# Commonsense and Embodied Agents
# A Panel Discussion

Stuart C. Shapiro
Computer Science and Engineering
University at Buffalo, The State University of New York
Buffalo, NY, USA
shapiro@cse.buffalo.edu

Eyal Amir
Computer Science
Stanford University
Stanford, CA, USA
eyal@cs.stanford.edu

Henrik Grosskreutz
Computer Science V
Aachen University of Technology
Aachen, Germany
grosskreutz@cs.rwth-aachen.de

David Randell
Electronic & Electrical Eng.
Imperial College
London, UK
d.randell@ic.ac.uk

Mikhail Soutchanski
Computer Science
University of Toronto
Toronto, Canada
mes@cs.toronto.edu

## 1 Introduction

The purpose of this panel is to discuss the implications for commonsense knowledge representation and reasoning of implementing a robot/agent with all or most of the following characteristics:

- The agent can act.
- The agent can perceive.
- The agent can communicate with humans.
- The agent does first-person reasoning: the knowledge base and reasoning are the beliefs and reasoning of the agent, rather than about it.
- The agent does on-line reasoning: perceiving, reasoning and communicating with humans is performed while the agent is acting, rather than before it begins acting.

We are not distinguishing the terms "embodied agent" and "cognitive robot."

The panelists were chosen from among the expected attendees of Commonsense 2001 as researchers who have had experience implementing embodied agents, or as representatives of research groups that have implemented embodied agents. The research groups represented by the panelists (whether they, themselves, are implementors or theorists), each at the panelist's respective university, are:

| | |
|---|---|
| Amir: | Formal Reasoning Group |
| Grosskreutz: | Knowledge-based Systems Group |
| Shapiro: | The SNePS Research Group |
| Randell: | Cognitive Robotics, Intelligent and Interactive Systems |
| Soutchanski: | Cognitive Robotics Group |

# 2 Demographics

## 2.1 General Characteristics

The following table summarizes the characteristics of the robots the panelists are discussing.

| Issue | Amir | Grosskreutz | Randell | Shapiro | Soutchanski |
|---|---|---|---|---|---|
| Implementation | | | | | |
| Hardware | Nomad 200 | RWI B21 | Kephera, LinuxBot | Nomad 200 | RWI B21 |
| Software | Yes | Yes | No | Yes | Yes |
| Perception | | | | | |
| Sonar | Yes | | Yes | Yes | Yes |
| Bumpers | Yes | Yes | | Yes | |
| Vision | | | Yes | Yes | |
| Laser range finder | | Yes | | | Yes |
| Infra-red | | | Yes | | |
| Odometry | Yes | | | | |
| Proprioceptors | | | Yes | Yes | |
| NL Input | No | | No | Yes | No |
| NL Generation | No | | No | Yes | No |
| First-Person | Yes&No | No | Yes | Yes | Yes |
| On-Line | Yes | Yes | No | Yes | Yes |

## 2.2 Tasks

The following paragraphs give the tasks the robots are undertaking:

**Amir:** Our robot, LiSA, is doing obstacle avoidance and navigation in an office environment. It is implemented in C++ and Prolog, running on a mobile Nomad 200 robot. We have also used a simulated environment in which we allowed LiSA to use the elevators in the building in its navigation between offices. It uses a subsumption architecture with 5 layers, all implemented using the PTTP theorem prover. The layers run in parallel and communicate using TCP/IP, sending axioms to each other, and receiving information from the sensors as axioms.

**Grosskreutz:** Our research is concerned with the high-level control of mobile service robots, which are to perform tasks in typical, unmodified human environments. In particular, we work on the control of an RWI B21 robot, which is operating in one floor of our computer science department. The robot's jobs are delivery tasks like delivering letters from one office to another, or serving coffee. In order to safely travel in its environment, our robot makes use of the basic-task control system beeSoft, which has successfully been used in the museum-tourguide-projects RHINO and MINERVA. The beeSoft system consists of different modules, which among other things allow the robot to safely travel in its environment, avoid collisions with persons or objects, and allow the robot to keep track of its current position.

**Shapiro:** We refer to all instances of our computational agent as "Cassie." The robot we implemented in hardware is Cassie in the role of a "Foveal Extra-Vehicular Activity Helper-Retriever (FEVAHR)." Cassie, the FEVAHR, was implemented on a commercial Nomad robot, including sonar, bumpers, and wheels, enhanced with a foveal vision system consisting of a pair of cameras with associated hardware and software. There have also been several software simulated versions of the FEVAHR. FEVAHR/Cassie operates in a $17' \times 17'$ room containing: Cassie; Stu, a human supervisor; Bill, another human; a green robot; and three indistinguishable red robots. Cassie is always talking to either Stu or Bill—taking statements, questions, and commands from that person (all expressed in a fragment of English), and responding and reporting to that person in English. Cassie can be told, by the person addressing her, 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. FEVAHR/Cassie is able to understand and use deictic terms including "I," "you," "come," and "go," by maintaining a representation of itself, and a belief about whom it is conversing with. It maintains a concept of personal time, including a `NOW` pointer which is updated whenever it acts, and which is used to determine the tense and aspect of sentences it utters to report on its current and past actions.

We have also implemented a modification of FEVAHR/Cassie as a robot that cleans up unexploded ordnance (UXO remediation). This Cassie has only existed as a software simulation. The UXO-clearing-Cassie exists in an area consisting of four zones: a safe zone; an operating zone that possibly contains UXOs; a drop-off zone; and a recharging zone. The UXO-clearing-Cassie contains a battery that discharges as she operates, and must be recharged in the recharge zone as soon as it reaches a low enough level. She may carry charges to use to blow up UXOs. Her task is to search the operating zone for a UXO, and either blow it up by placing a charge on it and then go to a safe place to wait for the explosion, or pick up the UXO, take it to the drop-off zone, and leave it there. The UXO-clearing-Cassie has to interrupt what she is doing whenever the battery goes low, and any of her actions might fail. (She might drop a UXO she is trying to pick up.) She takes direction from a human operator in a fragment of English, and responds and reports to that operator. There is a large overlap in the grammars of FEVAHR/Cassie and the UXO-clearing-Cassie.

**Randell:** The first project was implemented on Kephera robots connected to networked PCs via "umbilical" cords. These ran around a small model of a simple office. The model had rooms connected via doorways. The doors could be open or closed. The robot could do two things:

1. route finding from one room to another, and re-planning if necessary where an expected open door was found closed;
2. map-building, i.e. constructing a logical model of the connectivity of the rooms of the office layout.

The robots embodied a suite of infra-red proximity sensors and used odometry information. From a robotics standpoint what the robots were able to do was arguably not impressive. But this was not the key point; rather it was how this was done. The robots exploited an abductive planner. The unit of representation was a logical sentence, and the unit of computation a resolution proof step.

The second project (ongoing) is implemented on LinuxBots using a wireless network link. Embodied senses include vision (stereo), a suite of sonar proximity sensors and proprioceptors. The intended working environment is a restricted but real-world office environment, with the robots acting upon and interpreting visual information for planning and navigation tasks.

**Soutchanski:**

- The robot solves planning tasks, determines schedules for actions included in a plan and executes computed plans. In particular, implementations consider an office delivery task in the real office environment.
- The primitive actions included in plans are axiomatized in the situation calculus.
- Natural constraints on the planning tasks are expressed in terms of Golog programs (Golog is a logic-based high-level programming language). Golog has all standard programming constructs and several non-deterministic operators, it can be used to compose complex controllers from primitive actions specified in the situation calculus. The planning tasks are formulated as entailment problems: thanks to the constraints provided by Golog programs, solving these problems is computationally efficient. The implementations follow two different perspectives: deterministic (all primitive actions and durations of processes initiated by actions are deterministic) and probabilistic (actions of the robot are stochastic and the interaction of the agent with the environment is modeled by a Markov Decision Process).
- *Deterministic perspective.* In this case, the uncertainty and dynamics of the environment can be accounted only by sensing the real outcomes of actions executed by the agent, by determining possible discrepancies between the observed outcomes and the effects expected according to the logical model of the world and then by recovering, if necessary, from the relevant discrepancies: to recover the agent reasons on-line to find an appropriate correction of the program that is being

executed. The implementation has demonstrated that given a non-deterministic Golog program with actions that have an explicit temporal argument, the monitor computes on-line an alternative schedule of deliveries in a real office environment, when the robot encounters unmodeled obstacles that unexpectedly delay its traveling between offices.

- it Probabilistic perspective: the robot solves decision-theoretic planning problem. By using the domain specific procedural knowledge expressed in the Golog program, an off-line DT-Golog interpreter computes an optimal policy given a reward function and the situation calculus based representation of an MDP. This gives to the programmer the unique flexibility in designing complex controllers: both the opportunity of designing efficient controllers and the simplicity of leaving the solution of the problem of what constitutes an optimal policy to the off-line interpreter. The paper on DT-Golog programming outlines a robotics implementation and provides arguments that the proposed approach allows for the seamless integration of programming and planning. The subsequent paper introduces a new *on-line* DT-Golog interpreter that combines planning with the execution of policies. This on-line architecture allows one to compute an optimal policy from an initial segment of a Golog program, execute the computed policy on-line and then proceed to computing and executing of policies for the remaining segments of the program.
- All implementations rely on the low-level software that was developed by the team from the University of Bonn and CMU. This software provides collision avoidance, navigation, localization and other functionalities.

# 3 Discussion

## 3.1 Introduction

This section contains the answers the panelists have provided to a set of questions posed by the panel chair. Not every panelists answered every question.

## 3.2 In general, what are the implications of implemented embodied agents for the logical formalizations of commonsense reasoning?

**Amir:** There are several implications in my opinion:

1. I view the logical theories that I write using a behavioral perspective. I build the theory as an engineer: build a prototype, test it, change/correct, test again, change/correct, and so on. This corresponds (roughly) to the "microworlds" approach for building logical theories, but starting with a theory that resembles the commonsense situation as much as possible/appropriate.
2. There are many practical problems that arise when trying to create an implemented embodied agent:
   (a) Strategies for reasoning are very important. They have a dramatic effect on the time taken for reasoning. This is critical in on-line reasoning systems and embodied reasoning systems.
   (b) Theorem provers (or other reasoning engines) must be simple and easy to change/augment. Typicaly, it is not difficult to find theorem provers that suit most of the needs for a particular system. However, it seems that many times one still needs a few "extra features" (e.g., semantic attachments, linking to other applications, etc). Ease of change of the chosen theorem prover is very important in that case.
   (c) Errors in sensors must be treated and cannot be ignored. If the robot's sonars are not reliable (as sonars usually do), then the system may have to spend extra reasoning cycles compensating for those. Example of such reasoning effort would be taking into account neighboring sonars or previous state.

3. Several philosophical and methodological issues that arise:
   (a) When you prove "Move(X,Y)", what does that mean with respect to your system/ontology. Does that mean that the robot "is going to move to X,Y", or that "should move to X,Y" or others. Notice that logic is typically viewed as making assertions about the world, so saying that "Move(X,Y)" is a 'command' to the robot does not fit this view.
   (b) Connecting symbols to sensors grounds your symbols and gives them a particular *practical* meaning, rather than philosophical or ontological meaning that may be open to arguments. For example, an obstacle is exactly its meaning in your reasoning system. The characteristics of this object are determined by the static theory and the sensory information.
   (c) Issue specific to subsumption architectures: Each layer has a very limited view of the world. Also, these views of the world are very different from each other. How do I design layers and how do I choose the level of abstraction/focus for each layer? Also, those views may be inconsistent. Should we worry about that?
4. Forces us to confront larger goals:
   (a) Include learning mechanisms
   (b) Incorporate commonsense into sensors, and collaboration between sensors and knowledge bases.
   (c) Build knowledge bases that can account for many tasks/capabilities.
5. Theories for reasoning in an agent can be stated in first person or third person. In a comprehensive system that is capable of many tasks, some theories are high level and some are low level. Such systems need to be able to use both kinds of theories and possibly combine them.

**Grosskreutz:** First of all, while trying to implement an embodied agent like a mobile robot, we realized that the high-level reasoning and acting component of the robot should not directly operate the robot's sensors and effectors. A robot has to fulfill a variety of tasks like navigation, collision avoidance and position estimation, and many of these tasks are better dealt with using specialized (non-logical) methodologies than using the reasoning component for everything. In particular, the amount of information provided by physical sensors like a laser-range-finder or a camera is usually too large to be handled by the reasoning component. Instead, sensor information has to be pre-processed by specialized processes which significantly reduce the amount of information processed by the reasoning component.

These considerations naturally lead us to a layered robot control architecture, where specialized low-level processes like a navigation process control the robot's physical sensors and effectors, and where a high-level controller located on top of the low-level processes takes care of the robot's long-term goals. Within this layered control architecture, we identify the "reasoning" with what happens inside the high-level controller.

As a result of such an architecture, the high-level controller (which is only causing change in the world by interacting with the low-level processes) should be able to represent and reason about low-level processes. That is, it has to be able to reason about processes like a navigation or a door state estimation process, which have several peculiarities. In particular, they have a temporal extent, and they can be run concurrently. For example, while traveling from one room to another, a robot can still take camera images, inspect the opening angles of doors, or play sound files (i.e. "talk" to persons). Thus, such processes can not adequately be represented as atomic actions, which is often the way robot "actions" are represented in a reasoning framework. The robot's reasoning component should have a notion of time, and should have a notion of the set of active low-level processes. Additionally, the reasoning component should be able to represent the uncertainty involved in real robot control. Most low-level processes have noisy effects, and as these effects can not always be abstracted away from, the reasoning component should be able to deal with (probabilistic) uncertainty. Additionally, the reasoning component should be able to represent the fact that the occurrence of actions can have ontologically different causes. Some actions are executed by the robot, while other actions are exogenous, and the reasoning should be able to represent and reason about both robot actions and exogenous actions.

Finally, an embodied agent should act in real-time, meaning that the reasoning done by the high-level controller should be fast, and thus the high-level controller should only try to deduce "simple" queries from the logical theories. This does not mean that the logical theories have to be simple; instead, as argued above, they should be quite realistic, much more realistic than classical STRIPS descriptions. However, the reasoning done using such theories has to be limited in a way that the agent's reaction or decision time is tolerable in real environment.

**Randell:** The implications fall into several related but distinct categories:

1. ontological
2. computational and implementational
3. cognitive and epistemological
4. control
5. reasoning
6. symbol grounding

For (1) ontological: this relates to what is represented in the domain model, how the domain is carved up, assumed primitives and defined categories, questions on the sufficiency of the theory to account for this, which bears on both (2) and (3) given that with any implemented theory a tight coupling exists between what is factored out and represented and what can be done with it.

The central point for an embodied agent is rooted in its egocentric or deictic viewpoint. It now needs knowledge/beliefs about itself as well as the domain its embedded in, e.g: size, shape, mobility and connectivity of its constituent parts. It needs knowledge about causality, other objects, its own actions on objects and its integrated senses and the dominance of these. The restricted viewpoint brings to the fore the need to deal with partial models and explanations, ambiguity, a sense of self and others and location, of the use and role of indexicals in its communication language (I, you, we, they, this, that, here, there, . . . ) and the relationship between them. Arguably an embodied agent also needs an explicit representation of its perceptual acts, its sensori-motor functions and actions and bodily awareness, and what it can and cannot do. The key here is the necessary localisation of the agent in its environment, the perceived objects and their causal impact on the embodied agent.

(2) computational: abstract structures derived from (1) can be exploited computationally, e.g. lattice structures (sort/relational) used in semantic networks. The decision whether or not to use these depends on how the metalevel structures are to be reasoned about, if at all. Implementation issues also relate to the correctness of the program and the underlying formal theory and its semantics. The computational question also relates to (4) control

(3) cognitive and epistemological: this relates to questions on cognitive adequacy (i.e. does the theory account for the way we habitually describe the domain? Does it highlight the same or similar objects, relationships and events that we single out?) It also bears on questions of epistemic priority and the order of experience and knowledge (van Benthem), e.g. that bodies are arguably immediate in terms of our sense perceptions and points derived from abstractions on these (c.f the interest in and sufficiency of primitive observational languages) which contrasts with the order of knowledge that can go the other way, e.g. in geometry it has been from points to bodies.

(4) control: this impacts on the interleaving of perception, reasoning and controlling and monitoring sequences of actions and how these are implemented. An embodied agent needs to be able to weight the stream of information coming in via its senses, c.f. Maslow's hierarchy of needs? Low level processing should only impact on high level plans if the execution of the plans are likely to be thwarted. But how does the agent come to know this? This is where (in)consistency of beliefs comes in, i.e. if the incoming information is consistent with the goal and plan, carry on, otherwise replan and focus on the low level information. But do we need to explicitly represent all the theory, including e.g. the control aspects? The sense-plan-act cycle [Shanahan et al.] has been criticised [Shapiro et al.] for not forming part of the logic and being explicitly represented, i.e. how does the robot know it has completed an

action? But is this necessary other than using the logic to formally specify the relationship between the subsystems?

(5) reasoning: the central questions are, what needs to be reasoned about and at what level does/should reasoning take place, and what does the embodied agent add to this? Reasoning by an embodied agent is clearly time-bound c.f. reasoning about an abstract geometry problem using a theorem prover. This places reasoning firmly in a pragmatic context, e.g. if no answer is deduced within say 5 seconds, terminate—answer unknown, and perhaps garner more information from the environment by moving. Similarly, the embodied agent ties reasoning to its senses, location and mobility. If for example, I think some object may lie behind another, but it's a long distance away, it may not be worth while moving to it to resolve the query; if another agent is close by, ask it. Reasoning about sensory information seems generally not done unless ambiguity/inconsistency arises. We can reason through sets of actions and come to an explanation of these, but do not necessarily have transparent access to these—this relates to control (4) questions. Should the embodied agent be able to equally reason about its sensory states as well as the sequence of actions in order to execute a plan?

(6) symbol grounding: This is an immediate and direct consequence of using an embodied agent. In a software simulation the map between the tokens in the formal language and the objects they denote is stipulated (as it is for any model used to interpret a logical theory). One question that arises is, to what extent does meaning depend on the network of associations between tokens in the embodied formal theory? In a formal theory the possible models are constrained by the axioms and structure, and the consequence class of deductions for the theory. If now the implemented theory reduces the explicit structures used, do the models satisfying it increase and the meaning become less clear cut? Or is it physics (and the physical world) that providing the model, collapses the set of possible meanings? Is the requirement for a complex formal theory a red-herring when it comes down to embodied agents?

**Soutchanski:** There are several implications. I'll mention a few.

- Complexity of the real world should be taken seriously in logical formalizations. In particular, one has to acknowledge that there are other agents (people, nature, devices, etc.) in the world who behave independently: some of their actions are not observable, and the robot can learn about effects of their actions only from the sensory input it receives. In addition, if certain aspects of the world are not captured in a logical formalization, the robot must rely on sensory information to augment its current logical model. Additional complication arises because of imprecision in effectors of the robot: if the robot starts performing an action, it may result in different outcomes because effectors are not perfect.
- Because sensing is essential for successful functioning of the robot, those terms in the logical formalization which represent sensory data get their denotations at the run time from the real world and not from interpretations that one may consider for a given logical theory.
- A large-scale formalization of commonsense reasoning (about solid objects, liquids, different materials, space, etc.) in predicate logic is necessary to design controllers for autonomous embodied agents.

## 3.3 How does embodiment affect where beliefs come from?

**Amir:** Beliefs in each layer in a subsumption architecture come from several sources:

1. Beliefs that the system had initially for that layer. For example, "the result of moving to place X is that the robot is in place X."
2. Beliefs that result from sensory information (temporarily). For example, "the robot is at position X,Y at the moment."
3. Beliefs that are received by the layer from higher layers (temporarily). For example, "there is an object to your left at distance 5 meters."
4. Beliefs that result from applying defaults to the current knowledge of the layer. For example, "There are no objects in this room besides objects A,B and C."

5. Beliefs that are maintained over time (currently, our system supports that, but does not account for belief change in its semantics). For example, "The robot's previous location was room 218."

The beliefs in 2–5 are updated at every cycle, while the beliefs in 1 are kept static throughout the computation. The defaults mentioned in 4 are also static assumptions. For example, "all the objects in the world are those whose existance I can prove."

**Grosskreutz:** In our system, there are two different sources of beliefs. Initially, the robot starts with prior beliefs about the state of the world, which are asserted by the user. This kind of belief is rather declarative, stating for example which requests have been issued, how probable it is that a door is closed, or how many objects are on a desk. The second source of beliefs is sensor information provided by low-level processes which read physical sensors, and pre-process the sensor input. This information is then used to update the robot's current beliefs. What distinguishes this source of beliefs from the first is that during execution the robot almost never gets declarative assertions about the state of the world, but instead gets indirect information about the state of the world, observations, which are then used to update the robot's declarative beliefs.

**Shapiro:** Our architecture is a three-level one: a mental, "knowledge level" (KL), for the symbolic representation of the agent's beliefs and "conscious" reasoning (in SNePS); and two body levels, one for the implementation of what are action primitives at the KL, and one for control of the sensors and effectors. In this discussion, I will only make the two-way distinction of KL vs. BL (for "body level").

Beliefs may enter Cassie's KL from several sources:

1. Initial domain knowledge when Cassie is initialized.
2. NL input from a human. (Cassie is currently entirely gullible.)
3. Each BL acting procedure inserts into the KL the belief that Cassie is performing that act at the current time.
4. Beliefs about states sensed by Cassie's sensors are inserted into the KL by the BL.

Sources (1) and (2) are standard for computational agents. Sources (3) and (4) are part of our implementation of embodiment. This is especially the case with source (3), which implements Cassie's first-person privileged knowledge of what she, herself, is doing.

**Soutchanski:** In addition to beliefs represented in a logical theory, an embodied agent executes sense actions which may provide information that will lead to reconsidering some of the previously held beliefs. (In particular, sensory data obtained from the real world may ground certain terms in axioms in a way that will exclude some interpretations as possible models of axioms.)

### 3.4 Does embodiment reduce the use of reasoning? If so, explain.

**Amir:** Embodiment both reduces and increases the use of reasoning.

Reasoning is somewhat simpler (computationally) as a result of sensors typically being deterministic. When sensory information is converted to logical sentences, it typically does not result in sentences that include disjunctions.

On the other hand, one sometimes want to reason about the fallacies of sensors so that the robot's reaction to its sensors is more reliable. Thus, those additional reasoning tasks need to be adressed.

Overall, I find that the embodiment increases the need for reasoning and highlights places where reasoning can be used. Every capability that one wishes to add to the system can be approached using another reasoning module. For example, if one wishes the robot to follow another robot, then the robot would need to reason about the whereabouts of the leading robot (e.g., if the leader went around the corner), and also reason that the robot it detects at a particular place is the same robot that was there a second ago.

**Grosskreutz:** I find this question difficult to answer, as it somewhat depends on which non-embodied application an embodied agent is compared to. In our case, embodiment forces us to consider realistic tasks that can be achieved using our robot, and thus to investigate where reasoning can usefully be applied in service of robotics. In this application, then, I would answer that at the actual state of service robotics there are fewer applications of reasoning than I would have expected. The use of reasoning involved in typical applications is relatively small, but this may be due to today's limitation of real physical robots. However, to me the reasoning performed seems to have to take into account a lot of aspects usually ignored, like time, continuous change, or probabilistic effects.

**Shapiro:** Yes.

Object avoidance was performed by FEVAHR/Cassie entirely at the BL, using the sonar and bumpers. No reasoning was involved in this.

There is no need for reasoning to decide that after the agent has begun a durative act, and before it has finished it, it is still performing it, because the BL reports to the KL what it is doing at any time.

Similarly, there is no need for reasoning to decide that perceived states, whether external to the agent or states of the agent's body, are still holding, as long as they are still being perceived. Proprioception provides continuous perception of states of the agent's body.

Finally, beliefs about the effects of actions the agent, itself, performs need not, and should not, be used to reason that they hold after the action is done. Since actions may take some time to perform, and may, in fact, fail, the agent should not believe its effects hold until and unless it perceives them holding. This does not eliminate the need to reason about effects either for planning or for reasoning about the effects of other actors whose performance is outside the perceptual field of the agent.

**Randell:** The brief answer is that it can do so, but not necessarily. It depends in part on what one means by "reduce" here, for a reduction in the use of reasoning may be a reduction in the search space, a reduction in the time to execute a set of actions, or if using a resolution reasoning program, a reduction from hierarchical planning, or a reduction from being able to use a sparser symbolic model of the domain, and so on. The uncontroversial answer to the question is "yes," and vice-versa; but a more accurate answer is more likely that while embodiment does not necessarily reduce the use of reasoning, it does increase (the need for) it.

Take a simple first order monotonic theory about a domain. We have a set of primitive concepts, axioms and definitions. The theory can be characterised by its consequence class of deductions. It has an intended model and many other unintended ones, indeed an infinite number of models. Moreoever, from the theory we also assume we can lift out several metalevel structures, e.g. sort (encoding classification or taxonomic information) and various relational lattices; also possibly sets of useful lemmas, and say explicitly represented information about change, and time in the theory as in a simple envisionment theory. We may well factor out and exploit different structures using different implemented subsystems, but all can be related to the formal theory which guarantees the semantics of the representational structures used, and the embodied valid rules of inference in the theory. Now, let us assume that without these metalevel structures, reasoning becomes intractable and not if not. Has the reasoning reduced? Not necessarily. What has reduced may be the search space, or number of explicitly represented axioms, or the time taken to solve the problem by factoring out these structures and using specalised reasoning subsystems, but none of these actually means an actual reduction in the use of reasoning; rather, if anything the use of reasoning increases but its now at the metalevel. So how does this perspective change where an embodied agent is now introduced? The answer is not at all. So what does embodiment give you?

What embodiment gives you is (i) a locus for reasoning tasks which may or may not require an increase or reduction in reasoning, and (ii) constraints from real world physics. On the one hand the limited viewpoint gives you a naturally restricted viewpoint, hence partial model of the domain, which means less to "think about". But on the other hand it's also an argument for an increased need for, for example, commonsense knowledge about the agent's domain, notions of its own size, shape and the impact on its environment, of understanding causality, the effect of distance and movement and the way this impacts on ones senses, inferring objects and their objective spatial relationships from images, of needing to deal with communication between other agents and accommodating their individual view and knowledge, and so on. All this can be seen as an increase in the need for, rather than a reduction is reasoning. But again, not necessarily so. If by reasoning we are honing in on an analogue of conscious reasoning, then the reasoning can reduce, when from learning, hitherto conscious acts become unconscious. Furthermore, one can also argue for a reduction in the amount of reasoning because an embodied agent can always survey the world when it doesn't know how to act, either because the (unsure) agent deliberately consults the world to see what is the case (rather than working it out from scratch), or because the (indecisive) agent just acts and then sees what happens as a result (rather then trying to plan everything in advance). "If in doubt, muck about" could be a good motto for embodied agents!

**Soutchanski:** Because of the complexity of the real world, embodiment requires more reasoning that an agent needs if it operates in a simulated environment (where all actions achieve their expected effects). For example, if sense actions inform the robot that its recently executed action did not have expected effects, then the robot may need to reason whether this discrepancy can be attributed to action(s) of an external agent, and if yes, it may need to reason how to overcome obstacles created by those action(s).

## 3.5 How does first-person reasoning affect the reasoning task or requirements?

**Amir:** A subsumption architecture needs to reason about the robot in both first-person terms and third-person terms.

From a computational perspective, the reasoning task is somewhat more demanding than reasoning that is robot-implicit (first-person). Formulae that are more complicated and include more functions and variables/terms are typically "heavier" for automated reasoning.

From an ontological perspective, theories that are written in third-person (the robot is an explicit object in the vocabulary of the logical theory) present an "external to the robot" perspective (e.g., "in(robot,room218)"). In contrast, theories that are written in first-person are written in an "internal to the robot" perspective (e.g., "direction(object1)=degrees($\pi$)"). Clearly, these influence the kind of tasks that the theory can support, but also the kind of extensions that it may have.

Another ontological effect is the possibility of talking about the relations between the robot and another agents (humans/robots/inanimate things). For example, the robot may consider saying something to a human if it detected that there is something dangerous in the room, and it decides that it "likes" the human because the human likes it.

**Shapiro:** First-person reasoning can take into account first-person privileged knowledge, specifically what the agent is currently doing, that a separate reasoner might not have access to. However, more significantly, a first-person reasoner cannot have access to beliefs that don't come from the sources mentioned above. So a first-person reasoning agent might not have knowledge about precisely where it is, nor about aspects of the world it has neither sensed nor been told about.

**Soutchanski:** The robot has a direct access only to its own beliefs and to its own sensors; in particular, it does not know what actions other agents do and how an environment changes unless occurrences of these actions/changes predicted in axioms or entailed from axiom and sensory information.

## 3.6 How does on-line reasoning affect the reasoning task or requirements?

**Amir:** Computationally, online reasoning forces the use of strategies in reasoning. Theoretically, one can view a strategy that is detailed enough as a program that computes the solution to the reasoning task. In this sense, strategies are effective and limited at the same time. However, in the case of embodied reasoning tasks, many times the tasks repeat themselves, and the strategies need to be produced/revised only rarely. One may think about devising several strategies for different conditions, employing the strategies as needed.

If by "task" you mean the "query" for the reasoning (i.e., the question is "How does online reasoning affect the query asked from the knowledge base of your reasoning"), then it is helpful if the queries are directly related to the effect that you wish to accomplish. For example, $\exists \mathrm{X}, \mathrm{Y}.\mathrm{move}(\mathrm{X}, \mathrm{Y})$ would yield the answer $\mathrm{move}(217, 400)$ from the theorem prover, if it was able to prove the query). In the lowest layer in a subsumption architecture, the query/task is directly related to actions to be performed by the robot. In higher layers, the queries may be made of different predicates/terms that would influence the behavior of lower layers (e.g., proving the existence of extra objects).

Computationally, we also need to limit the time given to each reasoning task, or otherwise make the reasoning task "dynamic". For example, if the robot needs a response from its theorem provers as to what to do, lower layers (or the robot) would need to interpret a lack of answer as saying something (e.g., "stay put" or "continue what you are doing" or "disregard my last message to you" or "my last message to you may still be used with some discount factor"). I also need to be able to assert fresh sensory information to the reasoner, and some reasoners need to "restart" after the introduction of new information.

**Grosskreutz:** Due to the fact that a robot has to react in a reasonable amount of time, on-line reasoning must be fast, which interdicts the possibility to search from scratch for a possible behavior. At the same time, the task encountered by embodied agents like office robots are very often repeated. This makes it possible and reasonable to renounce categoric approaches like planning and instead to use user-provided procedural knowledge. For example, when letters are to be delivered, it does not seem reasonable to search for a possible plan for each request, but instead seems more appropriate to simply program the robot to deliver one letter after each other.

On the other hand, within such programs it may be useful to explicitly appeal to reasoning tasks like the projection of a plan. In particular, a high-level robot control program might deliberate, i.e. reason about a relatively small set of possible high-level programs, in order to decide which one to execute next. For example, given a robot which is to deliver letters, and additionally has to be in a certain room R at a certain time T, a high-level program might condition the execution of the next delivery on the delivery's predicted execution time, meaning that the robot would only deliver another letter if it has enough time to be back in room R in time.

**Shapiro:** The ability to accept input from humans or other agents means that time must be represented in some objective fashion (see below). This affects the representation of time for on-line reasoning. For example, there is no particular time that can be used in the representation of an acting rule such as "When the light is green, cross the street," because if some particular time, say $t$, were used, someone might tell the robot at a time later than $t$ that the light was green at $t$, and that should not be used to cause the robot to try to cross the street at $t$, which by then is in the past. Therefore, the acting rule must not mention any time, but be triggered when the robot concludes that the light is green at any current "now."

In addition, determining if any state holds "now" in order to decide whether or not to take some action, may, itself, take some time. This requires representing a nest of now's of different granularities, and using the appropriate one for each such decision.

In performing a sequence of actions, a distinction must be made between goalless actions, such as wandering around, and actions with goals, like going to Door1. In linguistics, this distinction is termed "telicity"—actions with goals are called "telic" and those with out goals are called "atelic." Atelic actions may be interrupted at any time, and still be considered successful. Telic actions, however, are successful only if continued until their goal is reached. If a sequence consists of a telic action, $a1$ followed by another action, $a2$, the robot should not start $a2$ until the goal of $a1$ has been reached, and this typically requires preceiving $a1$'s goal obtaining in the world. Meanwhile, the robot might be able to do other things. For example, after being put on hold, you may do some other activities; you continue with your phone conversation once the other party returns.

If an action fails or is interrupted, the robot must decide what it should now do to achieve its goal given its current state. It cannot return to the state the world would have been in if it had not performed the initial part of the failed or interrupted action.

**Soutchanski:**

1. Once an action is executed on-line, the robot cannot undo it; consequently, subsequent resoning has to take into account the fact that the action has been executed.
2. When the robot executed an action, it is important only what outcome that action had in reality: other possible outcomes of the action do not matter for the subsequent reasoning tasks.

### 3.7 What is the effect of the ability to accept input from humans or other agents?

**Amir:** In our system, humans can add/send axioms to the different layers. This way, adding information or advice can be done online, while the robot is acting/deliberating. The reasoning takes care of the new information and incorporates it fairly easily. For example, if you wish to say to your robot that it is located between two landmarks, you can simply state that, provided that the robot has the theory that can "understand" this sentenece (otherwise, you need to send that extra theory too).

An example of some problems that may occur with additional information that is sent from the human is the case where we want to tell the robot that a good way to reach the kitchen is via the hallway that is close to the elevator. Naively, we can simply say that there is a plan that leads the robot to the kitchen and that involves the robot being in the hallway close to the elevator. We would assume that the reasoner would find a plan faster than it would otherwise, because we have created an "island" in its search space. However, our advice creates an alternative way of getting to a situation in which the robot is in the kitchen, but we are no closer to finding a plan for getting to the kitchen.

You can also modify the behavior of the robot by asserting new information. For example, if you tell the robot ("send a new assertion") that there is always an object very close behind it, then the robot would be more "brave" and would move faster in whichever direction it goes. However, asserting flawed information or information that would lead the robot to get into tight spots (e.g, a local minima in its potential field) can create flawed behaviors corresponding to this new information. Also, since (at least in my system) you cannot specify/change the reasoning strategies directly by asserting information while the system is running, the new information may throw the reasoning procedure "off track", causing it to spend a lot of time in fruitless search spaces.

**Shapiro:** As indicated above, such input might assert that some state held in the past. Even if this is a state that should trigger some robot action if detected, it should not trigger an action in this circumstance, since the robot cannot act in the past. On the other hand, if another agent tells the robot that such a state holds "now," it should act accordingly, unless it is designed to be somewhat skeptical and has to "see for itself" that the state holds. In summary, reasoning about time becomes more complicated if the robot can learn about states holding in the past as well as in the present.

In addition, the representation of actions and states should be the same, independent of who is acting, and how the agent came to believe that the action was taken or that the state held. This requires a unification of the representations motivated by acting agents and by NL-understanding agents.

### 3.8 What is the effect of the requirement for natural language reporting of beliefs and actions?

**Shapiro:** As for the previous question, the representations of actions and states for acting, for NL-understanding, and for NL-generation should be the same. Moreover the representation of time used for representing the belief that the agent is acting, when it does, and for representing the states that it perceives, should be appropriate for generating reports of those actions and states when the times move into the past.

# 4 For More Information

**Amir:**

**Research group home page:**
`http://www-formal.stanford.edu`

**Project web page:**
`http://www-formal.stanford.edu/eyal/lsa`

**Papers:**

- Eyal Amir and Pedrito Maynard-Reid II, Logic-Based Subsumption Architecture,*Proceedings of Sixteenth Internation Joint Conference on Artificial Intelligence (IJCAI'99),* 1999.
- Eyal Amir and Pedrito Maynard-Reid II, LiSA: A Robot Driven By Logical Subsumption, *Papers of Commonsense'01,* 2001.

**Grosskreutz:**

**Research group home page:**
`http://www-i5.informatik.rwth-aachen.de/kbsg/`

**Papers:**

- W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, Experiences with an Interactive Museum Tour-Guide Robot, *Artificial Intelligence 114,* 1–2 (October 1999), 3–55.
- H. Grosskreutz and G. Lakemeyer, cc-Golog: Towards More Realistic Logic-Based Robot Controllers, *Proc. AAAI'2000*, 2000.

**Shapiro:**

**Research group home page:**
`http://www.cse.buffalo.edu/sneps/`

**Papers:**

- H. O. Ismail and S. C. Shapiro, Two Problems with Reasoning and Acting in Time. In A. G. Cohn, F. Giunchiglia, & B. Selman, Eds., *Principles of Knowledge Representation and Reasoning: Proceedings of the Seventh International Conference (KR 2000),* Morgan Kaufmann, San Francisco, 2000, 355–365.
- H. O. Ismail and S. C. Shapiro, Conscious Error Recovery and Interrupt Handling. In H. R. Arabnia, Ed., *Proc. International Conference on Artificial Intelligence (IC-AI'2000),* CSREA Press, Las Vegas, NV, 2000, 633–639.
- S. C. Shapiro, "Embodied Cassie," *Cognitive Robotics: Papers from the 1998 AAAI Fall Symposium,* Technical Report FS-98-02, AAAI Press, Menlo Park, CA, 1998, 136–143.
- S. C. Shapiro, Symbol-Anchoring in Cassie, *Papers of the AAAI Fall Symposium on Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems,* North Falmouth, MA, November 2–4, 2001 (accepted to appear).
- S. C. Shapiro, H. O. Ismail, and J. F. Santore, "Our Dinner with Cassie," *Working Notes for the AAAI 2000 Spring Symposium on Natural Dialogues with Practical Robotic Devices,* AAAI, Menlo Park, CA, 2000, 57–61.

**Randell:**

**Research group home page:**
`http://www.iis.ee.ic.ac.uk/~m.witkowski/cogrob2/`

**Papers:**

- John Demiris and Gillian Hayes, Imitation as a Dual-Route Process Featuring Predictive and Learning Components: A Biologically-Plausible Computational Model. In K. Dautenhahn and C. Nehaniv, Eds.,*Imitation in Animals and Artifacts,* MIT Press, 2001 (to appear).
- David Randell, Mark Witkowski and Murray Shanahan, From Images to Bodies: Modelling and Exploiting Spatial Occlusion and Motion Parallax, *Proc. 17th IJCAI*, 2001 (accepted to appear). This paper is also presented at the Commonsense 2001 Symposium.
- Paulo Santos and Murray Shanahan, From stereoscopic vision to symbolic representation, *Papers of the AAAI Fall Symposium on Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems,* North Falmouth, MA, November 2–4, 2001 (accepted to appear).
- Murray Shanahan, A Logical Account of the Common Sense Informatic Situation for a Mobile Robot, *Electronic Transactions on Artificial Intelligence,* 1999.
- Murray Shanahan, An Abductive Event Calculus Planner, *The Journal of Logic Programming, 44* (2000), 207–239.
- Murray Shanahan and Mark Witkowski, High-Level Robot Control Through Logic, *Proceedings ATAL 2000, Lecture Notes in Artificial Intelligence,* Springer-Verlag, 2000 (to appear).
- Mark Witkowski, David Randell and Murray Shanahan, Deriving Fluents from Sensor Data for Mobile Robots, *Papers of the AAAI Fall Symposium on Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems,* North Falmouth, MA, November 2–4, 2001 (accepted to appear).

**Soutchanski:**

**Research group home page:**
`http://www.cs.toronto.edu/cogrobo/`

**Papers:**

- Boutilier, C., Reiter, R., Soutchanski, M., and Thrun, S., Decision-Theoretic, High-level Robot Programming in the Situation Calculus, *Proc. of the 17th National Conference on Artificial Intelligence (AAAI'00)*, Austin, Texas, 2000.
- W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, Experiences with an Interactive Museum Tour-Guide Robot, *Artificial Intelligence, 114*, 1–2 (1999).
- Levesque, H.J., Pirri, F., and Reiter, R., Foundations for the Situation Calculus, *Linköping Electronic Articles in Computer and Information Science, 3,* 18, (1998).
- Levesque, H.J., Reiter, R., Lespérance, Y., Lin, F., Scherl, R., Golog : A logic programming language for dynamic domains, *J. of Logic Programming, Special Issue on Actions 31,* 1–3, (1997), 59–83.
- Reiter, R., *Knowledge in Action: Logical Foundations for Describing and Implementing Dynamical Systems*, MIT Press, 2001.
- Soutchanski, M, An On-line Decision-Theoretic Golog Interpreter, *17th International Joint Conference on Artificial Intelligence (IJCAI)*, Seattle, Washington, 2001.