Embodied Cassie

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, U.S.A.
shapiro@cs.buffalo.edu

Abstract

We have enhanced a computational cognitive agent by embodying it with real and simulated bodies operating in real and simulated worlds. This has allowed us to experiment with various ways that embodiment influences the creation and meaning of the agent's beliefs and other terms in its knowledge base, including: symbol-grounding by perception and action; first-person privileged knowledge; the representation and use of indexicals; having a personal sense of time; and low-level bodily awareness.

Introduction

We have been engaged in a series of projects in which Cassie, the SNePS cognitive agent (Shapiro & Rapaport 1987; Shapiro 1989; Shapiro & Rapaport 1991; 1992; Shapiro & The SNePS Implementation Group 1998), has been incorporated into a hardware or software-simulated cognitive robot. The capabilities of the embodied Cassie have included: input and output in fragments of English; reasoning; performance of primitive and composite acts; and vision. In this paper, I give an overview of these projects, and discuss some of the ways embodiment influences the creation and meaning of the agent's beliefs and other terms in its knowledge base. The issues discussed are: symbol-grounding by perception and action; first-person privileged knowledge; the representation and use of indexicals; having a personal sense of time; and low-level bodily awareness.

Interaction with Cassie

Interaction with Cassie is carried out in a fragment of English implemented in an ATN analysis/generation grammar (*see* (Shapiro 1982; 1989)). Each input can be a statement, a question, or a command. A summary of the I/O loop is:

1. The input is analyzed using syntax, semantics, and pragmatics, along with all of Cassie's current beliefs, supplemented by inference if needed, by interaction with the (real or simulated) world if needed, and by clarification dialogue with the user if needed.
2. Analysis of the input may result in new terms being introduced into Cassie's belief space, some of which might represent new beliefs (*see* (Shapiro 1993)).

3(a) If the input was a statement, an English sentence is generated from the SNePS term representing the main proposition expressed by the input, and this generated statement is output preceded by the canned phrase "`I understand that`".

(b) If the input was a question, the answer to the question is retrieved or inferred based on Cassie's current beliefs, and that answer is output in English.

(c) If the input was a command, Cassie carries out the command, and outputs generated English sentences expressing what she is doing as she is doing it.

Of course, something might go wrong—Cassie might not understand the input, might not be able to answer the question, or might not be able to carry out the command.

The FEVAHR

From 1994 to 1997, we were involved in a project sponsored by NASA to embody Cassie as a "Foveal Extra-Vehicular Activity Helper-Retriever (FEVAHR)." For the hardware version of this robot, we used a commercial Nomad robot enhanced, by Amherst Systems, Inc., with a foveal vision system consisting of a pair of cameras with associated hardware and software. The Nomad came supplied with sonar, bumpers, and wheels.

Cassie, in her role as a FEVAHR, operates in a $17' \times 17'$ room containing:

- Cassie;
- Stu, a human supervisor;
- Bill, another human;
- a green robot;
- three indistinguishable red robots.

(In the actual room in which the Nomad robot operated, "Stu" was a yellow cube, "Bill" was a blue cube, the green robot was a green ball, and the red robots were red balls.) Cassie is always talking to either Stu or Bill (initially to Stu). That person addresses Cassie when he talks, and Cassie always addresses that person when she talks. Cassie can be told to talk to the other person, or to find, look at, go to, or follow any of the people or robots in the room. Cassie can also engage in conversations on a limited number of other topics in a fragment of English, similar to some of the conversations in (Shapiro 1989).

The Architecture of Embodied Cassie

The architecture we have been using for the embodied Cassie is GLAIR (Grounded Layered Architecture with Integrated Reasoning) (Hexmoor, Lammens, & Shapiro 1993; Lammens, Hexmoor, & Shapiro 1995). This is a three-level architecture consisting of:

The Knowledge Level (KL): the location of symbolic "conscious" reasoning, implemented by the SNePS Knowledge Representation and Reasoning (KR&R) system, in which terms of the SNePS logical language represent the mental entities conceived of and reasoned about by Cassie;

The Perceptuo-Motor Level (PML): the location of routine behaviors that can be carried out without thinking about each step, and of the data objects that these behaviors operate on;

The Sensori-Actuator Level (SAL): the location of control of individual sensors and effectors.

SNePS (and hence the KL) is implemented in Common Lisp. The SAL has been implemented in C. The PML has been implemented in three sub-levels:

1. The highest sub-level (which I will refer to as PMLa) has been implemented in Common Lisp, and contains the definitions of the functions that implement the activity represented by SNePS action-terms.
2. The middle sub-level (henceforth PMLw) contains a set of Common Lisp symbols and functions defined in the `WORLD` package which use Common Lisp's foreign function facility to link to
3. the lowest sub-level (henceforth PMLc), which has been a C implementation of "behavioral networks" (Hexmoor 1995).

The Common Lisp programs, PMLc, and the SAL run on different processes, and, in some circumstances, on different machines.

During development of the KL part of the FEVAHR, and subsequently, we used several simulations of the robot and of the world it operates in:

The ASCII Simulation replaces everything below PMLa, with functions which just print an indication of what the FEVAHR would do;

The Garnet Simulation simulates the FEVAHR and its world by Garnet (Gar 1993) objects in a Garnet window.

The VRML Simulation simulates the FEVAHR and its world by VRML (Virtual Reality Modeling Language, *see* `http://www.vrml.org/`) objects visible through a World-Wide Web browser.

The Nomad Simulator uses the simulator that was included with the Nomad robot, enhanced by a simulation of the FEVAHR's world and its vision system.

It is significant that no code at the KL or PMLa levels need be changed when switching among these four different simulations and the hardware robot. All that is required is a different PMLw file that just prints a message, or makes calls to its appropriate PMLc sub-level.

Terms, Entities, Symbols, and Objects

Cassie uses SNePS terms to think about the objects in her world. These objects, as Cassie conceives of them, are Cassie's mental entities, and may correspond more or less well with objects in the real world. When the hardware Nomad is being used, there really are objects in the real world. However, when one of the simulators is being used, we must use simulated objects that are, nevertheless, distinct from the SNePS terms that represent them. The simulated objects we use are Common Lisp symbols[1] in the `WORLD` package. It will be remembered that these symbols reside in the PMLw level of the GLAIR architecture.

For example, Cassie uses the individual constant `B5` to represent Bill, and at the beginning of the interaction she has the following two beliefs about him (expressions of SNePS logic will be shown using the syntax of SNePSLOG (Shapiro *et al.* 1981; Shapiro & The SNePS Implementation Group 1998, Chapter 7), one of the available SNePS interface languages):

```
Person(B5)
Propername(B5, "Bill")
```

That is, the term `B5` denotes the mental entity, *a person named "Bill"*. Meanwhile, the simulated Bill is `WORLD:BILL`; the vision researchers actually used a big blue cube to stand in for Bill; and the real Bill is down the hall in his office.

Embodiment and Beliefs

Symbol-Grounding by Perception

Symbol-grounding, as discussed by Harnad (Harnad 1990) and others, deals with the grounding of KR symbols in nonsymbolic representations of perceptions of real-world objects. One way to focus on the issue is to consider two ways for a computational cognitive agent to convince us that she understands the color green. One way is purely in language—name green things, contrast them with things of other colors, discuss different shades of green, etc. Another way is to pick out green things in the real world. Building and experimenting with cognitive robots gives us the ability to experiment with and demonstrate this latter way of showing "understanding."

[1] In the Garnet simulation, we use structured objects, but I will continue to use the term "symbol" to avoid confusion with real-world objects.

The Cassie FEVAHR grounds some of its symbols in perception by "aligning" some of its SNePS KL terms with sub-KL *descriptions.* A description is a pair, $\langle color, shape \rangle$, where each of *color* and *shape* is a number or symbol that can be used by the PMLw or lower levels to find the real or simulated objects. In the Garnet simulation, these are symbols imported from the Garnet packages that specify the color and shape of the Garnet objects. In the Nomad robot, these are numbers which, when passed as arguments to SAL C functions, designate the appropriate colors and shapes to the vision system. Table 1 shows the descriptions of Stu,

KL term	ASCII World Description
`Stu`	⟨`WORLD:YELLOW, WORLD:SQUARE`⟩
`Bill`	⟨`WORLD:BLUE, WORLD:SQUARE`⟩
`green`	⟨`WORLD:GREEN, NIL`⟩
`red`	⟨`WORLD:RED, NIL`⟩
`robots`	⟨`NIL, WORLD:CIRCLE`⟩

Table 1: Descriptions aligned with KL terms

Bill, the color green, the color red, and the category of robots in the ASCII simulation. Partial descriptions are unified to get full descriptions. For example, the full description of the green robot in the ASCII simulation is ⟨`WORLD:GREEN, WORLD:CIRCLE`⟩

Consider how the Nomad robot FEVAHR responds to the command, "Find the green robot."

1. The parser finds the SNePS term (`B6`) that represents the green robot.
2. The PMLa function for finding is given `B6`, finds its description to be $\langle 11, 22 \rangle$, and calls the PMLw function for finding something of that description.
3. The PMLw function calls the appropriate C procedures, with arguments 11 and 22, that direct the vision system to move the cameras until they focus on something of color 11 and shape 22. The implementors of the vision system have already trained the vision system so that color 11 is what we would call "green" and shape 22 is what we would call "spherical."
4. A SNePS term is created that represents the belief that Cassie has found `B6`.
5. Cassie expresses this belief by generating the sentence "I found the green robot."
6. The PMLw symbol `WORLD:GREENIE` is placed as the value of the PMLa variable `*STM*`, which serves as Cassie's short-term iconic memory.
7. A SNePS term is created that represents the belief that Cassie is looking at `B6`.
8. Cassie expresses this belief by generating the sentence "I am looking at the green robot."

In the process, Cassie demonstrates her understanding of "green" and of "robot" by the Nomad robot's actually turning its cameras to focus on the green robot (ball).

The following is an example of an interaction with the ASCII version of the FEVAHR. Sentences preceded by ":" are input; sentences beginning with "`The FEVAHR`" are the ASCII simulations of FEVAHR actions; the other sentences are Cassie's output.

```
 : Find the green robot.
The FEVAHR is looking for something
    that's GREEN and a CIRCLE.
The FEVAHR found WORLD:GREENIE.
 I found the green robot.
The FEVAHR is looking at WORLD:GREENIE.
 I am looking at the green robot.
```

The fact that the descriptions are sub-KL symbols captures the phenomenon of "I know what she looks like, but I can't describe her." They do not represent mental entities like KL terms do. Bill is not a blue square, and Cassie doesn't think he is. Descriptions are just arbitrary symbols used by the (real or simulated) vision system to locate objects in the (real or simulated) world. They are links that ground KR terms in objects in the world.

The PMLa variable `*STM*` serves as Cassie's iconic short-term memory, and replaces, in the case of the simulations, or supplements, in the case of the hardware robot, the cameras and SAL vision system. `*STM*` always contains the PMLw symbol standing for the object Cassie is currently looking at. From this symbol, the description of the object is directly accessible. For example, in the Garnet simulation, the green robot is simulated by a structured object whose `:FILLING-STYLE` slot is filled by `OPAL:GREEN-FILL`. If the symbol in `*STM*` satisfies the description of an object Cassie is requested to find or look at, she doesn't have to do anything:

```
 : Look at a robot.
 I am looking at the green robot.
```

Symbol-Grounding by Action

In addition to demonstrating her understanding of "green" and of the shape of robots, Cassie has demonstrated her understanding of the verb "find". Similarly, if we ask her to "Go to the green robot," the Nomad robot will actually move until it is next to the green robot with its cameras focussing on it, demonstrating an understanding of "go to." Using the ASCII simulation:

```
 : Go to the green robot.
The FEVAHR is going to WORLD:GREENIE
 I went to the green robot.
 I am near the green robot.
```

If we ask the Nomad robot to follow Bill, it will move until it is next to Bill, and then remain next to him even if he moves.

```
 : Follow Bill.
The FEVAHR is looking for something
```

```
   that's BLUE and a SQUARE.
The FEVAHR found WORLD:BILL.
 I found Bill.
The FEVAHR is looking at WORLD:BILL.
 I am looking at Bill.
The FEVAHR is going to WORLD:BILL
 I went to Bill.
 I am near Bill.
The FEVAHR is following WORLD:BILL
 I am following Bill.
```

FEVAHR primitive action terms include `talkto`, `find`, `goto`, `follow`, and `stop`. In this paper, action terms will be written as though they were function symbols taking as arguments terms denoting the objects the action is to be performed on. The functional term, itself, represents an *act*. For example, the functional term `find(B6)` represents the act of finding the green robot, represented by `B6`.

We consider an *event* to be something that happens over some time interval. We use two event-forming functions:

`Near(p, o)`: Agent `p` is near object `o`.

`Act(p, a(o))`: Agent `p` performs act `a` on object `o`.

For example, `Near(B1, B6)` represents the event of Cassie, represented by `B1`, being near the green robot, and `Act(B1, find(B6))` represents the event of Cassie finding the green robot.

Following, being near, going-to, looking-at and finding are connected by the following KL rule (expressed in SNePSLOG):

```
all(p)(Agent(p)
=> all(o)(Thing(o)
       => {Precondition(Act(p, follow(o)),
                        Near(p, o)),
           Goal(Near(p, o),
                Act(p, goto(o))),
           Precondition(Act(p, goto(o)),
                        Act(p, lookat(o))),
           Goal(Act(p, lookat(o)),
                Act(p, find(o)))}))
```

That is,

- A precondition of an agent's following an object is that the agent is near the object.
- The way to achieve the goal that an agent is near an object is for the agent to go to the object.
- A precondition of an agent's going to an object is that the agent is looking at the object.
- The way to achieve the goal that an agent is looking at an object is for the agent to find the object.

Both `Precondition` and `Goal` take two events as arguments, and form terms representing propositions:

`Precondition(e1, e2)`: The proposition that in order for event `e1` to occur, event `e2` must be occurring.

`Goal(e1, e2)`: The proposition that the way to achieve the goal that `e1` occurs is to get `e2` to occur.

Actions and sequences and other structures of actions are represented and implemented in SNePS using the SNePS Rational Engine (SNeRE) (Kumar 1996), which also allows for action in the service of reasoning and reasoning in the service of action (Kumar & Shapiro 1994). `Precondition` and `Goal` propositions are used by the SNeRE executive.

- If Cassie is to perform an act `a` and the proposition `Precondition(e, Act(B1, a))` is in or is inferable from the KL knowledge base, then before performing the act, Cassie must achieve that event `e` occurs.
- If Cassie is to achieve that event `e` occurs and the proposition `Goal(e, Act(B1, a))` is in or is inferable from the KL knowledge base, then Cassie performs the act `a`.

(The individual constant `B1` is not built into SNeRE. Instead, SNeRE uses whatever term is the value of the variable `*I`. See below for a discussion of `*I`.)

Actions (and, by extension, the acts they are the actions of) may be *primitive* or *composite*. Composite acts are decomposed by the SNeRE executive. If Cassie is to perform the composite act `a`, and the proposition `Plan(a, b)` is in or is inferable from the KL knowledge base, then Cassie, instead, performs `b`. Presumably, `b` is a primitive act which constitutes a *plan* for accomplishing `a`. The FEVAHR knows of two such plans:

```
all(o)(Thing(o) => Plan(lookat(o), find(o)))
all(a)(Agent(a)
=> Plan(help(a),
        snsequence(talkto(a), follow(a))))
```

The primitive action `snsequence` is provided by SNeRE, and performs its argument acts in order. For other primitive actions provided by SNeRE, see (Shapiro & The SNePS Implementation Group 1998).

Terms denoting primitive actions are grounded by aligning them with PMLa functions, which call PMLw functions. PMLa functions do a bit more than just call their PMLw versions. What else they do will be discussed in a later section.

For Cassie to perform an act represented by some primitive act term, the PMLa function aligned with the action term is called, and given the argument terms as its arguments. For example, the SNePS term `find` is aligned with the PMLa function `findfun`. So `find(B6)` is performed by calling the Common Lisp function `findfun` on the argument `B6`. `findfun` retrieves the description aligned with `B6`, and calls the function `WORLD:FIND-OBJ` with that description as its argument.

The primitive action `goto` is aligned with the PMLa function `gofun`. `gofun` works by calling `WORLD:GOTO` on the value of `*STM*`. Thus the FEVAHR goes to whatever it is looking at, but since looking at an entity is a precondition for going to it, and looking at an entity is achieved by finding it, and finding an entity results in `*STM*` being set properly, it all works correctly. `*STM*` is a key link connecting vision (finding/looking) with action (going), and both of them with language and reasoning. This assumes, of course, eye-body coordination—that the FEVAHR can successfully go to what it is looking at. This is the responsibility of the robotics folks and the various simulation implementors.

Thus, symbols representing colors, objects, and categories of objects are grounded in perception, while symbols representing actions are grounded in behavior, and behavior is directed at the correct objects by eye-body coordination.

First-Person Privileged Knowledge

Knowledge of what one is physically doing does not have to be obtained by reasoning. For example, although you might have to reason to decide that I am currently sitting down, I do not. This is called "first-person privileged knowledge." Cassie acts by executing the PMLa function associated with an action term. This PMLa function creates the SNePS term that represents Cassie's belief that she is doing the act, adds this term to the KL knowledge base, and gives it to the English generator for output. (These are some of the additional operations performed by PMLa functions mentioned above.) Since the belief is created by the act itself, it is justified true belief, *i.e.,* knowledge, and since it is created by Cassie's own action, it is first-person priviliged knowledge.

The proposition that an agent `p` performs some action `a` on some object `o` at some time `t` is represented by a functional term (*see* (Shapiro 1993)) of the form `Occurs(Act(p, a(o)), t)`. For example, the proposition that Cassie found the green robot at the time represented by `B13` is represented by `Occurs(Act(B1, find(B6)), B13)`. Recall that `find(B6)` is performed by calling `findfun` on the argument `B6`. The operations that `findfun` performs include creating `B13` to represent the time of finding (see below), and creating, asserting, and expressing `Occurs(Act(B1, find(B6)), B13)`.

The Representation and Use of Indexicals

Indexicals are words whose meanings are determined by the occasion of their use, such as "I", "you", "now", "then", "here", and "there". Cassie understands and can use a set of indexicals with the aid of a triple, ⟨`*I`, `*YOU`, `*NOW`⟩, of values, based on the "Deictic Center" of (Duchan, Bruder, & Hewitt 1995):

`*I` is the SNePS term that represents Cassie, herself;

`*YOU` is the SNePS term that represents whomever Cassie is currently talking to;

`*NOW` is the SNePS term that represents the current time.

The input analyzer interprets first person pronouns to refer to `*YOU`, interprets second person pronouns to refer to `*I`, and interprets "here" to refer to the location of `*YOU`. Similarly, the generator uses first person pronouns to refer to `*I`, uses second person pronouns to refer to `*YOU`, and uses the value of `*NOW` to help determine the tense of generated sentences.

The following shows the use of indexicals by the ASCII version of the FEVAHR.

```
 : Come here.
The FEVAHR is going to WORLD:STU
 I came to you.
 I am near you.

 : Who am I?
 you are a person
and your name is ‘Stu’.

 : Who have you talked to?
 I am talking to you.

 : Talk to Bill.
The FEVAHR is starting to talk to WORLD:BILL
 I am talking to you.

 : Come here.
The FEVAHR is looking for something
    that’s BLUE and a SQUARE.
The FEVAHR found WORLD:BILL.
 I found you.
The FEVAHR is looking at WORLD:BILL.
 I am looking at you.
The FEVAHR is going to WORLD:BILL
 I came to you.
 I am near you.

 : Who am I?
 you are a person
and your name is ‘Bill’.

 : Who are you?
 I am the FEVAHR
and my name is ‘Cassie’.

 : Who have you talked to?
 I talked to Stu
and I am talking to you.
```

Notice that

- Cassie's interpretation of "here" and "I" depend on who is talking to her.
- Cassie addresses whomever she is talking to as "you," but refers to Stu as "Stu" when talking to Bill.
- Cassie understands that when Stu or Bill use "you" they mean her, and she has beliefs about herself which she expresses using "I."
- Cassie uses present tense when reporting who she is currently talking to, but past tense to report past in-

stances of talking, even though those instances were reported in the present tense while they were occurring.

The interpretation of indexicals is done by the analysis grammar, and the generation of indexicals is done by the generation grammar. The SNePS representation is not affected. B5 is the interpretation of "I" when Bill says "I", and the interpretation of "Bill" when Stu says "Bill." When Cassie wants to refer to the individual represented by B5 to Stu, she uses the referring expression "Bill", and when she wants to refer to him to Bill, she uses "you."

A Personal Sense of Time

As mentioned above, Cassie's deictic center includes the variable *NOW, which always contains the SNePS term that represents the current time. We use the relations AFTER and DURING to relate times (we may use additional temporal relations in the future), so Cassie can have beliefs about what she is doing *vs.* what she did in the past, and so she can have beliefs about the temporal ordering of her past actions. The question is when should *NOW move? The simple answer is whenever Cassie acts, but how *NOW moves is more involved.

We categorize Cassie's actions into punctual actions and durative actions, as well as into several modalities. Based on the Garnet simulation, we consider finding, going, and stopping to be punctual actions (but this presents a problem for the Nomad version—see below), and talking, looking, and following to be durative actions. Finding, going, looking, and following are all in one modality, while talking is in another modality. (See Table 2.) An action in one modality must interrupt another action in the same modality that is directed at another object (Cassie must stop looking at Stu in order to find Bill, but not in order to go to Stu.), but needn't interrupt an action in the other modality (Cassie can continue talking to Stu while going to a red robot). We classify stopping in the same modality as looking and following, because for the current FEVAHR those are the durative actions that stopping stops. (*See* (Crangle & Suppes 1994, pp. 159–172) for a more involved discussion of the problems of "saying 'stop' to a robot.")

	punctual	**durative**
speech/hearing		`talkto`
vision/motion	`find` `goto, stop`	`lookat` `follow`

Table 2: Categorization of actions

```
 : Who are you looking at?
 I am looking at you.

 : Come here.
The FEVAHR is going to WORLD:STU
 I came to you.
 I am near you.

 : Who are you looking at?
 I am looking at you.

 : Find Bill.
The FEVAHR is looking for something
    that's BLUE and a SQUARE.
The FEVAHR found WORLD:BILL.
 I found Bill.
The FEVAHR is looking at WORLD:BILL.
 I am looking at Bill.

 : Who are you looking at?
 I looked at you
and I am looking at Bill.

 : Who are you talking to?
 I am talking to you.

 : Follow a red robot.
The FEVAHR is looking for something
    that's RED and a CIRCLE.
The FEVAHR found WORLD:REDROB-2.
 I found a red robot.
The FEVAHR is looking at WORLD:REDROB-2.
 I am looking at a red robot.
The FEVAHR is going to WORLD:REDROB-2
 I went to a red robot.
 I am near a red robot.
The FEVAHR is following WORLD:REDROB-2
 I am following a red robot.

 : Who are you talking to?
 I am talking to you.

 : Who am I?
 you are a person
and your name is 'Stu'.

 : Stop.
The FEVAHR is stopping.
 I stopped.

 : Who are you looking at?
 I looked at you
and I looked at Bill
and I looked at a red robot.

 : Who are you following?
 I followed a red robot.

 : Who are you talking to?
 I am talking to you.
```

The movement of *NOW depends on whether the action to be done is punctual or durative:

punctual: The action is performed at a new time which is AFTER the curent *NOW; an additional time term is created which is AFTER the time of the act;

and `*NOW` is moved to this latest time.

durative: The action is performed at a new time which is `AFTER` the curent `*NOW`; an additional time term is created which is `DURING` the time of the act; and `*NOW` is moved to this latest time.

In addition, if any durative action was being done before this action was started, and that action is either in a different modality from this one, or is directed to the same object as this one is, then the new `*NOW` is also asserted to be `DURING` the time of that action.

All the operations discussed in this subsection are carried out by the PMLa functions, and this completes the discussion of what the PMLa functions do. To summarize, each PMLa function: calls the appropriate PMLw function to effect the action; if appropriate, changes the values of `*YOU` and `*STM*`; creates one or more new time terms, creates propositions relating them, and updates the value of `*NOW`; creates a proposition that Cassie has done (or is doing) the action, and generates an English expression of that proposition.

Low-Level Bodily Awareness

Although the creation of action beliefs by PMLa functions captures the notion of first-person privileged knowledge, it is not good enough to capture the timing of durative actions for the Nomad robot version of the FEVAHR, nor for any version for which PMLc and lower levels operate as processes separate from and asynchronous to the KL and PMLa processes. In those cases, the PMLa functions and the PMLw functions they call only initiate the actual bodily actions; they terminate before the bodily actions are complete. Moreover, `find` and `goto`, which I categorized as punctual actions above, are durative actions for the real robot. When Cassie says "I went to Bill" it is because the PMLa function responsible for going to Bill has executed, but the robot, under control of the SAL is probably still moving across the floor. To solve this problem, and make Cassie more accurately aware of what her body is doing, we must have feedback all the way from the SAL to the KL.

The Nomad robot version of the FEVAHR uses vision to avoid obstacles on its path. Therefore, while it is going to Bill or any other object as a result of a specific command, it may look at and avoid other objects in its way. Feedback from the SAL to the KL is needed to make the FEVAHR "consciously aware" of these objects, so that, for example, it could accurately answer the questions "What are you looking at?" or "What have you looked at."

In addition, the current domain rules for FEVAHR actions assume that a sequence of actions such as is illustrated here (with the "`The FEVAHR ...`" print-outs deleted to save space):

```
: Go to the green robot
  and then go to Bill and help him.
I found the green robot.
I am looking at the green robot.
I went to the green robot.
I am near the green robot.
I found Bill.
I am looking at Bill.
I went to Bill.
I am near Bill.
I am talking to you.
I came to you.
I am near you.
I am following you.
```

may be performed as quickly as the PMLa and PMLw functions allow. When this command was given to the hardware robot, the PMLa functions executed so quickly that we didn't even notice it making any movement toward the green robot. It seemed to skip the first command, and just immediately go to Bill. This must be changed so that subsequent actions in a sequence are performed when and only when SAL feedback and sensory actions indicate that the earlier actions have been completed. It is also clear from this ASCII simulation that Cassie should not perform an action whose goal is already accomplished.

We began to address these problems during the final stages of the Nomad FEVAHR project, and will continue as we transfer the embodied Cassie to a new domain.

UXO Remediation

We are currently in the process of transferring the FEVAHR implementation to a robot that will perform the "unexploded ordnance (UXO) remediation" task. This robot will operate on a rectangular field in which are some unexploded land mines, or other ordnance. The robot will have to find a UXO, and either carry it to a drop-off place at a corner of the field, or set a charge on it to blow it up, meanwhile moving to a safe place. The robot will also have to sense when its batteries are low, and then interrupt what it is doing to go to a recharge station and recharge them. The robot will have to be able to report at any time what it is doing and where it is in the field. As with the FEVAHR, we will develop the KL and PMLa levels of the UXO robot using simulations of the lower levels and of the world, and then transfer these upper levels to real robot hardware.

Summary

My colleagues and I have been engaged in a series of projects in which Cassie, the SNePS cognitive agent, has been incorporated into a hardware or software-simulated cognitive robot. In this paper, I gave an overview of these projects, and discussed some of the ways embodiment influences the creation and meaning of the agent's beliefs and other terms in its knowledge base.

The knowledge base resides at the Knowledge Level (KL), the locus of "conscious" reasoning. All the entities the robot can think about and discuss are represented at the KL by terms of SNePS, the KR&R system used to implement the KL. Object, property, and category terms are grounded by aligning them with sub-KL descriptions that are used by a real or simulated vision system to locate objects in the real or simulated world, and place sub-KL representations of them in `*STM*` a variable that is the robot's short-term iconic memory. Action terms are grounded by aligning them with sub-KL functions that carry out the represented actions, using `*STM*` and/or the hardware vision system to effect eye-body coordination. Action functions insert into the knowledge base beliefs that they are being done, so that the robot has first-person privileged knowledge of its actions. Action functions also insert into the knowledge base terms representing the times the actions are done, and beliefs that give temporal relations among the times, providing the robot with a personal sense of time. The robot can understand and use indexical words by using a deictic center of variables(⟨`*I`,`*YOU`,`*NOW`⟩) containing the terms representing: the robot itself; the person it is talking with; and the current time. The latter two variables are updated by the action functions, so the robot's sense of its place in the world comes from its embodied aspect rather than from thinking about it.

Although the sub-KL symbols and functions we have been using do capture these notions of embodiment, we need to give the KL feedback from even lower bodily levels of the robot architecture, so that the robot has a better idea of what it is doing, and when it has accomplished its tasks.

Acknowledgements

This work was supported in part by NASA under contracts NAS 9-19004 and NAS 9-19335, and in part by ONR under contract N00014-98-C-0062. I appreciate the comments made by Bill Rapaport, Debra Burhans, Fran Johnson, and Haythem Ismail on earlier drafts of this paper.

References

Crangle, C., and Suppes, P. 1994. *Language and Learning for Robots.* Stanford, CA: CSLI Publications.

Duchan, J. F.; Bruder, G. A.; and Hewitt, L. E., eds. 1995. *Deixis in Narrative: A Cognitive Science Perspective.* Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

1993. Garnet reference manual, version 2.2. School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Harnad, S. 1990. The symbol grounding problem. *Physica D* 42:335–346.

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., eds., *Proceedings of The Sixth Florida AI Research Symposium (FLAIRS 93).* The Florida AI Research Society. 325–329.

Hexmoor, H. H. 1995. *Representing and Learning Routine Activities.* Ph.D. Dissertation, Department of Computer Science, State University of New York at Buffalo, Buffalo, NY. Technical Report 98-04.

Kumar, D., and Shapiro, S. C. 1994. Acting in service of inference (and *vice versa*). In Dankel II, D. D., ed., *Proceedings of The Seventh Florida AI Research Symposium (FLAIRS 94).* The Florida AI Research Society. 207–211.

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

Lammens, J. M.; Hexmoor, H. H.; and Shapiro, S. C. 1995. Of elephants and men. In Steels, L., ed., *The Biology and Technology of Intelligent Autonomous Agents.* Berlin: Springer-Verlag, Berlin. 312–344.

Lehmann, F., ed. 1992. *Semantic Networks in Artificial Intelligence.* Oxford: Pergamon Press.

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

Shapiro, S. C., and Rapaport, W. J. 1991. Models and minds: Knowledge representation for natural-language competence. In Cummins, R., and Pollock, J., eds., *Philosophy and AI: Essays at the Interface.* Cambridge, MA: MIT Press. 215–259.

Shapiro, S. C., and Rapaport, W. J. 1992. The SNePS family. *Computers & Mathematics with Applications* 23(2–5):243–275. Reprinted in (Lehmann 1992, pp. 243–275).

Shapiro, S. C., and The SNePS Implementation Group. 1998. *SNePS 2.4 User's Manual.* Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY.

Shapiro, S. C.; McKay, D. P.; Martins, J.; and Morgado, E. 1981. SNePSLOG: A "higher order" logic programming language. SNeRG Technical Note 8, Department of Computer Science, State University of New York at Buffalo. Presented at the Workshop on Logic Programming for Intelligent Systems, R.M.S. Queen Mary, Long Beach, CA, 1981.

Shapiro, S. C. 1982. Generalized augmented transition network grammars for generation from semantic networks. *The American Journal of Computational Linguistics* 8(1):12–25.

Shapiro, S. C. 1989. The CASSIE projects: An approach to natural language competence. In Martins, J. P., and Morgado, E. M., eds., *EPIA 89: 4th Portugese Conference on Artificial Intelligence Proceedings, Lecture Notes in Artificial Intelligence 390.* Berlin: Springer-Verlag. 362–380.

Shapiro, S. C. 1993. Belief spaces as sets of propositions. *Journal of Experimental and Theoretical Artificial Intelligence (JETAI)* 5(2&3):225–235.