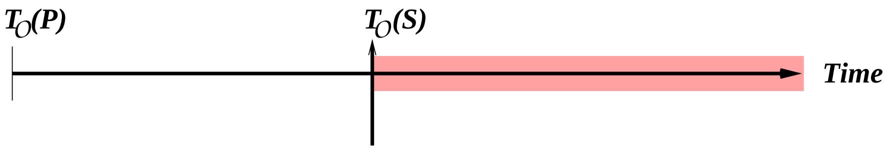

Figure 3: The structure of $\overleftarrow{\text{atelic}}$ acts; $\mathcal{T}_{\mathcal{O}}(\texttt{complete}(\alpha))$ is not after $\mathcal{T}_{\mathcal{O}}(\mathcal{S})$.

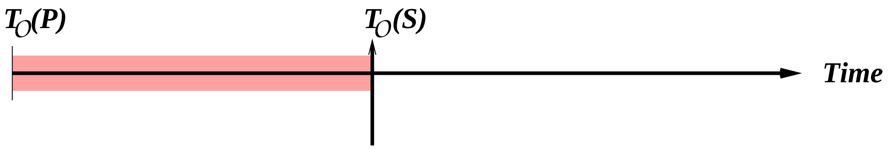

Figure 4: The structure of $\overrightarrow{\text{atelic}}$ acts; $\mathcal{T}_{\mathcal{O}}(\texttt{complete}(\alpha))$ is not before $\mathcal{T}_{\mathcal{O}}(\mathcal{S})$.

supports the intuition that sentences like (10) and (11) are different from the more traditional atelic examples (i.e., those that are $\overleftrightarrow{\text{atelic}}$ according to our analysis).

Examples of $\overrightarrow{\text{atelic}}$ acts are those that essentially lack the **L** feature. Such acts have to reach some state but then may go on indefinitely (see Figure 4). An example from the domain of the running agent is (12).

(12) Run past the store.

Other examples may include:

(13) Run no less that 2 miles.

(14) Drink no less than three cups of coffee.

(15) They lifted at least four tables.[10]

Other examples may include those analyzed by Declerck (1979) as sentences that "can be used to describe situations that are unnecessarily protracted beyond the potential terminal point" (Declerck, 1979, pp. 783–784).

(16) John painted the door.

(17) John sharpened the saw.

(18) John washed the sheet.[11]

A more elaborate discussion of the linguistic ramifications of the proposed analysis is beyond the scope of this paper; future work shall address these issues in more detail.

An agent that is expected to exhibit correct behavior should distinguish the four classes of acts mentioned above (five classes, if we consider the $\overset{\rightarrow\bullet}{\text{telic}}$ and $\overset{\rightarrow\ldots\bullet}{\text{telic}}$ distinction). However, one should remember that telicity is a linguistic notion, it's about sentences (or conceptualizations) rather than actual performances. If an agent is given the instruction (4), even though no restrictions are provided as to when to stop running, it will probably have the (implicit) intention of "stop when at the store", "stop when tired", or "stop when something interesting happens"; for instance. It definitely is not intending to keep on running forever. That is, before starting to

[10] Due to (Verkuyl, 1989, p. 83).

[11] These are (respectively) sentences (15), (91a), and (91b) in (Declerck, 1979). Also see the examples in footnote 33 therein.

run, the agent would generally foresee a (maybe not very well-defined) state that would signal an end to the running.[12] Acts falling in any of the other three classes have certain restrictions on such an ending state. An agent acting in real time should be aware of these states since they provide a way for consciously controlling the performance of composite acts, in particular sequences of acts. Before further investigating this point, we need to make precise the notions of primitive and composite acts.

4 Primitive and Composite Acts

To begin with, an act is either primitive or composite. That is, the two notions are complementary; in order to define one, it suffices to define the other. In what follows, we shall attempt to give an informal characterization of primitive acts; acts that do not conform with such a characterization shall be considered composite.

Consider two ways by which we may characterize primitive acts. We may call these the *epistemic* characterization and the *kinesthetic* characterization. For the former, an act is primitive if the agent does not need to (or even *cannot*) be "told" how to perform it. For instance, it is hard to linguistically explain to an agent how to ride a bicycle (or tie a shoe lace), maybe show it, but not tell it. Accordingly, such an act may be considered epistemically primitive. On the other hand, one may explain to the agent how to cross the street for instance: press the button, wait for the walk light, and then walk to the other side. Crossing the street would therefore be considered epistemically composite. In general, an act is epistemically primitive if the agent knows *how to* do it but cannot (easily) reason about how it does it.[13]

Kinesthetic characterization of primitive acts is based on the relation between an agent's intentions and its body. Here, an act is primitive if the agent has no control over its performance. For instance, a person may intend to move her arm and she can do that by contracting her muscles in a certain way. She has some control over the degree and speed of these contractions and she can interrupt the motion of her arm by simply deciding to do so. Nevertheless, she does not have full control over the whole process. The movement of the arm is made up of bits of *events* that are predetermined by our neuro-muscular make-up. The person does not have control over, or

[12] More technically, the act would be conceived of as *bounded* even though it might not be telic. See (Depraetere, 1995) for an interesting discussion on the difference between telicity and boundedness.

[13] This is, more or less, the distinction between procedural and declarative knowledge.

awareness of, the quality, duration, or speed of such events; neither can she interrupt them once they start. Actions that directly reduce to such uncontrollable events are what we may call kinesthetically primitive. Anything else is kinesthetically composite. In this case, the epistemically primitive acts of riding the bicycle or tying the shoe lace would be kinesthetically composite. However, if an act is kinesthetically primitive then it is also epistemically primitive.

The above characterizations are just two examples of what a primitive act may be. The philosophy of action literature contains various possible accounts for what may be a *basic* act; a notion that is similar (if not identical) to that of primitive acts (Goldman, 1970; McCann, 1998, for instance). This research, however, is primarily concerned with *human* action. What is reasonable to assume about humans need not be suitable for other agents, robots for instance. Humans are extremely complex agents; they are provided with a set of primitive acts that could be combined in various ways to yield a large set of composite acts. This is required because of the complex environment in which humans exist. Other agents might exist in less demanding environments and therefore need not be as complex. In particular, computational agents are usually designed to operate in relatively simple environments. In such environments, due to the limited number of behaviors expected from the agent, primitiveness may be very coarsely defined. For example, finding a red robot, making spaghetti, and giving coffee to a person are considered primitive acts in the systems described by Shapiro (1998), Artale and Franconi (1998), and Reiter (1998), respectively. Such acts are arguably not primitive for humans (not even epistemically primitive).

The main point is that an act being primitive or composite depends on the very nature of the agent. We assume that "[a]ny behaving entity has a repertoire of *primitive actions* it is capable of performing" (Shapiro et al., 1989, original emphasis). In designing an artificial agent, one has to make decisions regarding which acts are primitive and which are not. The notion of primitiveness that we adopt in this work has mainly an epistemic nature but also has some kinesthetic features. In particular, we make the following assumptions (**P** stands for *primitive*).

P1. The agent can perform any of its primitive acts; it cannot reason about how it performs them.

P2. When performing a primitive act, the agent is aware that it is performing the act. Nevertheless, it has no conscious awareness of its progression, nor of its different stages if it has any. Note that this is not a very unreasonable assumption; people, with enough skills, can perform certain acts while their attention is totally directed to something else. We view our agent to be skillful enough to carry out its primitive acts without any intellectual interference.

P3. In principle, the agent may interrupt its performance of a primitive act at any point.[14] Of course, "point" here is very coarse-grained; the agent cannot interrupt kinesthetically primitive acts (the switching of a relay, for instance).

For example, we may consider running to the store a primitive act. That is, running, reaching the store, and stopping are not consciously planned and serialized by the agent; it just knows *how* to "run-to-the-store". On a different account, running to the store may be a composite act involving the agent continuing to run based on its conscious awareness of its position with respect to the store. Our system allows us to adopt such an approach and to reduce all (physical) primitive acts to basic *movements* as suggested in (Israel, 1995). We, however, choose to make the assumption that the kind of perceptuo-motor coordination suggested above takes place at a sub-mental level.

5 Acting Consciously

An agent that knows how to perform its primitive acts should be able to perform any sequence of them. An arbitrary sequence of primitive acts is not primitive, though. Faced with a novel sequence, the agent will have to consciously control its execution. Unlike primitive acts, sequences of them have stages that the agent is aware of; it should proceed to one stage when, and only when, the previous one has been completed. As pointed out in Section 1, most theories of acting do not address this issue. A different theory is required, one that interleaves perception, reasoning, and acting to give the agent the ability to control and monitor the execution of a sequence of acts. Before getting into the details of our proposal, we need to briefly discuss the system to which our theory is applied.

5.1 Cassie: An Embodied Agent

We use **GLAIR** (Grounded Layered Architecture with Integrated Reasoning) (Hexmoor et al., 1993; Hexmoor and Shapiro, 1997) as the architecture of **Cassie**, our embodied agent. GLAIR consists of three levels: the

[14] That is only *in principle* since it is still being developed.

Knowledge Level (KL), the Perceptuo-Motor Level (PML), and the Sensori-Actuator Level (SAL).

1. **The Knowledge Level:** The level at which conscious reasoning takes place. The KL is implemented by the SNePS system (Shapiro and Rapaport, 1987; Shapiro and Rapaport, 1992; Shapiro and the SNePS Implementation Group, 1999), where SNeRE (the SNePS Rational Engine) (Kumar, 1994; Kumar and Shapiro, 1994a; Kumar and Shapiro, 1994b; Kumar, 1996) is used for initiating and controlling the execution of acts.

2. **The Perceptuo-Motor Level:** The level at which routines for carrying out primitive acts are located. This is also the location for other subconscious activities that allow for Cassie's consciousness of its body and surroundings.

3. **The Sensori-Actuator Level:** The level controlling the operation of sensors and actuators (being either hardware or simulated).

As mentioned by Shapiro (1998), the PML could be viewed as having three sublevels (for details of these, see (Shapiro, 1998)). In this paper, we are only interested in the top-most among these sublevels, what is called **PMLa**. The PMLa together with the KL are fixed and require no changes when switching between different (hardware or simulated) implementations of the robot. The two lower PML levels, **PMLw** and **PMLc**, on the other hand, are sensitive to various embodiments of CASSIE. Anything lower than PMLa (i.e., PMLw, PMLc, and SAL) shall be referred to as *the body* in the discussion that follows.

As stated above, acts are initiated and controlled by SNeRE. Acts are represented by SNePS terms at the KL. For every act term there is a procedure *attached* to it (i.e., associated with it). SNeRE uses a network activation mechanism (Kumar, 1994) where activating an act term corresponds to executing the procedure attached to it. The following is a discussion of the three types of acts recognized by SNeRE with examples of some of those readily provided by the system that shall be referred to in the rest of the paper.

1. **Physical Acts:** These are primitive acts that affect the state of Cassie's external environment (the *world*). SNePS terms denoting physical acts are associated with procedures at the PMLa. Examples include raising one's arm, taking a step, finding a block, etc. Note that the specification of physical acts depends on the level of granularity

at which we set our notion of "primitiveness", a choice that is largely determined by the overall architecture of the agent.

2. **Mental Acts:** These are primitive acts affecting Cassie's mental state; adding or retracting beliefs. Two mental acts are of particular relevance here:[15]

   - `believe`(*object1*): if the negation of the proposition *object1* is asserted, removes it, asserts *object1*, and initiates forward inference.
   - `disbelieve`(*object1*): unasserts the proposition *object1*.

3. **Control Acts:** These are acts that control various ways in which a set (possibly a singleton) of acts are to be performed. The following are relevant examples (for a complete specification of the control acts provided by SNeRE, see (Shapiro and the SNePS Implementation Group, 1999)).

   - `snsequence`(*object1* ... *objectn*): performs the acts *object1* ... *objectn* in order.
   - `snif`(*object1*), where *object1* is a set of guarded acts, and a guarded act is either of the form `if`(p, a), or of the form `else`(*elseact*), where p is a proposition, called the *condition* of the guarded act, and a and *elseact* are act terms. `snif` chooses at random one of the guarded acts whose condition is asserted, and performs its act. If none of the conditions is satisfied and the `else` clause is present, the *elseact* is performed.
   - `sniterate`(*object1*), where *object1* is a set of guarded acts. `sniterate` chooses at random one of the guarded acts whose condition is asserted, performs its act, and then performs the entire `sniterate` again. If none of the conditions is satisfied and the `else` clause is present, the *elseact* is performed.

`snsequence` does not exactly perform its argument acts in order; rather, it only *initiates* them in order. If these acts are physical acts, the procedures attached to them would be executed at the PMLa. Such PMLa activities result in initiating lower level activities of the body. Bodily activities typically proceed in a rate that is much slower than that of initiations at the PMLa. As a result of this, they would not be given a chance to continue to

[15]For clarity, we are using a slightly different syntax from that advertised by Shapiro and The SNePS Implementation Group (1998).

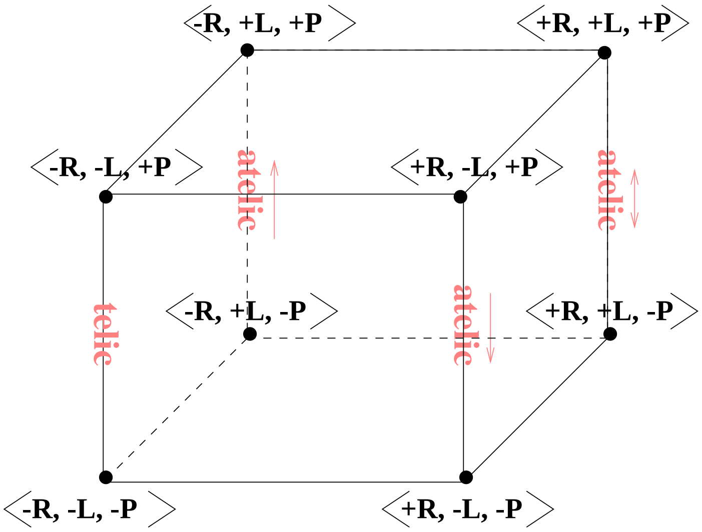

Figure 5: The RLP Cube.

completion, but would be interrupted as soon as a new activity is initiated by the PMLa routines.[16] A control act that would interleave acting, bodily feedback, and reasoning is certainly required. This shall be presented in Section 5.5.

5.2 The Anatomy of Acts

In this section we put together the ideas developed in sections 3 and 4 and take a close look at the resulting system. In doing so, we develop a deeper understanding of the structure of acts; in particular, the structure of their execution. Such an understanding is essential to the study of the properties different kinds of acts exhibit when they are embedded in a sequence. In section 3, we analyzed telicity in terms of two binary features: **R** and **L**. Adding a feature, **P**, for primitiveness, we end up with a three-dimensional feature space: the RLP cube (see Figure 5). Are there acts corresponding to the eight vertices of the cube? And if so, what is the structure of these acts? First, consider the bottom plane, that of composite acts. Consider an agent with two primitive acts: pouring coffee from a pot into a cup and

[16]For example, responding to a command to go to location A and then to location B, an agent would start going toward A and then abruptly changes directions and heads toward B.

taking one sip of coffee from the cup. In such a case, one can readily come up with four acts to fill the bottom plane.

(19) Drink coffee. ($\overleftrightarrow{\text{atelic}}$)

(20) Drink no less than three cups of coffee. ($\overrightarrow{\text{atelic}}$)

(21) Drink no more than three cups of coffee. ($\overleftarrow{\text{atelic}}$)

(22) Drink three cups of coffee. (telic)

Drinking one cup of coffee is the composite act of sipping from a cup of coffee until it is empty. A precondition for such an act is for the cup to contain coffee, a state that may be achieved by pouring coffee from the pot into the cup. The above acts are therefore composite for such an agent. To act according to (19), the agent will start the process of drinking one cup of coffee. Note that since this process is composite, it requires the agent's conscious monitoring and control. The process may then be repeated for an indefinite number of times. At any point, the agent may stop drinking coffee by either finishing one cup and not starting to drink another or by just stopping sipping coffee from a nonempty cup. Such a decision to stop drinking is certainly a conscious one and may be caused by various events including the agent's finding out that no more coffee is left in the pot. Before starting to perform the act, the agent does not have a definite scenario of how it will end, the only thing it knows is that *at some time the act will be complete.*

On the other hand, performing (22) requires the agent to, not only monitor what it is doing, but also keep track of how many cups of coffee it has drunk to stop drinking when and only when it finishes the third cup. In this case, the agent a-priori knows what completes the act. Performing (20) is a simple sequence of the performance of (22) and (19). First the agent drinks three cups of coffee to reach the lower bound the instruction indicates and then continues to drink coffee indefinitely. In this case, like in (19), the agent only knows that the act will eventually be complete. However, unlike (19), the agent knows that the act will reach completion in a state in which it has drunk three cups of coffee.

Whereas $\overrightarrow{\text{atelic}}$ acts like (20) are decomposable into two clearly distinguished components: one telic and another $\overleftrightarrow{\text{atelic}}$, $\overleftarrow{\text{atelic}}$ acts like (21) are not as simply structured. The agent starts to drink coffee while consciously monitoring the three-cups upper limit. In that respect, (21) is essentially similar to (22). However, in the case of (21), the agent has one more degree of freedom; it does not have to actually reach the three-cups limit for the act to be complete. Similar to (19), prior to finishing three cups, the agent may decide to stop drinking at any point for reasons that may have nothing to do with drinking coffee. It is as if the agent executes both (19) and (22) in parallel and the act completes whenever one of them does.

Thus, while the execution of $\langle$**+R**, **-L**, **-P**$\rangle$ acts is structured by the sequencing of telic and $\overleftrightarrow{\text{atelic}}$ acts, the execution of $\langle$**-R**, **+L**, **-P**$\rangle$ acts is made up of the interleaving, or parallel execution, of telic and $\overleftrightarrow{\text{atelic}}$ acts. This reveals the more complex nature of $\overrightarrow{\text{atelic}}$ and $\overleftarrow{\text{atelic}}$ acts over that of telic and $\overleftrightarrow{\text{atelic}}$ acts. This complexity is formally manifest in the different signs of the **R** and **L** features which reflect the heterogeneous nature of these acts.

Things get more elusive as we move up to the plane of primitive acts. As pointed out in section 4, designating an act as primitive or composite by and large depends on the nature of the agent. One might, therefore, argue that we can fill the upper plane by just thinking of an agent for which (19)-(22) designate primitive acts. However, as we shall argue, such a move is, at least, not obviously valid. Consider our running agent from section 3.[17] One might consider (4) (repeated here for convenience) to be primitive, and hence $\langle$**+R**, **+L**, **+P**$\rangle$.

(4) Run.

To perform (4), the agent starts running and then may cease to run at any point it decides to. Note that this is compatible with our assumptions **P1**-**P3**: (i) the agent may know how to run but not how it runs, (ii) the agent may be unaware of the different stages of the running process, and (iii) it may interrupt the running at any point thereby putting an end to its activity. Thus, the main difference between primitive and composite $\overleftrightarrow{\text{atelic}}$ acts is that the agent has no control over or awareness of the structure of the former but consciously controls and monitors that of the latter. In both cases, however, the agent has the same epistemic status regarding their completion: it only knows that at some state they would be complete, without any further qualification of such a state.

[17] As mentioned above, we might consider an agent for which (19)-(22) are primitive. However, since those examples are not even epistemically primitive for humans, we choose to discuss other examples that could be more appreciated by the reader. Such a choice of examples has no effect on the claims to follow.

For the same, or a slightly different, agent, (5) may be considered primitive.

(5) Run to the store.

How would (5)-as-primitive be performed? The only way to conceive of (5) as a primitive act, is to assume that the agent is designed to reactively (not deliberately) stop when it reaches the store. That is, the agent starts to run and as a reaction to reaching the store it stops running. This involves no cognitive processing. In this case, it is easy to see that (5) is compatible with **P1**-**P3**.

Restricting ourselves to the same domain, can (6) and (12) be considered primitive (and therefore examples of ⟨**-R**, **+L**, **+P**⟩ and ⟨**+R**, **-L**, **+P**⟩ acts, respectively)?

(6) Run toward the store.

(12) Run past the store.

First, consider (6). Although, as noted above, $\overleftarrow{\text{atelic}}$ acts have a complex structure, there are ways to conceive of (6) as primitive. A performance of (6) may be just a performance of (5)-as-primitive. The only difference is that interrupting the first completes the act while interrupting the second does not. It could be shown that such an account is compatible with **P1**-**P3**. Regarding **P1**, the agent need not know how it runs toward the store, in particular it need not know that it does that by simply doing what it would do if it were running *to* the store. That is, the agent knows that it can perform two acts, (5) and (6), what it is not aware of is how they are both performed and that they are both actually performed in the same manner. Regarding **P2**, the agent has no awareness (and need not have any) of the internal structure of the act. Finally, the agent may interrupt the act at any point and like $\overleftrightarrow{\text{atelic}}$ acts, and unlike telic acts, the act would be considered complete.

We are aware that there might be some *pragmatic* difficulties with the above discussion. First, it is not very clear how instructions like (6) may be useful in practice. Running toward the store is *usually* something that one realizes *has* happened not something that one *intends* to do. That is, the agent may intend to run *to* the store but end up only running toward the store. In addition, since the existence of a $\overleftarrow{\text{atelic}}$ primitive act would be accompanied by that of a telic act, it does not seem reasonable to make the former primitive. For example, one might consider only (5) to be primitive, and performing (6) would simply be an intentional performance of (5) that may be interrupted. Nevertheless, it should be noted that such concerns, although valid, do not provide any argument against the *logical* possibility of ⟨**-R**, **+L**, **+P**⟩ acts. Other examples, may also stand more strongly in the face of the above objections.

(23) Pour some coffee into the cup.

(24) Lift the block above the table.

Now, let us consider (12). One might argue that since (4) and (5) may be thought of as primitive, then so is (12). Similar to our analysis of (20), (12) may be a sequence of (5) and (4). However, for (12) to be primitive, the sequencing of (5) and (4) must be hard-wired into the agent; instead of stopping when reaching the store as in (5)-as-primitive, the agent *reactively* starts performing (4). Such an apparently plausible account is not totally sound though. According to **P3**, the agent should be able to interrupt any of its primitive acts. Now, suppose that, for some reason, the agent's performance of (12) is interrupted. A reasonable requirement of an intelligent agent is to remember what it has done. For our agent to know whether it has run past the store, it needs to know whether it had reached the store at some point during the performance of (12). This simply means that the agent is aware of the internal structure of (12). Obviously this contradicts the assumption of (12)-as-primitive since it runs counter to **P2**. Note that this also means that the agent knows (whether explicitly or implicitly, fully or partially) how it runs past the store, which is incompatible with **P1**.

In general, starting from the assumption that an intelligent agent should know the outcome of its activities, if a $\overrightarrow{\text{atelic}}$ act is interrupted, the agent needs to know during which of its two stages the interruption occurred. Such necessary awareness of the structure of an act is contradictory to the notion of primitiveness. Note that this issue does not arise with telic and $\overleftrightarrow{\text{atelic}}$ (also, $\overleftarrow{\text{atelic}}$) acts since for the first, an interruption directly means that the act has not completed; and for the second, an interruption signals the completion of the act. We shall therefore assume that no acts correspond to the ⟨**+R**, **-L**, **+P**⟩ vertex of the RLP cube.

To summarize the above discussion we highlight the main points:

1. For atelic acts[18], the agent does not know *how* they will end, only that at some point, they will. In particular, atelic acts reach completion when they are interrupted.

[18] Those with **-R** and/or **-L** features.

2. For telic acts, the agent knows exactly in what state they will be complete.

3. $\langle$**+R**, **-L**, **+P**$\rangle$ acts are not logically possible.

5.3 Acts and Their Goals

Given a sequence of acts, $\langle \alpha_1 \ldots \alpha_n \rangle$, α_i $(1 < i \leq n)$ should be performed when and only when the agent *believes* that α_{i-1} has been completed.[19] The first problem though is to provide a precise account of what it means for an act to be completed and how the agent would come to know of such an event.

What it means for an act to be completed depends on whether it is telic or atelic. Telic acts have two components, $\mathcal{P}$ and $\mathcal{S}$ (see Section 3), and the act is completed when and only when $\mathcal{S}$ starts to hold. That is, Cassie *noticing* that $\mathcal{S}$ holds is sufficient for her to come to *know* that the act is completed. This gives us an approximate idea of what we are seeking. An act is completed when some state holds. In the process of performing a $\overset{\to\bullet}{\text{telic}}$ act, the agent anticipates or foresees some state that would signal the end of its activity. In a sense, such a state is the *goal* of the act. The notion of goal alluded to here is a very localized one and is tightly associated with the particular categorization of the corresponding act (more on this below). The goal of an act, α, as far as this work is concerned, is not a pragmatic one; it is that state that signals successful completion of α thereby allowing the agent to start executing other acts contingent upon its completion.[20] For a telic act, the goal is identical with the state $\mathcal{S}$.

Things are more complicated for atelic acts. A $\overset{\longleftrightarrow}{\text{atelic}}$ act does not have an $\mathcal{S}$ component in the first place; it is a pure process. It is not clear what the goal of such an act may be. One possibility is to say that $\overset{\longleftrightarrow}{\text{atelic}}$ acts do not have goals, and that such acts may be considered completed once they start. That may not be a bad idea if the act is a mental act that is arguably

[19] By performing an act we mean initiating the procedure attached to it. Also, to be precise we should say "perform $[\![\alpha_i]\!]$" rather than "perform α_i", where $[\![\alpha_i]\!]$ is the denotation of the term α_i. However, we will allow ourselves to be a little informal and use the simpler notation.

[20] In McCann's terms, this is the *result* of the act. We choose not to adopt this terminology, however, since it only works with $\overset{\to\bullet}{\text{telic}}$ acts. Goals of $\overset{\to\ldots\bullet}{\text{telic}}$ acts are *consequences*, according to McCann. Also the term *goal* has been used by some linguists to refer to the same notion (see (Declerck, 1979, p. 762) and the quotes therein).

instantaneous (think of `believe` or `disbelieve`, for instance)[21]. However, if the act is physical, such an account does not seem plausible. Consider, for instance, the act *Run* which is $\overleftrightarrow{\text{atelic}}$. If an agent is instructed to run, one should be able to actually *see* it running. Ending the activity of running immediately after it starts is definitely unacceptable. The same holds for $\overleftarrow{\text{atelic}}$ and $\overrightarrow{\text{atelic}}$ acts.

According to Section 5.2, the only thing the agent knows about the completion of an atelic act is that it will eventually happen. Thus, the goal of an atelic act, α is simply the state `complete(`α`)`. Cassie's coming to believe that such a state holds is what causes her to move on to the following step of a sequence in which α is embedded. How this belief emerges in Cassie's mind is the subject of the next section.

The above seems to provide fairly precise descriptions of what the goals of telic and atelic acts are. What about control acts? Control acts are not the kind of acts that may be described as telic or atelic. As long as the act does not have a *cover term*; i.e., an atomic conceptual structure expressible by a simple sentence; it cannot be said to be telic or atelic. For example, consider the following `snsequence` describing a recipe for a useful exercise.

```
snsequence(run-a-mile, swim, take-a-shower)
```

It is not clear whether one can ascribe any telic properties to such an act without making a category error. Control acts have structures that are far more complex than those of the acts discussed so far. In particular, several states with complex temporal and causal (in case of `snif`) relations make up the internal structure of control acts. What then is the goal of a control act, for example an arbitrary `snsequence`? One might argue that the goal of a sequence of acts is the goal of the last act in the sequence. That might sound appealing, but there are reasons not to adopt such a position. For example, consider the following sequence:

```
snsequence(Pick-up-a-block, Mark-the-block,
           Put-down-the-block)
```

Suppose that the three acts in the sequence are primitive. What is the goal of the `Put-down-the-block` act? Given the localized sense of "goal" that we are assuming, it would be something like *being empty-handed*; this is the state that signals the completion of putting down the block. Intuitively, however, it does not seem right to say that this is the goal of the whole sequence. Being empty-handed may be achieved if the agent drops the

[21] Also see (McCann, 1998, p. 85) on the same point.

block before marking it. This, by no means, signals successful completion of the sequence. In addition, a precondition of the above act is the very state of being empty-handed (to allow for picking up a block); if the goal of performing an act already holds, there is no need to perform it. In fact, in many cases, if the goal of an act already holds, then there is no *way* to perform it (think of running to the store when one is already at the store or putting down a block when one is empty-handed, for instance).

One may object by arguing that an agent might still need to perform an act even if its goal already holds. The reasoning being that an act may be performed to achieve one of its effects (the goal being only a distinguished effect). To support such an objection, one has to come up with a situation in which the goal of some act holds and an agent, nevertheless, needs to perform it for one of its effects. To start with, it is extremely difficult to come up with such a situation. Given the notion of "goal" that we are adopting (i.e., the state that signals the successful completion of an act), it is very hard to describe a realistic situation in which it is appropriate (or even possible) to perform an act while its goal holds. However, let us try to construct such a situation.

> John is a bachelor who leads a pretty dull social life. Due to the nature of his job, he spends the whole day driving his car and visiting customers. To get out of the car, John unlocks his door, opens it, and steps out. John can unlock his door either manually or electronically by pressing a button. Because John is always alone in the car, for him the goal of pressing that button is to unlock *his* door. Nevertheless, pressing the button causes the four doors of the car to be unlocked as a side effect. One day John offers his colleague, Pat, a ride to work. When they arrive, John starts to perform the usual three-step sequence to get out of the car. John's door happens to be unlocked because, for some reason, he has manually unlocked it on their way. That is, the *goal* of unlocking the door is already achieved; John can directly open his door and step out of the car. However, he still needs to perform the act of electronically unlocking *his* door in order to unlock Pat's door, a side effect of the act.

The above example seems to be a suitable one for an act whose goal holds and still needs to be performed for one of its effects. The problem though is that this assumes that the relation between an act and its goal is accidental; the goal is just an arbitrary effect of the act that happens to get some special status due to pragmatic reasons (John's life-style in the example). Such a position confuses two concepts: the act and the procedure attached to it. The act is a particular categorization (or conceptualization) of some motor program (a PMLa procedure). The same PMLa routine may be attached to different acts. The acts themselves have their goals as inherent parts of their characterization. In the above example, one would have two acts: `Unlock-my-door` and, for example, `Unlock-passenger-door` (or even more specifically, `Unlock-Pat's-door`). The goal of the first is my (i.e., John's) door's being unlocked and that of the second is the passenger's door's being unlocked. Even though they may both be attached to the same PMLa procedure (pushing the same button), the two acts are distinct mental entities, and part of that distinction is their distinct goals. More generally, for every effect of a PMLa procedure, one may have a distinct act with that effect as its goal.

This indeed suggests that an act should not be performed if its goal already holds. The simple assumption that the goal of a sequence is that of its last act is therefore not quite satisfactory.[22] Intuitively, the goal of a sequence of acts is to achieve the goals of its individual acts in order. However, this is a more complicated way of saying "to correctly complete the act". The problem is that there is no one single state that, when achieved, would signal the completion of a control act; the act being complete is something that the agent concludes based on its conscious monitoring of its progression. Therefore, like atelic acts, the goal of a control act α is simply the state complete(α), such a state is asserted to be achieved not based on sensory input or interruptions but on the very process of executing the control act.

To sum up the above discussion, we give a semi-formal definition of goals.

Definition 5.1 *Let Γ be a partial function from acts to states such that $\Gamma(\alpha)$ is the goal of α. The function is defined as follows:*

1. *If α is a telic act, then $\Gamma(\alpha) = \gamma$, where goal(γ, α) is deducible,*

2. *If α is a mental act, then $\Gamma(\alpha)$ is undefined, and*

3. *If α is an atelic or a control act, then $\Gamma(\alpha) =$ complete(α).*

Note the following:

1. Γ is a function. That is, we are assuming that each act has a *unique* goal (if it has any).

[22]Things are even more complicated with other control acts like `snif` and `sniterate`.

2. Cassie has explicit beliefs about the goals of telic acts. Ideally, however, knowledge about the goal of an act should be structural rather than assertional (Woods, 1975) since many linguists argue that "reference to the goal is an essential part of the description of the situation" (Declerck, 1989, p. 277; also see Depraetere, 1995).

3. For simplicity, we are not being precise in defining the goals of control acts. In the state `complete`(α), α should be an event token (a term denoting a particular performance of the act) rather than an act type.

5.4 Awareness

In the above section, we attempted to make precise the notion of an act being completed. The question now is how to make Cassie aware of such an event. We vaguely hinted at the answer in our preceding discussion of goals. Control acts are completed when all their stages are. Since arbitrary control acts are composite, their execution is under the conscious control of the agent. Therefore, when the final stage of a control act α is over, Cassie simply `believe`s that the state `complete(`α`)` holds. Such a belief develops over time as Cassie monitors the successful completion of the different stages of the act. The same applies for $\overrightarrow{\text{atelic}}$ acts which, as argued in Section 5.2, are sequences of telic and $\overleftrightarrow{\text{atelic}}$ acts.

For $\overleftrightarrow{\text{atelic}}$ acts, Cassie knows that the act is completed as a result of the conscious decision to interrupt it. If Cassie is performing an atelic act, then if, for whatever reason, she interrupts it, that causes her to `believe` that the state `complete(`α`)` holds.[23]

For telic acts (primitive or composite), the goal state is not a mental state, but rather a physical one. This could be either a state of the agent's body (like being empty-handed, for instance) or a state of its environment (like the block's being on the table). In both cases, the agent cannot just believe that the goal is achieved; such knowledge would have to be based on perception and/or bodily feedback. Where does this knowledge come from? Intuitively, from the body, traveling all the way up to the mind. In other words, perception and bodily feedback are modeled by having the body (GLAIR levels below PMLa) pass information up to the PMLa. This

[23] Of course, one may interrupt an act and then resume it; the interrupt is not a sign of completion in such a case. We are currently working on a theory of interrupts, and until that is fully developed, we make the assumption that interrupting an act is equivalent to ending it.

may be the status of various sensors and actuators. Such information is *translated* at the PMLa and an assertion is added with forward inference to the KL.

Two points to note:

1. Whatever assertion will be added by the PMLa to the KL will be part of Cassie's consciousness, something that she can think or talk about. Deciding what such things are is part of the design of the agent. Humans, for example, are not aware of any of the complicated processes that underlie seeing a block on the table. We can talk about the block being on the table but not about the states of rods and cones. On the other hand, one may decide that an agent should be capable of reasoning and talking about the states of its sensors and effectors (as in (Shanahan, 1998)).

2. The assertion is done with forward inference since the goal of an act may not be what one directly perceives (especially if it is a composite act). For example, consider the act `Check whether the light is on`. The goal of such an act is simply to know whether the light is on. However, one does not directly perceive that; one either sees that the light is on or that the light is off. In either case, there could be a rule to the effect that "*If I see that the light is on or that the light is off, then I know whether the light is on*".[24] By asserting whatever is perceived with forward inference, the consequent of the rule would be derived, thereby realizing that the goal of the act has been achieved.[25]

$\overleftarrow{\text{atelic}}$ acts have a dual nature. In some respects they are like telic acts, in other respects they are more like $\overleftrightarrow{\text{atelic}}$ acts. While performing a $\overleftarrow{\text{atelic}}$ act, α, the agent believes that achieving $\mathcal{S}$ (see Section 3) completes the act. That is, $\mathcal{S} \Rightarrow \texttt{complete}(\alpha)$. However $\overleftarrow{\text{atelic}}$ acts may end in two different ways. First, they might continue until they reach the state $\mathcal{S}$ beyond which they naturally expire. In that case, $\mathcal{S}$ is asserted to be achieved with forward inference just like with telic acts. This results in the belief that $\texttt{complete}(\alpha)$ holds. Second, they may be interrupted at any point during their progression. Similar to the case of $\overleftrightarrow{\text{atelic}}$ acts, the conscious decision to interrupt the act results in Cassie's believing that $\texttt{complete}(\alpha)$ is achieved.

[24] See (Maida and Shapiro, 1982) for a suggested representation of "knowing whether".

[25] It should be noted that, ultimately, the proposed account of perception may require some revision. Forward inference should be limited to only *simple* inferences, the definition of "simple" being the basic challenge.

In summary, here is how Cassie comes to know that the goals of different types of acts have been achieved (α denotes the act):

1. Control acts (including $\overrightarrow{\text{atelic}}$ acts): Coming to believe that the final stage of α is complete, Cassie performs believe(complete(α)).
2. $\overleftrightarrow{\text{atelic}}$ acts: Interrupting α, Cassie performs believe(complete(α)).
3. Telic acts: Perceiving (asserting with forward inference) a state $\mathcal{S}'$ results in asserting $\mathcal{S}$. Note that $\mathcal{S}$ could itself be $\mathcal{S}'$.
4. $\overleftarrow{\text{atelic}}$ acts: Same as (2) or (3), whichever happens first.

5.5 Cascades

Given a sequence of acts, Cassie should start performing the first and form the belief that, when its goal is achieved, she can perform the rest of the sequence. That is, achieving the goal of the first act will have a *cascade* effect resulting in the rest of the sequence being performed in a similar fashion. Such an effect requires some way to *transform* the belief of a goal having been achieved into a performance of an act (in this case a sequence). Kumar (1994) formalizes the required notion of transformers (also see (Kumar and Shapiro, 1994a) and (Kumar and Shapiro, 1994b)). The following is a slight modification of a *proposition-act* transformer suggested by Kumar (also see Shapiro and The SNePS Implementation Group, 1999)

- M: whendo(p, a), where M and p are propositions, and a is an act. If forward inference causes both M and p to be asserted, then a is performed and M is disbelieved.

We are now ready to give a precise account of the performance of sequences. This is achieved by the following control act.

- cascade(*object1* ... *objectn*) where *objecti* ($1 \leq i \leq n$) is an act term. If Γ(*object1*) is defined, then the cascade reduces to:

  ```
  snif( if( Γ(object1), cascade(object2 ... objectn)),
        else( snsequnce(
               believe(whendo( Γ(object1), cascade(object2 ... objectn))),
               object1)))
  ```

  Otherwise, it reduces to:

`snsequence`(*object1*, `cascade`(*object2* ... *objectn*))

As shown above, what the agent will exactly do when performing a cascade depends on two factors: whether the first act has a goal and whether that goal is already achieved. If the first act does not have a goal, then the cascade reduces to the sequential execution of this act directly followed by the execution of the rest of the cascade. According to Definition 5.1, this will happen in case the first act is an atelic mental act. It should be noted, however, that this is only the way we are proposing to use `cascade`; other users may have other theories of acts and their goals, and this, by no means, would affect the way cascaded acts are performed. In other words, the procedural semantics of `cascade` is not dependent on Definition 5.1. If the goal of the first act is defined, then the cascade reduces to a `snif` act. The `snif` simply makes sure that the first act is not performed if its goal is already achieved. If the goal does not already hold, then Cassie starts performing the first act only after forming the belief that, when she is done, she will perform the rest of the cascade. Note that this is guaranteed to work, since achieving the goal of an act results in forward inference that would *activate* the believed `whendo`. Breaking up the sequence in this way, allows for simple solutions to deal with errors and interrupts. This is the topic of current research.

6 Examples

In this section we use three simple examples to demonstrate the operation of cascades. The three examples are simulations of an agent carrying out the instructions represented by (1)-(3) in Section 2.1. More impressive examples require either errors and interrupts, which we are still investigating, or an actual robot acting in the world, something that we cannot present on paper. The examples are only intended to give a feel of how cascading works. The demonstrations are the output of actual SNePS runs. These outputs are slightly edited for formatting and are broken down into sections to allow for explanation. The ":" is the SNePS prompt and inputs are either assertions, commands, or simulated sensory inputs. Cassie's acts are simulated by generating English sentences describing what she is doing. These are surrounded by asterisks in the output.

First, we provide Cassie with some general rules about the goals of acts.

```
: all(x) (goal(holding(x), pickup(x))).
```

```
: all(x) (goal(at(x), {walkto(x), goto(x), runto(x)})).
: all(x,y) (goal(on(x, y), puton(x, y))).
: all(x,y) (goal(has(x, y), give(y, x))).
```

The above respectively assert that the goal of picking something up is to be holding it; the goal of walking, going, or running to some place is to be at that place; the goal of putting some object, x, on some object, y, is for x to be on y; and the goal of giving some object, y, to some agent, x, is for x to have y. As mentioned before, one would ultimately like to have a natural language interface that would extract the goals of acts by a detailed analysis of telic sentences.

The first example shows Cassie performing the sequence of acts presented in (1), repeated here for convenience.

(1) Pick up a block and then walk to the table and then put the block on the table.

In the initial situation Cassie is holding a block.

```
: holding(block).
: perform cascade(pickup(block),
                  walkto(table),
                  puton(block, table))
**Walking to TABLE**
: perform believe(at(table))  ;;Sensory input. At the table.
**Putting BLOCK on TABLE**
: perform believe(on(block, table))  ;;Sensory input.
                                     ;;The block is on the table.
```

A couple of points to note. First, since the goal of picking up a block already holds, the act was skipped and the second step in the cascade was performed right away. Second, note that Cassie does not start putting the block on the table until she comes to know (via simulated perception) that she is at the table (note the prompt).

The second example shows Cassie acting according to (2).

(2) Run to the store and then buy a bottle of milk and then come back here.

This example illustrates two main points: (i) cascading control acts and (ii) reasoning and acting while performing a cascade. We first give Cassie recipes for performing some composite acts.

```
: all(x)(ActPlan(greet(x), {say("Hi", x), say("Hello", x)})).
: ActPlan(buy(bottle-of-milk),
        cascade(goto(dairy-section),
                pickup(bottle-of-milk),
                goto(cashier),
                give(money, cashier)))
```

The first of these sentences says that to greet X (presumably a person) either say "Hi X" or "Hello X". The second says that to buy a bottle of milk (assuming that you're already in the store), go to the dairy section, pick up a bottle of milk, go to the cashier, and give money to the cashier.

Next, we ask Cassie to perform (2). To simplify matters, the last step of the cascade is for Cassie to run to the `house`. This matches (2) if we assume that the instruction was given in the `house`. At the same time, it avoids complications introduced by deictic expressions.[26]

```
: perform cascade(runto(store),
                  buy(bottle-of-milk),
                  runto(house))
**Running to STORE**
```

Now, Cassie has started running to the store but has not gotten there yet. In the meantime, we can talk to Cassie and she can perform simultaneous acts as long as they do not interfere with her running to the store.

```
: all(x)(wheneverdo(near(x), greet(x))).
: perform believe(near(Stu))  ;;Sensory input. Stu is near.
Hello STU
```

The first of the above two sentences tells Cassie that whenever she's near someone, she should greet them. The `wheneverdo` construct is similar to `whendo` (see Section 5.5) but the rule does not get discarded when the act is activated; it is used for general acting rules, rather than occasional ones. By sensing that Stu is near, forward inference activates the rule, and Cassie greets Stu. The important point here is that Cassie can reason and act while in the midst of performing a cascade.

Having reached the store, Cassie carries out the plan for buying a bottle of milk, all the while observing the greeting rule.

[26] This does not mean that Cassie cannot understand deictic expressions. See various papers in (Duchan et al., 1995).

```
: perform believe(at(store))  ;;Sensory input. At the store.
**Going to DAIRY-SECTION**
: perform believe(at(dairy-section))  ;;Sensory input.
                                      ;;Reached the dairy section.
**Picking up BOTTLE-OF-MILK**
: perform believe(holding(bottle-of-milk))  ;;Sensory input.
                                            ;;Holding the milk.
**Going to CASHIER**
: perform believe(near(Bill))   ;;Sensory input. Bill is near.
Hi BILL
: perform believe(at(cashier))  ;;Sensory input. Reached the cashier.
**Giving MONEY to CASHIER**
: perform believe(has(cashier, money))  ;;Sensory input.
                                        ;;The cashier has the money.
**Running to HOUSE**
: perform believe(near(Sally))  ;;Sensory input. Sally is near.
Hello SALLY
: perform believe(at(house))  ;;Sensory input. At the house.
```

The second step of the top-level cascade (buying the milk) expanded into another lower-level cascade. It is only after the latter has been completed that Cassie resumed performing the former. Observe that Cassie started running back to the house only after (successfully) giving the money to the cashier. What initiated the act of running to the house? According to Definition 5.1, it is achieving the goal `complete(buy(bottle-of-milk))`. Note that Cassie does not come to know that this goal has been achieved through sensory input; Cassie knows that by successfully finishing the lower-level cascade. Asserting the completion of the act of buying the milk happens internally, essentially when the last step of the lower-level cascade terminates successfully.

The final example demonstrates Cassie's performance of the sequence of acts represented by (3).

(3) Stick a stamp on the envelope and then bring the secretary here and then give her the envelope.

In what follows, we assume *Gloria* to be the secretary.

```
: all(x) (ActPlan(bring(x), say("Come here", x))).
: goal(on(envelope, stamp), stickon(envelope, stamp)).
: goal(here(Gloria), bring(Gloria)).
```

The above defines the goals of sticking a stamp on the envelope and of bringing Gloria, and asserts that to bring someone just call them.

```
perform cascade(stickon(envelope, stamp),
                bring(Gloria),
                give(envelope, Gloria))
**Sticking ENVELOPE on STAMP**
: perform believe(on(envelope, stamp)) ;;Sensory input.
                                       ;;The stamp is on the envelope.
Come here GLORIA
```

At this point, Cassie is physically not doing anything. Having called Gloria, she can only wait for her to arrive in order to hand her the envelope. In the meantime, we can talk to Cassie and she can engage in other activities.

```
: good-secretary(Gloria).
: perform believe(near(Bill)) ;;Sensory input. Bill is near.
Hi BILL
: late(Gloria).
: perform believe(here(Gloria)) ;;Sensory input. Gloria arrives.
**Giving ENVELOPE to GLORIA**
: perform believe(has(Gloria, envelope)) ;;Sensory input.
                                         ;;Gloria has the envelope.
```

Merely terminating bodily activities would not have been an appropriate signal to start giving the envelope to Gloria. Cassie had to wait till she senses that Gloria has arrived. The performance of a cascade could be indefinitely extended in time since achieving some goals might be contingent upon exogenous events (in this case, Gloria's arrival).[27]

7 Conclusions

Performing sequences of acts is not as simple as it may seem. An embodied agent that properly executes a sequence of acts makes use of its awareness of the environment and its own body to reason about when to move to the next step in the sequence. Such an apparently simple concept does not seem to have been explicitly addressed in general and abstract enough terms.

[27] Note that if the goal of bringing Gloria is for her to be *near*, rather than *here*, Cassie should have greeted her. However, in such a case, Gloria being near would activate two acts: the pending cascade, and the greeting. We are still investigating such cases in our current research. The basic idea is to define a system of dynamic priorities among acts.

When an agent is acting, it has a certain conceptualization of what it is doing. Such conceptualizations vary along the dimension of telicity. Different types of telicity correspond to different criteria regarding what signals successful completion of an act. Successful completion of an act is signaled by some state —the goal— starting to hold. Telic acts have a built-in goal that is essentially part of their structure. Atelic acts, on the other hand, do not have such a built-in goal, and complete by being consciously terminated by the agent at some arbitrary point. Control acts, which are neither telic nor atelic in the classical senses of the terms, are consciously performed. The agent is aware of their progression and the goal of such acts is simply the state of their being complete, something that the agent comes to believe by virtue of its very performance of the act.

We introduced a mechanism for performing sequences of acts, *cascades*, that gives the agent more conscious control of their execution. To cascade a sequence of acts, the agent starts performing the first and forms the belief that when its goal is achieved it shall (recursively) perform the rest of the cascade.

8 Future Work

The cascading mechanism presented in this paper might seem a little bit reactive. However, we are currently developing an agent that makes use of cascades and can, at any point, reason about what to do next based on a dynamic system of priorities among acts. This agent is capable of handling interrupts and errors. It is primarily the use of cascades that gives the agent a chance to think about what to do and allows it to interleave various processes. The agent also uses a model of temporal progression based on (Shapiro, 1998) that we are planning to extend to account for various problems of temporal granularity. Other directions in which this work may proceed include a thorough linguistic investigation of the different types of telicity presented and whether they may help us understand more about linguistic aspect.

9 Acknowledgments

The authors thank the members of the SNePS Research Group of the University at Buffalo for their support and comments on the work reported in this paper. Comments by William J. Rapaport and Frances L. Johnson on an earlier draft are highly appreciated.

This work was supported in part by ONR under contract N00014-98-C-0062.

References

Ambros-Ingerson, J. and Steel, S. (1988). Integrating planning, execution and monitoring. In *Proceedings of the 7th National Conference on Artificial Intelligence*, pages 83–88, San Mateo, CA. Morgan Kaufmann.

Artale, A. and Franconi, E. (1998). A temporal description logic for reasoning about actions and plans. *Journal of Artificial Intelligence Research*, 9:463–506.

Chen, X. and De Giacomo, G. (1999). Reasoning about nondeterministic and concurrent actions: A process algebra approach. *Artificial Intelligence*, 107:63–98.

Comrie, B. (1976). *Aspect*. Cambridge University Press, Cambridge.

Dahl, Ö. (1981). On the definition of the telic-atelic (bounded-nonbounded) distinction. In Tedeschi, P. and Zaenen, A., editors, *Tense and Aspect*, volume 14 of *Syntax and Semantics*, pages 79–90. Academic Press, New York.

Davis, E. (1992). Semantics for tasks that can be interrupted or abandoned. In Hendler, J., editor, *The 1st International Conference on Artificial Intelligence Planning Systems (AIPS '92)*, pages 37–44, San Francisco, CA. Morgan Kaufmann.

De Eugenio, B. (1998). An action representation formalism to interpret natural language instructions. *Computational Intelligence*, 14(1):89–113.

De Giacomo, G., Lespérance, Y., and Levesque, H. (1997). Reasoning about concurrent execution, prioritized interrupts, and exogenous actions in the situation calculus. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, pages 1221–1226, San Francisco, CA. Morgan Kaufmann.

De Giacomo, G. and Levesque, H. (1998). An incremental interpreter for high-level programs with sensing. Technical report, Department of Computer Science, University of Toronto.

De Giacomo, G., Reiter, R., and Soutchanski, M. (1998). Execution monitoring of high-level robot programs. In Cohn, A. G., Schubert, L. K., and Shapiro, S. C., editors, *Proceedings of the 6th International Conference on Principles of Knowledge Representation and Reasoning (KR '98)*, pages 453–464, San Francisco, CA. Morgan Kaufmann.

Declerck, R. (1979). Aspect and the bounded/unbounded (telic/atelic) distinction. *Linguistics*, 17:761–794.

Declerck, R. (1989). Boundedness and the structure of situations. *Leuvense Bijdragen*, 78:275–304.

Depraetere, I. (1995). On the necessity of distinguishing between (un)boundedness and (a)telicity. *Linguistics and Philosophy*, 18:1–19.

Dowty, D. (1977). Toward a semantic analysis of verb aspect and the English 'imperfective' progressive. *Linguistics and Philosophy*, 1:45–77.

Duchan, J., Bruder, G., and Hewitt, L., editors (1995). *Deixis in Narrative: A Cognitive Science Perspective*. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ.

Galton, A. (1984). *The Logic of Aspect*. Clarendon Press, Oxford.

Georgeff, M. and Lansky, A. (1987). Reactive reasoning and planning. In *Proceedings of the 6th National Conference on Artificial Intelligence*, pages 677–682, Los Altos, CA. Morgan Kaufmann.

Goldman, A. (1970). *A Theory of Human Action*. Princeton University Press, Princeton, NJ.

Hexmoor, H., Lammens, J., and Shapiro, S. C. (1993). Embodiment in GLAIR: a grounded layered architecture with integrated reasoning for autonomous agents. In Dankel, II, D. D. and Stewman, J., editors, *Proceedings of The Sixth Florida AI Research Symposium (FLAIRS 93)*, pages 325–329. The Florida AI Research Society.

Hexmoor, H. and Shapiro, S. C. (1997). Integrating skill and knowledge in expert agents. In Feltovich, P. J., Ford, K. M., and Hoffman, R. R., editors, *Expertise in Context*, pages 383–404. AAAI Press/MIT Press, Menlo Park, CA / Cambridge, MA.

Israel, D. (1995). Process logics of action. Draft. Avialable at <http://www.ai.sri.com/~israel/>.

Kumar, D. (1994). *From Beliefs and Goals to Intentions and Actions: An Amalgamated Model of Inference and Acting.* PhD thesis, Department of Computer Science, State University of New York at Buffalo, Buffalo, NY. Technical Report 94-04.

Kumar, D. (1996). The SNePS BDI architecture. *Decision Support Systems*, 16:3–19.

Kumar, D. and Shapiro, S. C. (1994a). Acting in service of inference (and vice versa). In Dankel, II, D. D., editor, *Proceedings of the Seventh Florida Artificial Intelligence Research Symposium*, pages 207–211, St. Petersburg, FL. The Florida AI Research Society.

Kumar, D. and Shapiro, S. C. (1994b). The OK BDI architecture. *International Journal on Artificial Intelligence Tools*, 3(3):349–366.

Lespérance, Y., Tam, K., and Jenkin, M. (1998). Reactivity in a logic-based robot programming framework. In *Cognitive Robotics: Papers from the 1998 AAAI Fall Symposium*, pages 98–105, Menlo Park, CA. AAAI Press. Technical report FS-98-02.

Levesque, H., Reiter, R., Lespérance, Y., Lin, F., and Scherl, R. (1997). GOLOG: A logic programming language for dynamic domains. *The Journal of Logic Programming*, 31(1–3):59–83.

Maida, A. S. and Shapiro, S. C. (1982). Intensional concepts in propositional semantic networks. *Cognitive Science*, 6:291–330.

McCann, H. (1998). *The Works of Agency: On Human Action, Will, and Freedom.* Cornell University Press, Ithaca, NY.

Parsons, T. (1989). The progressive in English: Events, states, and processes. *Linguistics and Philosophy*, 12:213–241.

Reiter, R. (1998). Sequential, temporal GOLOG. In Cohn, A. G., Schubert, L. K., and Shapiro, S. C., editors, *Proceedings of the 6th International Conference on Principles of Knowledge Representation and Reasoning (KR '98)*, pages 547–556, San Francisco, CA. Morgan Kaufmann.

Shanahan, M. (1998). Reinventing shakey. In *Cognitive Robotics: Papers from the 1998 AAAI Fall Symposium*, pages 125–135, Menlo Park, CA. AAAI Press. Technical report FS-98-02.

Shapiro, S. C. (1998). Embodied Cassie. In *Cognitive Robotics: Papers from the 1998 AAAI Fall Symposium*, pages 136–143, Menlo Park, CA. AAAI Press. Technical report FS-98-02.

Shapiro, S. C., Kumar, D., and Ali, S. (1989). A propositional network approach to plans and plan recognition. In *Proceedings of the 1988 Workshop on Plan Recognition*, pages 21–41, Los Altos, CA. Morgan Kaufmann.

Shapiro, S. C. and Rapaport, W. J. (1987). SNePS considered as a fully intensional propositional semantic network. In Cercone, N. and McCalla, G., editors, *The Knowledge Frontier*, pages 263–315. Springer-Verlag, New York.

Shapiro, S. C. and Rapaport, W. J. (1992). The SNePS family. *Computers and mathematics with applications*, 23(2–5):243–275. Reprinted in F. Lehman, ed. *Semantic Networks in Artificial Intelligence*, pages 243–275. Pergamon Press, Oxford, 1992.

Shapiro, S. C. and the SNePS Implementation Group (1999). *SNePS 2.5 User's Manual*. Department of Computer Science and Engineering, State University of New York at Buffalo.

Talmy, L. (In Press). *Toward a Cognitive Semantics*. MIT Press.

Traverso, P. and Spalazzi, L. (1995). A logic for acting, sensing, and planning. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, volume 2, pages 1941–1947, San Mateo, CA. Morgan Kaufmann.

van Benthem, J. (1983). *The Logic of Time*. D. Reidel Publisihng Company, Dordrecht, Holland.

Verkuyl, H. (1989). Aspectual classes and aspectual composition. *Linguistics and Philosophy*, 12:39–94.

Woods, W. (1975). What's in a link: Foundations of semantic networks. In Bobrow, D. and Collins, A., editors, *Representation and Understanding: Studies in Cognitive Science*, pages 35–81. Academic Press, New York.