1992-:

AN AUTONOMOUS AGENT ARCHITECTURE FOR INTEGRATING PERCEPTION AND ACTING WITH GROUNDED, EMBODIED SYMBOLIC REASONING

Henry Hexmoor, Johan Lammens
Stuart Shapiro

92-21 **September 1992**

Department of Computer Science
State University of New York at Buffalo
226 Bell Hall
Buffalo, New York 14260

An Autonomous Agent Architecture for Integrating Perception and Acting with Grounded, Embodied Symbolic Reasoning*

Henry Hexmoor
Johan Lammens
Stuart Shapiro

Computer Science Department, State University of New York at Buffalo, Buffalo, NY 14260

September 9, 1992

Abstract

We describe an agent architecture in terms of general principles of organization of components for an autonomous agent that functions in the world. This architecture is described independent of computational components of the agent that are used to generate behavior. It specifies an integration of explicit representation and reasoning mechanisms, embodied semantics through grounding symbols in perception and action, mechanisms for finding and maintaining a correspondence between symbols and sensory-perceived objects, and implicit representations of special-purpose mechanisms of sensory processing, perception, and motor control for the agent. We then present components that we place in our general architecture to build an agent that exhibits situated activity and learning.

1 Overview

In this article we present both a general multi-level architecture for an autonomous cognitive agent with integrated sensory and motor capabilities, GLAIR[1], and an instantiation of that architecture for a particular situated cognitive agent, GLAIR-agent.

By an *architecture* we mean an organization of components of a system, what is integral to the system, and how the various components interact.[2] Which components go into an architecture for an autonomous agent has traditionally depended to a large extent on whether we are *building a physical system*, *understanding/modeling behaviors of an anthropomorphic agent*, or *integrating a select number of intelligent behaviors.* The organization of an architecture is also influenced by adopting the *modularity* assumption of Fodor [Fodor, 1983], or a *connectionist* point of view, e.g. [McClelland et al., 1986], or an *anti-modularity* assumption as in Brooks's subsumption architecture [Brooks, 1985]. The *modularity* assumption supports (among other things) a division of the mind into a *central system*, i.e., cognitive processes such as learning, planning, and reasoning, and a *peripheral system*, i.e., sensory and motor processing [Chapman, 1990]. Our architecture is characterized by a three-level organization into a Knowledge Level, a Perceptuo-Motor Level, and a Sensory-Actuator Level. This organization is neither modular, anti-modular, hierarchical,

*The first author thanks the AI Programs at the University of Georgia and its director professor Donald Nute for providing resources for his use during his 1992 summer scholar visit at the AI Programs

[1] Grounded Layered Architecture with Integrated Reasoning

[2] Our discussion of architecture in this paper extends beyond any particular physical or software implementation.

anti-hierarchical, nor connectionist in the conventional sense. It integrates a traditional symbol system with a physically grounded system, i.e., a *behavior-based* architecture. The most important difference with a behavior-based architecture like Brooks's subsumption architecture is the presence of three distinct levels with different representations and implementation mechanisms for each, particularly the presence of an explicit Knowledge Level. Representation, reasoning (including planning), perception, and generation of behavior are distributed through all three levels. Our architecture is best described using a resolution pyramid metaphor as used in computer vision work [Ballard & Brown, 1982], rather than a central vs. peripheral metaphor.

Architectures for building physical systems, e.g., robotic architectures [Albus et al., 1981], tend to address the relationship between a physical entity, (e.g., a robot), sensors, effectors, and tasks to be accomplished. Since these physical systems are performance centered, they often lack general knowledge representation and reasoning techniques. These architectures tend to be primarily concerned with the *body*, that is, how to get the physical system to exhibit intelligent behavior through its physical activity. We say these systems are not concerned with *consciousness*. These architectures address what John Pollock calls *Quick and Inflexible* (Q&I) processes [Pollock, 1989]. We define consciousness for a robotic agent operationally as being aware of one's environment, as evidenced by (1) having some internal representations that are causally connected to the environment through perception, (2) being able to reason explicitly about the environment, and (3) being able to communicate with an external agent about the environment. [3]

Architectures for understanding/modeling behaviors of an anthropomorphic agent, e.g., cognitive architectures [Anderson, 1983, Pollock, 1989, Langley et al., 1991], tend to address the relationships that exist among the structure of memory, reasoning abilities, intelligent behavior, and mental states and experiences. These architectures often do not take the *body* into account. Instead they primarily focus on the *mind* and *consciousness*. Our architecture ranges from general knowledge representation and reasoning to body-dependent physical behavior, and the other way around.

We are interested in autonomous agents that are embedded[4] in a dynamic environment. Such an agent needs to continually interact with and react to its environment and exhibit intelligent behavior through its physical activity. To be successful, the agent needs to reason about events and actions in the abstract as well as in concrete terms. This means combining situated activity with acts based on reasoning about goal-accomplishment, i.e., deliberative acting or planning. In the latter part of this paper, we will present an agent based on an application of our general architecture. This agent is designed with a robot in mind, but its structure is also akin to an anthropomorphic agent. Figure 1 schematically presents our architecture. There are several features that contribute to the robustness of our architecture. We highlight them below (an in-depth discussion follows later).

- We differentiate conscious reasoning from unconscious Perceptuo-motor and sensori-actuator processing.[5]

[3] A machine like a vending machine or an industrial robot has responses, but it is *unconscious*. See [Culbertson, 1963] for a discussion of independence of consciousness from having a response. Also, intelligent behavior is independent of consciousness in our opinion.

[4] "Embedded agents are computer systems that sense and act on their environment, monitoring complex dynamic conditions and affecting the environment in goal-oriented ways." ([Kaelbling & Rosenschein, 1990] page 1).

[5] We consider body-related processes to be *unconscious*, but that is not meant to imply anything about their complexity or importance to the architecture as a whole. Indeed, we believe that the unconscious levels of our architecture (the Perceptuo-Motor Level and the Sensori-Actuator Level) are at least as important to the architecture as the conscious one (the Knowledge Level). We reserve the term *sub-conscious* for implicit cognitive processes such as category subsumption in KRR systems. See [Shapiro, 1990] for a discussion of sub-conscious reasoning.

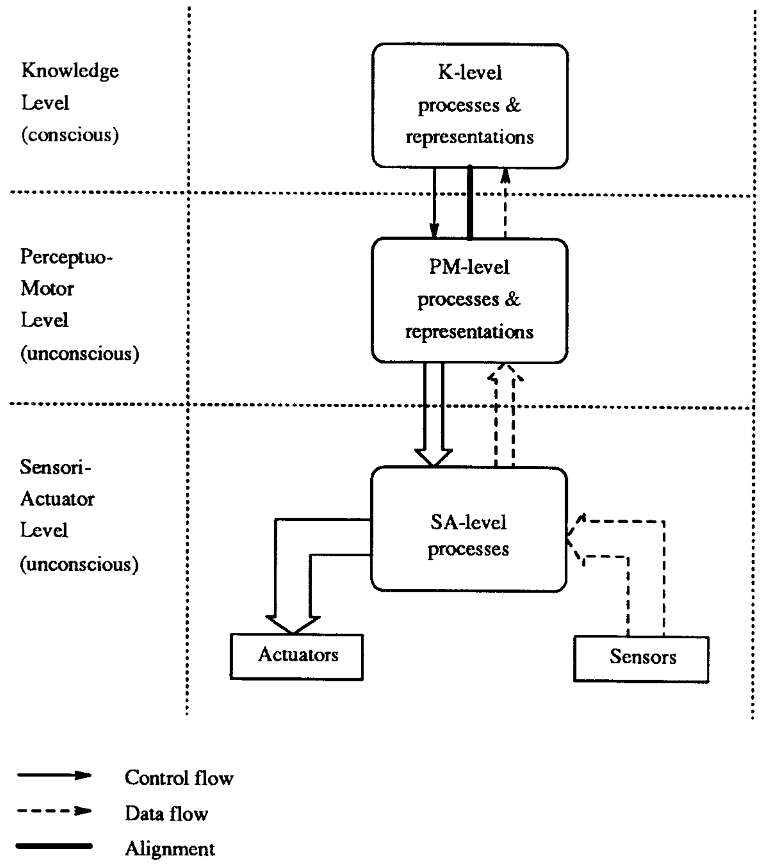

Figure 1: Schematic representation of the agent architecture. Width of control and data paths suggests the amount of information passing through (bandwidth). Sensors include both world-sensors and proprio-sensors.

- The levels of our architecture are semi-autonomous and processed in parallel.[6]
- Conscious reasoning takes place through explicit knowledge representation and reasoning. Unconscious behavior makes use of several different mechanisms.
- Conscious reasoning guides the unconscious behavior, and the unconscious levels, which are constantly engaged in perceptual and motor processing, can *alarm* the conscious level of important events, taking control if necessary. Control and generation of behavior are layered and not exclusively top-down.
- There is a correspondence between terms in the Knowledge Representation and Reasoning (KRR) system on one hand, and sensory perceived objects, properties, events, and states of affairs in the world and motor capabilities on the other hand. We call this correspondence *alignment*.

[6]This autonomy is similar to Brooks's subsumption architecture [Brooks, 1985], but at a more macroscopic level. Brooks does not distinguish between the three levels we describe, as his work is solely concerned with behaviors whose controlling mechanism we would situate at the Perceptuo-Motor Level.

2 The Architecture

In this section we discuss in detail our autonomous agent architecture for integrating perception and acting with grounded, embodied, symbolic reasoning.

2.1 Related Work

Architectures proposed in the literature do not fall into neatly separable classes, mainly because the scope of the models and the motivations vary widely. However, we can divide a review of related work into theoretical issues of agent architectures, on the one hand, and implemented architectures, on the other.

2.1.1 Theoretical Issues

A situated agent, at any moment, attends to only a handful of entities and relationships in its immediate surroundings. In this type of setting, the agent often does not care to uniquely identify objects. It is sufficient to know the current relationship of the relevant objects to the agent, and what roles the objects play in the agent's activities. Agre and Chapman in [Agre & Chapman, 1987] proposed indexical-functional representations (which [Agre, 1988] refers to as deictic representations) to be the more natural way agents refer to objects in common everyday environments. They called entities and relationships of interest *entities* and *aspects*, respectively. With respect to its current activities, the agent needs only to focus on representing those entities and relationships. Although the objects in the environment come and go, the representations of entities and relationships remains the same. For example, the-cup-that-I-am-holding[7] is an indexical-functional notation that abstracts the essentials of what the agent needs to know in its interaction. These representations serve to limit the scope of focus on entities. For example, if the agent wants to pick up a cup, it does not need to know *who owns the cup* or *how much coffee the cup can hold*; only the relevant attributes of the cup apply. The main use of this representation is its ability to narrow the focus of attention about the cup.

To our knowledge, research prior to ours created indexical-functional representations by hand.[8] We suggest that systems endowed with general KRR abilities can and should generate deictic representations to create and maintain a focus on entities in the world. Since we are concerned with an integrated agent architecture, we will sketch a solution to this problem (section 2.3).

The issue of focusing on the relevant information about objects differs from two other concepts that Agre used together with indexical-functional representation for focusing attention. These are *indexing* and *marking*, and we still need these concepts. Indexing refers to choosing an object in the environment of the agent that will be of interest because of its properties, e.g., the block that can be used *to hit the bee with.* Marking refers to remembering an entity for the purpose of tracking its properties over time, e.g., tracking *the position of a moving penguin.*

We believe that behavior-based AI has adopted the right treatment of every day behavior for agents that function in the world. However, this has been done at the expense of ignoring cognitive processing such as planning and reasoning. Clearly, what is needed is an approach that allows for both. We believe that our architecture meets this need. As in behavior-based AI, GLAIR gains validity from its being grounded in its interaction with the environment, while it benefits from a

[7]This kind of designation is merely a mnemonic representation intended to suggest the entity and aspect under consideration, for the purpose of our exposition. It is not the actual representation that would be used by an agent.

[8]This is perhaps because in the application environments, only a limited number of entities are ever of interest for the agent. For example, see [Agre & Chapman, 1987] and [Whitehead, 1992].

knowledge level that, independent of reacting to a changing environment, performs reasoning and planning.

The Model Human Processor (MHP) is a cognitive model [Card et al., 1983] that suggests the three components of perception, cognition, and motor. Cognition consists of working memory, long-term memory, and the cognitive processor. Perception is a hierarchy of sensory processing. Motor executes the actions in the working memory. This is a traditional symbol-system decomposition of human information processing. This type of decomposition has shown only limited success in building physical systems. Despite this, systems like SOAR adhere to this cognitive model. In our architecture, we purposively avoid this kind of top-down problem decomposition by allowing independent control mechanisms at different levels to take control of the agent's behavior, and pre-empt higher level control while doing so.

2.1.2 Implemented Architectures

Albus et al's hierarchical control architecture [Albus et al., 1981] is an example of a robotic architecture; we would say it is *body centered.* This architecture proposed abstraction levels for behavior generation, sensory processing, and world modeling. At higher levels, tasks were decomposed and passed to lower levels. By descending down the hierarchy, tasks were decomposed into robot-motion primitives. This differs from our architecture, which is not strictly top-down controlled. Concurrently, at each level of the hierarchy, feedback processing modules extracted the information needed for control decisions at that level from the sensory data stream and from the lower level control modules. Extracted environmental information was compared with the expected internal states to find differences. The differences were used for planning at higher levels.

Payton in [Payton, 1986] introduced an architecture for controlling an autonomous land vehicle. This architecture has four levels: mission planning, map-based planning, local planning, and reflexive planning. All levels operated in parallel. Higher levels were charged with tasks requiring high assimilation and low immediacy. The lower levels operated on tasks requiring high immediacy and low assimilation. This feature is similar to our architecture. The reflexive planning was designed to consist of pairs of the form $\langle virtualsensor, reflexivebehavior\rangle$. Each reflexive behavior had a priority and a central blackboard style manager arbitrated among the reflex behaviors. Some of the problems with the earlier implementation due to using the blackboard model were solved in [Rosenblatt & Payton, 1989].

Rosenschein and Kaelbling's work [Kaelbling, 1988, Kaelbling & Rosenschein, 1990] describes tools (REX, GAPP, RULER) that, given task descriptions of the world, construct reactive control mechanisms termed *situated automata.* Their architecture consists of perception and action components. The robot's sensory input and its feedback are inputs to the perception component. The action component computes actions that suit the perceptual situation. We should note that unlike Brooks's behavior modules, situated automata use internal states. They are mainly intended to produce circuits that operate in realtime, and they can prove properties about their operation. So their decisions are not Markovian (i.e., they are not ahistoric). The mechanism for generating situated automata, although impressive, seems unrealistic in an anthropomorphic sense. Perhaps the operation of our Perceptuo-motor level can be modeled by a situated automaton, but we are not convinced that this is the right formalism to use.

Gat in [Gat, 1991] describes ATLANTIS, an architecture for the control of mobile robots. This architecture has three components: control, sequencing, and deliberation. The control layer is designed as a set of circuit-like functions using Gat's language for circuits, ALPHA. The sequencing is a variation of Jim Firby's RAP system [Firby, 1987]. The deliberation layer is the least described layer. As for situated automata, we are not convinced that this is the right kind of formalism to use.

An architecture for low-level and high-level reactivity is suggested in [Hexmoor, 1989]. High-level reactivity is reactivity at the conceptual level. This architecture suggests that an autonomous agent maintains several different types of goals. High-level reactivity is charged with noticing impacts of events and actions in the environment on the agent's goals. Subsequently, high-level reactivity needs to guide the agent's low-level reactivity. Low-level reactivity is at the sensory, perceptual, and motor level. The mechanism for low-level reactivity is similar to other reactive systems that have components for perception and action arbitration. The novelty of this architecture is the incorporation of high-level reactivity and a supervisory level of planning and reasoning, which guides the choice of low-level reactive behaviors. In our present conception of agent architecture, we avoid a sharp separation between the two types of reactivity. We also relax the top-down nature of interaction between levels. Reactivity may be initiated at any level of our architecture either due to interaction with other levels or in direct response to external stimuli.

Brooks's *subsumption* architecture, [Brooks, 1985, Brooks, 1987, Brooks, 1990], clusters behaviors into layers. Low-level behaviors, like deciding the direction of motion and speed, can be interrupted by behaviors determined at higher levels, such as avoiding obstacles. Subsumption behaviors are written as finite state machines augmented with timing elements. A compiler is used to simulate the operation of finite state machines and parallelism. This architecture is implemented on a variety of mobile robots. Frequently used behaviors are placed at a lower level than less frequently-used behaviors. This organization of behaviors gives the system fast response time and high reactivity. Our architecture is similar to Brooks's in our intra-level implementations of behaviors. However, the subsumption architecture lacks the separation we have made into conscious and non-conscious spheres. In anthropomorphic terms, Brooks's agents are all non-conscious. Pattie Maes has experimented with a version of a behavior-based architecture, which she calls ANA [Maes, 1991]. This architecture consists of competence modules for action and a belief set in a network relating modules through links denoting successors, predecessors, and conflicts. Competence modules have activation levels. Activations are propagated and the competence module with the highest activation level is given control. Maes has explored learning and has applied her architecture to robotic systems.

The Servo, Subsumption, Symbolic (SSS) architecture [Connell, 1992] is a hybrid architecture for a mobile robot that integrates three independent layers of servo control, Brooksian behavior based modules of a subsumption architecture, and a symbolic layer. This architecture is similar to ours in its general spirit of identification and integration of three distinct levels corresponding to levels of affinity-of-interaction (i.e., the rate at which it is in real-time contact with the world) with the outside world. This similarity also constitutes a point of departure, however, in that SSS is defined with respect to specific (and different) implementation techniques. For example, the symbolic layer in SSS seems to be a decision table versus a general KRR as intended in GLAIR. Unlike GLAIR, SSS assigns particular tasks for each layer and uses a hard-wired interconnection channel among layers.

SOAR [Laird et al., 1987] was designed to be a general problem solving architecture. SOAR integrates a type of learning known as *chunking* in its production system. Recently, SOAR has been applied to robotic tasks [Laird et al., 1991]. In this framework, planning and acting is uniformly represented and controlled in SOAR. This approach lacks the ability of our architecture for generating behavior at non-conscious levels as well as the conscious level (or at different levels in general), and for having different-level behaviors interact in an asynchronous fashion. It also lacks our multi-level representations.

Simmons's Task Control Architecture (TCA) [Simmons, 1990] interleaves planning and acting by adding *delay-planning constraints* to postpone refinement of planning until execution. For example, a plan for a robot to collect used cups for trash is decomposed into: navigate to the cup; pick it up; navigate to trash bin; deposit the cup. Since the robot does not have sensory information about the cup yet, the plan to pick it up is delayed until the robot gets close enough. Selectively delaying refinement of plans allows for reactivity. This type of "stepwise refinement" follows effortlessly from our architecture, without the need to explicitly implement it. Since conscious planning which goes on at the Knowledge Level uses a more coarse-grained world model, there is simply no possibility to express fine details of planning and execution. These can only be represented and/or computed at the lower Perceptuo-Motor Level and Sensori-Actuator Level. Planning and execution in our architecture may proceed in a lock-step fashion, but they need not (see the discussion of engaged vs. disengaged reasoning in section 2.5). TCA used a message-passing scheme among modules that allows concurrent execution of tasks. TCA has been used to control the six-legged walking robot Ambler and a cup-collecting robot.

2.2 Architecture Levels

2.2.1 Motivation

The three levels of our architecture are of organizational as well as theoretical importance. Organizationally, the layered architecture allows us to work on individual levels in a relatively independent manner, although all levels are constrained by the nature of their interactions with the adjoining level(s). The architecture is hierarchical, in that level i can only communicate with levels $i-1$ and $i+1$, if any. This hierarchical architecture simplifies modularity of implementations.[9]

The levels of our architecture are semi-independent. While control flows mainly top-down and data mainly bottom-up, local control mechanisms at any level can preempt higher-level control, and these local mechanisms filter the data stream for their own purpose, in parallel with higher-level ones. Representations become coarser-grained from bottom to top, while control data becomes more fine-grained from top to bottom. The terms in the Knowledge Level's KRR system model conscious awareness of the world (and the body), and the perception and motor capabilities in the other levels provide the grounding for an *embodied semantics* of the former. Routine, reflex-like activities are controlled by close coupling of perception with motor actions at the (unconscious) Perceptuo-motor Level. This close coupling avoids having to exert control over these activities from the conscious level, as in purely top-down structured architectures with a symbol level at the top of the hierarchy. In the latter kind of system, signals must first be transformed to symbols and vice versa.[10] The low-level coupling provides for better real-time performance capabilities, and relieves the Knowledge Level of unnecessary work.

In general, we have multi-level layered representations of objects, properties, events, states of affairs, and motor capabilities, and the various levels are *aligned.* By alignment we mean a correspondence between representations of an entity at different levels. This organization contributes to the robustness and computational efficiency of implementations. The semi-autonomous nature of the levels allows for graceful degradation of system performance in case of component failure or situation-dependent incapacitatedness. Lower levels can function to some extent without higher-level control, and higher levels can function to some extent without lower-level input.[11]

[9] Not to be confused with a modular architecture as such, e.g., [Fodor, 1983]

[10] Our lower-level coupling is similar to Brooks-style behaviors where he constructs competence modules.

[11] For instance, in the context of autonomous vehicles, if obstacle avoidance or returning to the base is a lower-level behavior than planning exploration strategies, then a failure of the hardware implementing the latter does not

Our architecture allows us to elegantly model a wide range of behaviors: from mindless, spontaneous, reflex-like, and automatic behavior, e.g., "stop if you hit an obstacle", to plan-following, rational, incremental, and monitored behavior, e.g., "Get in the car now, if you want to go to LA on Friday".[12]

In anthropomorphic terms, we identify the Knowledge Level with consciously accessible data and processing; the Perceptuo-Motor Level with "hardwired", not consciously accessible processing and data involved with motor control and perceptual processing; and the Sensori-Actuator Level with the lowest-level muscular and sensor control, also not consciously accessible. The substrate of grounding and embodiment [Harnad, 1990, Lakoff, 1987, Suchman, 1988] of actions, concepts, and reasoning is mainly the Perceptuo-Motor Level and to some extent the Sensori-Actuator Level. Grounding is accomplished through the alignment of the Knowledge and Perceptuo-Motor Levels.

We will now explore representation and computation at the individual levels in more detail.

2.2.2 The Knowledge Level

The *Knowledge Level* is the level of a traditional KRR or planning system, using a relatively course-grained representation of objects, events (including actions), and states of affairs. For instance, objects are represented at this level as unique identifiers, typically without further detail about their physical characteristics or precise locations. It is possible to represent such detail explicitly at this level, but not required. Only if the detail becomes important to the reasoning at this level will it be represented, though not necessarily in the same way as it is represented at a lower level. For example, knowledge about the physical size and weight of an object might become available at the Knowledge Level through the agent's actively using measuring devices like a ruler or a scale, but this knowledge is not the same as the embodied knowledge about dimensions and weight represented at the Perceptuo-Motor Level for the particular object or its object class. As a rule of thumb, representations at this level are limited to objects, events, and states of affairs that the agent needs to be consciously aware of in order to reason and plan, and in order to communicate with other agents at the grain size of natural language. The Knowledge Level can be implemented using different KRR and/or planning systems.

Traditional use of the concept of world modeling refers to building models of interactions between the agent and its environment at the conscious level. These models maintain internal states for the agent. The difference in our use of the term "world model" is that we do not intend to have a precise model of all objects in the environment. Instead, we want to model only the entities relevant to the agent's interaction with its world. This requires filtering out some details accessible at the Perceptuo-motor Level as the entities are aligned with their counterparts on the Knowledge Level. This is known as "perceptual reduction". Physical details of interaction with entities are handled at the Perceptuo-motor Level. Representations at the Knowledge Level are needed only for explicit reasoning about entities, and contain only the information necessary for doing so. That might include details about physical characteristics in some cases, but it need not. In other cases, it may be limited to a nondescript intensional representation of an object.[13] Conversely, some entities may be represented at the Knowledge Level but not at the Perceptuo-Motor Level (abstract concepts, for instance). Knowledge Level representations are needed for reasoning about entities; Perceptuo-Motor Level representations are needed for physically interacting with entities.

necessarily prevent the former.

[12]The plan is to get in the car to go to the travel agency to get a ticket to fly to LA on Friday. Today is Thursday and it is near the end of the business day. Also, the agency won't accept telephone reservations. This example is suggested in [Pollock, 1992].

[13]See [Shapiro & Rapaport, 1987] for our use of "intensional representation".

Our architecture at the knowledge level comprises a general KRR system. All knowledge about cognitive behaviors for reasoning, planning, learning, as well as knowledge about the agent itself, is uniformly represented in the KRR system. Knowledge about the agent includes knowledge about its capabilities, i.e., primitive and complex actions. We will say more about agent capabilities in our discussion of our application of our architecture in building an agent (section 3; for a discussion of agent capabilities see [Shapiro et al., 1988]). Cognitive behaviors may produce behaviors that need to be realized by the agent in the world, but realization is in the realm of the lower levels in our architecture.

As is pointed out in [Shapiro et al., 1988], from the perspective of a general KRR system, external entities are extensions of concepts in the agent's mind. However, as we will discuss in section 2.3, we need not maintain an extension for each entity in the agent's mind. Instead, we need to identify objects in terms of how they relate to the agent and the agent's current task.

2.2.3 The Perceptuo-Motor Level

The *Perceptuo-Motor Level* uses a more fine-grained representation of events, objects, and states of affairs. For instance, they specify such things as size, weight, and location of objects on the kinematic side, and shape, texture, color, distance, pitch, loudness, smell, taste, weight, and tactile features on the perceptual side. At this level, enough detail must be provided to enable the precise control of actuators, and sensors or motor memory must be able to provide some or all of this detail for particular objects and situations. The Perceptuo-Motor Level is partly *aligned* with the Knowledge Level, in that there is a correspondence between object identifiers at the Knowledge Level and objects at the Perceptuo-motor Level.

Kinematic and perceptual representations of particular objects or typical object class instances may be unified or separate, and both kinds of representations may be incomplete. Also at this level are elementary categorial representations, which may unify kinematic and perceptual representations. By "elementary categorial representations", we mean the kinds of representations that function as the grounding for elementary grounded symbols at the Knowledge Level, i.e., sensory-invariant representations constructed from sensory data by the perceptual processor [Harnad, 1990].

The representations at the Perceptuo-Motor Level are *embodied* (cf. [Lakoff, 1987]). That means that the representations depend on the body of the agent, its particular dimensions and characteristics. Robots will therefore have different representations at this level than people would, and different robots will have different representations as well. These representations are agent-centered and agent-specific. For instance, they would not be in terms of grams and meters, but in terms of how much torque to apply to an object to lift it,[14] or what percentage of the maximum to open the hand to grasp an object. Weights of things in this kind of representation are relative to the agent's lifting capacity, which is effectively the maximum weight representable. An agent may have a conscious (Knowledge Level) understanding and representation of weights far exceeding its own lifting capacity, but that is irrelevant to the Perceptuo-Motor Level. When it comes to lifting it, a thousand-pound object is as heavy as a ten-thousand-pound one, if the capacity is only a hundred or so. Similarly, sizes are relative to the agent's own size. Manipulating small things is

[14] Of course this also depends on how far the object is removed from the body, or how far the arm is stretched out, but that can be taken into account (also in body-specific terms, of course). People's Perceptuo-Motor Level idea of how heavy something is is most likely not in terms of grams, either (in fact, a conscious estimate in grams can be far off), but in terms of how much effort to apply to something to lift it. That estimate can be off, too, which results in either throwing the object in the air or not being able to lift it at the first attempt, something we have all experienced. On the other hand, having a wrong conscious estimate of the weight of an object in grams does not necessarily influence one's manipulation of the object.

not the same as manipulating large things, even if they are just scaled versions of each other. A consequence of using embodied representations is that using different "body parts" (actuators or sensors) requires different representations to be programmed or (preferably) learned. While that may be a drawback at first, once the representations are learned they make for faster processing and reactive potential. Representations are direct; there is no need to convert from an object-centered model to agent-centered specifications. This makes the computations at this level more like table lookup than like traditional kinematics computations, which can be quite involved. Learning new representations for new objects is also much simpler; it is almost as easy as trying to grasp or manipulate an object, and merely recording one's efforts in one's own terms. The same holds, mutatis mutandis, for perceptual representations.

There are a number of behaviors that originate at this level: some are performed in service of other levels (deliberative behaviors), some are performed in service of other behaviors at this level, a few are ongoing, and some others yet are in direct response to external stimuli. An agent may consciously decide to perform Perceptuo-motor actions such as looking, as in *look for all red objects*, or to perform a motor action, such as *grasp a cup*. These actions originate at the Knowledge Level and are propagated to this level for realization. An agent has to perform special perceptual tasks to serve other behaviors, such as to *find the grasp point of a cup* in order to *grasp a cup*. These perceptual tasks may originate at this or another level.

An agent embedded in an environment always senses its immediate surroundings, e.g., *sees* its surrounding.[15] This is an ongoing process, since the agent does not decide to perform it; rather, it can't help doing it. We also differentiate reflexive action from reflex actions. We suggest that the perceptual sub-systems of an embedded agent continuously (a) create and maintain a representation for objects in the agent's immediate surrounding, i.e., indexing [Agre, 1988], (b) arbitrarily focus on objects of interest, i.e., marking [Agre, 1988], (c) track properties and attributes of objects, e.g., positions of objects.

At the Perceptuo-Motor Level, an agent has a close coupling between its behaviors, i.e., responses, and stimuli, i.e., significant world states. We observe that, for a typical agent, there are a finite (manageably small) number of primitive ("innate") behaviors available. As the agent interacts with its environment, it may learn sophisticated ways of combining its behaviors and add these to its repertoire of primitive behaviors. We will consider only an agent's primitive abilities for now. We further assume that the agent starts out with a finite number of ways of connecting world states to behaviors, i.e., reflex/reactive rules. Following these observations, we suggest that at this level, the agent's behavior-generating mechanism is much like a finite state automaton. As we noted earlier, learning will change this automaton. What we want to point out is that at this level, the agent starts with an automaton with limited acuity. The agent uses its conscious level to deal with world states not recognizable at the Perceptuo-Motor Level. For instance, the Perceptuo-Motor Level of a person beginning to learn how to drive, is not sophisticated enough to respond to driving conditions automatically. As the agent becomes a better driver, the conscious level is freed to attend to other things while driving. This is called *automaticity* in psychology. We discuss an implementation mechanism for these automated behaviors in section 3.5.

2.2.4 The Sensori-Actuator Level

The *Sensori-Actuator Level* is the level of primitive motor and sensory actions, for instance "move from $\langle x, y, z\rangle$ to $\langle x', y', z'\rangle$ or "look at $\langle x, y, z\rangle$". At this level, there are no object representations as there are at the Knowledge Level and the Perceptuo-Motor Level. There are no explicit declarative

[15] As opposed to *looks* at it.

representations of any kind, only procedural representations (on the actuator side) and sensor data (on the sensory side). Primitive motor actions may typically be implemented in a robot control language like VAL, and some elementary data processing routines may be implemented in a sensory sub-system, like dedicated vision hardware. At this level, we also situate *reflexes*, which we consider to be low-level loops from sensors to actuators, controlled by simple thresholding devices, operating independently of higher-level mechanisms. We see reflexes as primitive mechanisms whose main purpose is prevention of damage to the hardware, or to put it in anthropomorphic terms, survival of the organism. As such they take precedence over any other behavior. When reflexes are triggered, the higher levels are made "aware" of this by the propagation of a signal, but they have no control over the reflex's execution, which is brief and simple (like a withdrawal reflex seen in people when they stick their hand into a fire).[16] After the completion of a reflex, the higher levels regain control and must decide on how to continue or discontinue the activity that was interrupted by the reflex. Reflex-like processes may also be used to shift the focus of attention of the Knowledge Level.

2.3 Deictic (Indexical-Functional) Representations

Objects may be present for the first time in the agent's environment. Objects may be in motion. There may be several objects that are perceptually identical. In general, there may be a large number of objects in the environment of the agent. In these circumstances, it is inappropriate to assign unique names for each object, as is presupposed in traditional model theory. In this section, we describe a class of representations that address the *reference problem* in interacting with a dynamic world. This problem is to find and to maintain a correspondence between an agent's representations and objects in its surroundings. In planning, this type of representation is called *indexical-functional* [Agre & Chapman, 1987], or deictic [Agre, 1988] since it relates the referent of a representation to its situation of use (indexicality [Chapman, 1990]), and it relates the referent of a representation to the activity in which the agent is is engaged (functionality).

The advantages of using deictic (indexical-functional) representation over traditional KRR techniques are that:

- In realistic domains, there will be a large number of objects. There is no need to have a representation for each object in the world.
- There is no need for instantiation of object symbols to their extensions in the world.[17]

The referent in the world is (by necessity) only indirectly considered via its Perceptuo-Motor Level representation, hence the problem becomes one of aligning the Knowledge Level representations with the Perceptuo-Motor Level representations. Deictic representation is appropriate for this task. In our treatment of a situated agent (section 3), we give an account of a KRR system using deictic representations. In the remainder of this section, we will (for the purpose of our discussion) consider physical objects as referents of symbols, keeping in mind that the real referents of Knowledge Level symbols are Perceptuo-Motor Level representations.

Using deictic representation, objects are not uniquely identified with symbols. Instead, a deictic symbolic *name* is constructed for each object, as if to answer three questions about it:

[16] An appropriate reflex for a robot (arm) might be to withdraw or stop when it meets too much mechanical resistance to its movement, as evidenced for instance by a sharp rise in motor current draw. Such a reflex could supplant the more primitive fuse protection of motors, and make an appropriate response by the system possible. Needless to say, a robot that can detect and correct problems is much more useful than one that merely blows a fuse and stops working altogether.

[17] This is a complaint against traditional KRR systems. Our concept of alignment between levels is immune to this objection.

1. What type of object is it? E.g., *a coin, a cup, a saucer.*
2. What is the relation/relevance of the object to the agent? E.g., *in the pocket, in front of.*
3. What is the role of the object in the agent's current task? E.g., if the agent is *drinking* from a cup, the cup's role is *drinking-from*; if the agent is *using a fork to lift an object*, the fork is *used-in-lifting-an-object.*

Putting it together, we have descriptions like a-cup-in-front-of-me-i-am-drinking-from. Deictic *names* are useful only in so far as they limit the focus for processing and reasoning. A mnemonic form can be generated for human inspection of the functioning of the agent under development.

Let's consider two cases of how indexical-functionals can be generated and used.[18] In the first case, let's consider an object that is in the immediate sensing range; and in the second case, we seek an object in the field of view. In the first case, we assume that the agent's sensori-perceptual mechanism automatically identifies the object and creates a representation for it. This is what we refer to as an on going behavior in the Perceptuo-Motor Level. Furthermore, we assume that at the Knowledge Level, the first and second questions about the object are trivially answered, e.g., so far generating an-apple-on-the-table. Since (as a simplifying assumption) we *index* all objects in the visual field, this eliminates the need to *index* objects as suggested in [Agre, 1988] in response to actions involving objects already in the visual field. *Indexing* an object refers to selecting an object that has properties of interest, which Agre called *indexable.* Let's also assume that all moving objects are *marked. Marking* refers to placing a marker on an object or a location for future reference. This is used in tracking a moving object. When the agent decides to execute an act, let's say *grasping(apple)*, if the required types of objects of the act match the type of object thus far identified, the third question is answered by a form of functional unification (*fu*). In our example, the argument of *grasping* is of the same type as the object sensed; they are both *apple* types. Using *fu*, the object at the Knowledge Level is identified indexico-functionally as an-apple-on-the-table-I-want-to-grasp. Note that all representations are from the perspective of the agent, so 'I' is automatically integrated into indexical-functional representations. It is at this stage that the agent propagates its decision *to grasp* down to the Perceptuo-Motor Level.

In the second case, the agent has decided to execute *grasping(apple)* but it does not yet know the referent of the argument of the function *grasping()*. The agent must postpone executing its grasping until it situates itself in an environment where it can sense objects of the type matching the object types of the action. This is in effect an active vision task. Assuming the agent has found an object of the desired type, it can proceed as in the first case. The only difference in the second case is that the agent has to first plan and execute finding an object before it can proceed with *fu.*

2.4 Symbol Grounding: A Non-Tarskian Semantics

> Tarskian Semantics has nothing to say about how descriptions of objects in plans relate to the objects in the world [McDermott, 1991, p. 13].

One problem an agent has to solve is how to find and maintain a correspondence between a referent in the world and a symbol in an agent's world model. As noted above, the referent in the world is (by necessity) only indirectly considered via its embodied Perceptuo-Motor Level representation, hence the problem becomes one of aligning the Knowledge Level representations with the Perceptuo-Motor Level representations. We have proposed that Deictic (indexical-functional)

[18] We have not yet tested our suggestions in this section.

representation is appropriate for this task (section 2.3). From the perspective of cognitive science, the problem has been labeled the *symbol grounding problem* [Harnad, 1990]. The question is how to make the semantics of a robot's systematically interpretable Knowledge Level symbols cohere equally systematically with the robot's interactions with the world, such that the symbols refer to the world on their own, rather than merely because of an external interpretation we place on them. This requires that the robot be able to discriminate, identify, and manipulate the objects, events, and states of affairs that its symbols refer to [Harnad, 1992]. Grounding is accomplished in our architecture through the alignment of the Knowledge and Perceptuo-Motor Levels. Elementary symbols at the Knowledge Level are grounded in the sense that they can only attach to "the right kind" of representations at the Perceptuo-Motor Level. If we think of the Perceptuo-Motor Level as implementing categorial perception (and perhaps "categorial action"), then the elementary symbols of the Knowledge Level are the names attached to the categories. In other words, the alignment of the Knowledge and Perceptuo-Motor Level constitutes an *internal referential semantic model* of elementary symbols. Note that, like McDermott, we do not take the Tarskian stance which requires the referents of symbols to be in the world; rather, they are system-internal, similar to what Hausser proposes [Hausser, 1989], or what Harnad calls iconic representations: "proximal sensory projections of distal objects, events, and states of affairs in the world" [Harnad, 1990]. The Knowledge Level is the only level that is accessible for conscious reasoning, and also the only level that is accessible for inter-agent communication. Access to the Perceptuo-Motor Level and the Sensori-Actuator Level would not be useful for communication, as the representations and processing at these levels are too agent-centered and too agent-specific to be informative to other agents.

Since the Perceptuo-Motor Level representations serving as the grounding for symbols of the Knowledge Level are embodied (section 2.2.3), equivalent symbols may have somewhat different semantics for different agents having different bodies. We don't see that as a problem, as long as the differences are not too large.[19] Indeed, we believe that this is quite realistic in human terms as well; no two persons are likely to have *exactly* the same semantics for their concepts, which nevertheless does not prevent them from understanding each other, grosso modo at least (cf. [Rapaport, 1988]). The problems of translation and communication in general consist at least in part of establishing a correspondence between concepts (and symbols) used by the participants. It is helpful to be able to use referents in the external world as benchmarks, but one consequence of embodied semantics is that *even* if it is possible to establish these common external referents for symbols, there is still no guarantee that the symbols will actually *mean* exactly the same thing, because in effect the same referent in the world is *not* the same thing to different agents. If we accept this view, it is clear that approaches to semantics based on traditional logical model theory are doomed to fail, because they *presuppose* "identity of referents" and an unambiguous mapping from symbols to referents, the same one for all agents. Another problem is of course the presupposition that all objects are uniquely identifiable. The use of deictic representations does not impose such a condition; as far as our agents are concerned, if it looks and feels the same, it is the same. Nothing hinges on whether or not the objects in the agent's surroundings are *really* extensionally the same as the identical-looking ones that were there a moment ago or will be there a moment later.

2.5 Engaged and Disengaged Reasoning

Our architecture allows us to elegantly model two different modes of reasoning and planning, which we call *engaged reasoning* and *disengaged reasoning*. Engaged reasoning takes place when all the

[19] It is never a problem as long as agents need not communicate with the outside world (other agents), of course, cf. [Winston, 1975].

elementary symbols at the Knowledge Level that are involved in the current reasoning activity are aligned with representations at the Perceptuo-Motor Level. In practical terms, this means that the agent is reasoning about objects within its field of perception. This may require active perception to keep track of objects, or to shift attention to new objects as the reasoning progresses. For the purpose of object tracking and attention shifting, Knowledge Level sensory actions are defined and can be reasoned about like ordinary actions. Engaged reasoning can be done while the actions being reasoned about are actually being carried out, in a kind of plan-as-you-go mode with continuous monitoring of progress being made, or in a more hypothetical mode with no actions being carried out, but potentially affected objects being gauged while the plan is being developed.

Disengaged reasoning occurs when there is no alignment at all between the Knowledge Level symbols involved in the current reasoning activity and Perceptuo-Motor Level representations, e.g., when developing a plan for another place and/or another time, in a purely hypothetical fashion. This is the mode that traditional planners used to operate in all the time, by necessity. Intermediate forms of reasoning, between engaged and disengaged, are possible as well.

3 A Situated Agent: An Application of GLAIR

We now briefly discuss an application of GLAIR that is currently being developed.

3.1 Overview

Our architecture as described in section 2 can be populated with components that make up the machinery for mapping sensory inputs to *response* actions, (sensory-input $\mapsto$ actions), as does Russell in [Russell, 1991]. Next, we outline the components for a situated agent, GLAIR-agent, that give content to the levels of the architecture. Figure 2 schematically presents the structure of GLAIR-agent.

- A World model: In the general KRR system, we maintain a conscious world model.[20] This model explicitly represents the agent's knowledge (or beliefs) about the entities in its world. Representations at this level may or may not correspond to representations at the Perceptuo-Motor Level, as explained in section 2.2. As much as it might be desirable to avoid building internal models of the world,[21] having some modeling capacity is necessary, we believe.
- Specialized knowledge bases: for instance knowledge about action selection, planning, learning, experimentation, and perception.[22]
- A Kinematic/Perceptual model: At the Perceptuo-Motor Level, we maintain a model of the simple agent-level physics of the objects of interest to the agent.[23] The kinematic/perceptual model models motor capacities and motor memory that might be implemented in different ways (e.g., purely procedurally or in a network of nodes and weighted links), but we prefer the declarative approach for its ease of interpretation and debugging. It also contains perceptual

[20] Albus defines a world model as the agent's best estimate of objective reality [Albus, 1991].

[21] Situated cognition and reactive planning are proponents of avoiding world modeling, e.g., [Brooks, 1990, Suchman, 1988].

[22] The fact that we represent these knowledge bases as separate in figure 2 does not necessarily mean that there is a sharp dividing line between the two. They are all uniformly represented in a general KRR system.

[23] Even the prominent advocators of doing away with world models actually use a variety of models, some of which qualify as kinematic/perceptual models. For instance, see Chapman's work on Sonja [Chapman, 1990] where Sonja has to build a convex hull of obstacles and compute angles in order to decide the best way to avoid them.

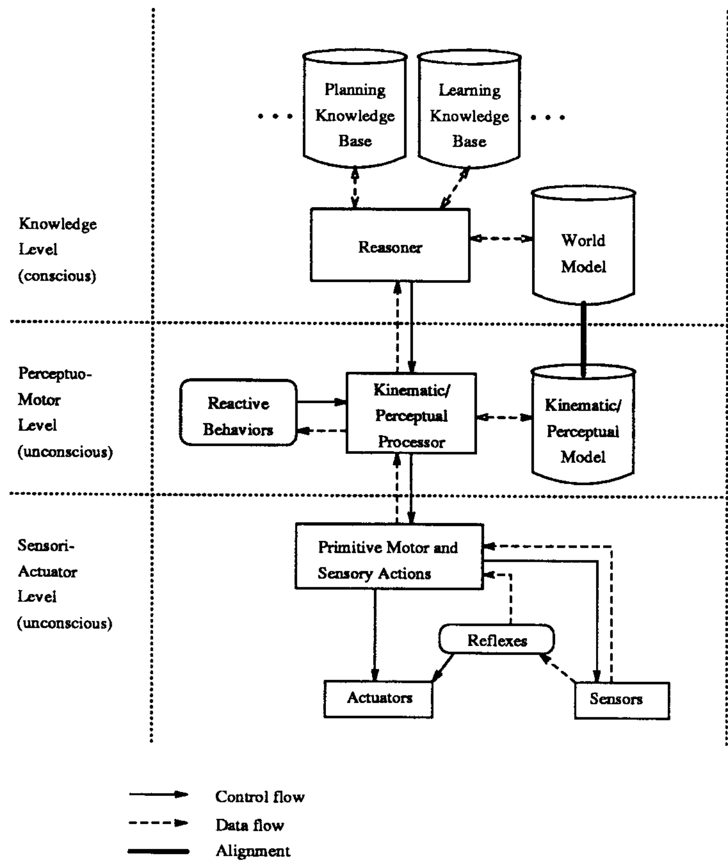

Figure 2: Schematic representation of the structure of GLAIR-agent, an agent conforming to the GLAIR architecture.

representations of perceived objects. Representations at this level are embodied and agent-centered (section 2.2.3).

- Reactive processes: Also at the Perceptuo-Motor Level, a number of independent processes monitor perceptual inputs, and control reactive behaviors of the agent.

- Primitive motor and sensory actions: At the Sensori-Actuator Level, a number of these primitive actions are implemented. A primitive motor action might be "move ahead", and a primitive sensory action might be "look at position $\langle x, y, z \rangle$ (which in turn may involve primitive motor actions). Sensing as such is not considered a primitive sensory *action*, and it goes on continuously.

- Reflexes: Also at the Sensori-Actuator Level, a number of reflexes are implemented as low-level independent processes that monitor raw sensory data and can control actuators directly, temporarily pre-empting higher level control.

Some important features of GLAIR-agent are:

- Varieties of behaviors are integrated: We distinguish between deliberative, reactive, and reflexive behaviors. At the unconscious level, behavior is generated by mechanisms with the computational power of a finite state machine (or less), whereas, at the conscious level, behavior is generated via reasoning (of Turing Machine capabilities). As we move down the architectural levels, computational and representational power (and generality) is traded off for better response time and simplicity of control. Embodied representations aid in this respect (section 2.2.3).

- Learning is incorporated: We specify that a successful agent learns from its successful interaction with the world as well as from its mistakes.
- Capacity for Engaged and Disengaged Reasoning: Our architecture allows us to model varying modes of reasoning, which range from what we call *engaged* to *disengaged* reasoning. Fully engaged reasoning occurs when all relevant constant symbols at the Knowledge Level are aligned with the Perceptuo-Motor Level, which is typical of a "hands-on" or "reason-as-you-go" mode of reasoning and acting. Fully disengaged reasoning occurs when none of the relevant constant symbols at the Knowledge Level are aligned with the Perceptuo-Motor Level, which is typical of a hypothetical or "other-place, other-time" mode of reasoning. Various intermediate forms are possible as well.

3.2 Agent Capabilities

We assume agents to possess a set of motor capabilities. The motor capabilities are primitive in the sense that (a) they cannot be further decomposed, (b) they are described in terms of the agent, and (c) no reference is made to external objects. The second property of motor capabilities is so that the success of performing an action should depend only on the agent's bodily functions and proprioceptive sensing. For example, for a robot arm, we might have the following as its motor abilities: calibrate, close-hand, raise-hand, lower-hand, move.

We further assume agents to possess motor skills. Unlike human infants, we assume these skills to be primitive at the start. An example set for a robot arm is: grasp, lift, lower, and carry. Note that these skills are complex and depend on integration of sensory data with primitive actions. For example, "grasp" needs the arm to move to the approach point and to approach the object grip point slowly with the fingers in the open position and then grasp the object (i.e., close fingers) and try to move it, testing for a secure grip.

3.3 Learning

Learning is an important feature for an agent that has to interact with the world. Since we are not designing domain-specific agents, we would like the agent to be able to cope with whatever it finds in the world. We are particularly interested in agents that learn to extend their abilities to interact with the world. For example, if all an agent knows is picking up square boxes by using its two fingers to squeeze on opposing faces of the box and lifting it, and now it faces a round ball, it has to learn to use either more fingers or use its two fingers to squeeze on diametrically opposing points of the ball.

An agent that is new to performing a particular act, or new to choosing an appropriate act, consciously reasons about its act and controls its execution. As the agent acquires more experience with an act, it is able to unconsciously pick the act and execute it. In psychological terms, this is called *automaticity*. In terms of our architecture, we say that the locus of choice/control of an act moves from the Knowledge Level to the Perceptuo-Motor Level. This places learning in the Knowledge Level.

3.4 A Repertoire of Behaviors

Our architecture provides a natural framework for modeling four distinct types of behavior, which we call reflexive, reactive, situated, and deliberative. Reflexive and reactive behaviors are predominantly unconscious behaviors, whereas situated and deliberative actions are conscious behaviors.

Reflexive behavior[24] occurs when sensed data produces a response, with little or no processing of the data. A reflex is immediate. The agent has no expectations about the outcome of its reflex. The reflexive response is not generated based on a history of prior events or projections of changing events, e.g., a gradual temperature rise. Instead, reflexive responses are generated based on spontaneous changes in the environment of the agent. These spontaneous changes in one way or another affect the agent, either *overtly* as in "fire near skin" or *covertly* as in peripheral visual perception. In anthropomorphic terms, this is innate behavior that serves directly to protect the organism from damage in situations where there is no time for conscious thought and decision making, e.g., the withdrawal reflex when sticking one's hand into a fire. Reflexive behavior does not require conscious reasoning or detailed sensory processing, so our lowest level, the Sensori-Actuator Level, is charged with producing these behaviors. Our initial mechanism for modeling reflexive behavior is to design rules of the form $T \mapsto A$, where T is a trigger and A is an action. A trigger can be a simple temporal-thresholding gate. The action A is limited to what can be expressed at the Sensori-Actuator Level, and is simple and fast.

Reactive behavior requires some processing of data and results in *situated action* [Suchman, 1988]. However, its generation is subconscious. *Situated action* refers to an action that is appropriate in the environment of the agent. In anthropomorphic terms, this is learned behavior. An example would be gripping harder when one feels an object is slipping from one's fingers, or driving a car and tracking the road. We use the term *tracking* to refer to an action that requires continual adjustments, like steering while driving. Examples of this type of reactive behavior are given in [Payton, 1986, Anderson et al., 1991].

Situated behavior requires assessment of the state the system finds itself in (in some state space) and acting on the basis of that. In anthropomorphic terms, this is learned behavior that requires only shallow reasoning. It might be modeled by the workings of a finite state automaton, for example, the Micronesian behavior described in [Suchman, 1988]. Situated action is used in *reactive planning* [Agre & Chapman, 1987, Firby, 1987, Schoppers, 1987]. Reactive planners and systems of situated behaviors share the following properties:

- Applicability of only one action is decided upon at any one time.
- Applicability of an action is decided based on the current situation and not on the history of what has happened.
- No predictions are made about what will be true after completing an action.

Deliberative behavior requires considerable processing of data and reasoning which results in action. In anthropomorphic terms, this is learned behavior that requires deep reasoning that can be modeled by a Turing machine (or first order logic), for example explicit planning and action.

3.5 Perceptuo-Motor Automata

In this section we discuss an implementation mechanism for behaviors at the Perceptuo-Motor Level. To recapitulate, we suggest that, at this level, the behaviors resulting in physical actions

[24] E.g., visual reflexes in [Regan & Beverly, 1978]: Here responses are generated to certain visual stimuli that do not require detailed spatial analysis.

specify an automaton, which we will call a PM-automaton (PMA), [Hexmoor & Nute, 1992]. A PMA is a implementation mechanism for routine activities at an "unconscious" level. A PMA is a finite state machine in which each state is associated with an act and arcs are associated with perceptions. In each PMA, a distinguished state is used to correspond to the no-op act. Each state also contains an auxiliary part we call Internal State (IS). An IS is used in arbitrating among competing arcs. Arcs in a PMA are situations that the agent perceives in the environment. Since perception is a continuous function over time, the arcs are designed to be anytime algorithms. In contrast to one-shot algorithms, these algorithms improve over time. When a PMA arc emanating from a state becomes active, it behaves like an asynchronous interrupt to the act in execution in the state. This causes the PMA to stop executing the act in the state and to start executing the act at the next state at the end of the arc connecting the two states. This means that in our model the agent is never idle, and it is always executing an act. We will illustrate the operation of PMA with an example.

B Not Last ([Drummond, 1989]): Given a table on which to assemble three blocks in a row: block A on the left, at location 1; block B in the middle, at location 2; and block C on the right, at location 3. The blocks are initially available for placement, and each block can be placed on the table once it's available. The exact means for moving the blocks does not matter: when a block is available it may be placed. The only constraint on assembly is that *block B cannot be placed last*: once A and C are down, there is not enough room to place B. B must be swept in from the left or from the right, and cannot be placed if A and C are in the way.

We build a PMA for this problem with four acts: placeA, LplaceB, RplaceB, placeC, and our start state no-op. On the arcs we have perceptions free1, free2, free3, availableA, availableB, and availableC. State transitions here are triples of the form [current state, perceptions, next state]:

- [no-op, (free1, availableA), placeA]
- [no-op, (free2, free3, availableB), RplaceB]
- [no-op, (free1, free2, availableB), LplaceB]
- [no-op, (free3, availableC), placeC]
- [placeA, (free2, free3, availableB), RplaceB]
- [placeC, (free1, free2,availableB), LplaceB]
- [RplaceB, (free3, availableC), placeC]
- [LplaceB, (free1, availableA), placeA]
- [LplaceB, (free3, availableC), placeC]
- [RplaceB, (free1, availableA), placeA]

In this example, IS in each state other than no-op is the conditions perceived in arcs leading to the state and the previous state. IS at placeA is free1, availableA, and one of (no-op, RplaceB, LplaceB). IS at RplaceB is free2, free3, availableB, and one of (no-op, placeA). IS at LplaceB is free1, free2, availableB, and one of (no-op, placeC). IS at placeC is free3, availableC, and one of (no-op, LplaceB, RplaceB). IS can be interpreted as the situation under which an act is applicable. This can be used to rewrite the PMA in terms of *situation rules* where the antecedent is a situation (i.e., conditions on arcs and the previous state) and the consequent is an act. The resulting rules are similar to Drummond's situation control rules [Drummond, 1989].

Primary mode of acquiring PMA is by converting plans in the Knowledge Level into PMA by a process described in [Hexmoor & Nute, 1992]. A PMA may become active by an intention to execte an action at the Knowledge Level. Once a PMA becomes active, sensory perception will be used by the PMA to move along the PMA arcs (i.e., PMA transitions). The sensory perceptions that form the situations on the arcs as well as subsequent actions on the PMA may be noticed at the Knowledge Level. In general, the sensory information is filtered into two streams. One stream is for the consumption of the PMA, the other for the Knowledge Level.

4 Concluding Remarks

We have presented a general architecture for autonomous agents that integrates behavior-based architectures with traditional architectures for symbolic systems. This architecture, independent of advocating a particular style of agent modeling, specifies how an agent establishes and maintains a conscious connection with its environment while mostly unconsciously processing sensory data, and filtering information for conscious processing as well as for reflexive and reactive acting. We ended our paper by instantiating the architecture with a particular agent embedded in its environment.

5 Acknowledgement

We appreciate comments made by John Pollock, Beth Preston, Donald Nute, William Rapaport, Stevan Harnad, and Chris Brown.

References

[Agre, 1988] Agre, P. (1988). *The Dynamic Structure of Everyday Life.* Technical Report 1085, MIT Artificial Intelligence Laboratory, MIT.

[Agre & Chapman, 1987] Agre, P. E. & Chapman, D. (1987). Pengi: An implementation of a theory of activity. In *Proceedings of AAAI-87, Seattle, Wa.* (pp. 268–272).

[Albus, 1991] Albus, J. (1991). Outline for a theory of intelligence. In *IEEE Transactions on Systems, Man, and Cybernetics, Vol. 21, No. 3* (pp. 473–509).

[Albus et al., 1981] Albus, J., Barbera, A., & Nagel, R. (1981). Theory and practice of hierarchical control. In *23rd International IEEE Computer Society Conference* (pp. 18–38).

[Anderson, 1983] Anderson, J. R. (1983). *The Architecture of Cognition.* Cambridge: Harvard University Press.

[Anderson et al., 1991] Anderson, S., Hart, D., & Cohen, P. (1991). Two ways to act. In *ACM SIGART Bulletin* (pp. 20–24). ACM publications.

[Ballard & Brown, 1982] Ballard, D. H. & Brown, C. M. (1982). *Computer Vision.* Prentice-Hall, Englewood Cliffs, NJ.

[Brooks, 1985] Brooks, R. (1985). *A Robust Layered Control System for a Mobile Robot.* Technical Report 864, MIT AI Labs, MIT.

[Brooks, 1987] Brooks, R. (1987). *Planning is Just a Way of Avoiding Figuring Out What to do Next.* Technical Report 303, MIT AI Labs.

[Brooks, 1990] Brooks, R. A. (1990). Elephants dont play chess. In P. Maes (Ed.), *Designing Autonomous Agents* (pp. 3–15). MIT Press.

[Card et al., 1983] Card, S., Moran, T., & Newell, A. (1983). *The Psychology of Human-Computer Interaction.* Erlbaum, Hillsdale, N.J.

[Chapman, 1990] Chapman, D. (1990). *Vision, Instruction, and Action.* Technical Report Technical Report 1204, MIT Artificial Intelligence Laboratory, MIT.

[Connell, 1992] Connell, J. (1992). Sss: A hybrid architecture applied to robot navigation. In *IEEE Conference on Robotics and Automation* (pp. 2719–2724).

[Culbertson, 1963] Culbertson, J. (1963). *The Minds of Robots.* U. of Illinois Press.

[Drummond, 1989] Drummond, M. (1989). Situated control rules. In *Proceedings Of the first conference on principles of knowledge representation* (pp. 103–113).: Morgan Kaufman.

[Firby, 1987] Firby, R. J. (1987). An investigation into reactive planning in complex domains. In *Proceedings of AAAI-87.*

[Fodor, 1983] Fodor, J. (1983). *The Modularity of Mind.* MIT Press.

[Gat, 1991] Gat, E. (1991). *Reliable Goal-Directed Reactive Control of Autonomous Mobile Robot.* Technical Report Tec, Dept. of Computer Science, Virginia Polytechnic Institute and State University.

[Harnad, 1990] Harnad, S. (1990). The symbol grounding problem. *Physica D*, 42(1-3), 335–346.

[Harnad, 1992] Harnad, S. (1992). Electronic symposium on computation, cognition and the symbol grounding problem. E-mail symposium (ftp archive at princeton.edu:/pub/harnad/sg.comp.arch*).

[Hausser, 1989] Hausser, R. (1989). *Computation of Language: An Essay on Syntax, Semantics and Pragmatics in Natural Man-Machine Communication.* New York, NY: Springer-Verlag.

[Hexmoor, 1989] Hexmoor, H. (1989). An architecture for reactive sensor-based robot. In *NASA Goddard conference on AI* Greenbelt, MD.

[Hexmoor & Nute, 1992] Hexmoor, H. & Nute, D. (1992). *Methods for Deciding what to do next and Learning.* Technical Report AI-1992-01, AI Programs, The University of Georgia, Athens, Georgia.

[Kaelbling, 1988] Kaelbling, L. (1988). Goals as parallel program specifications. In *Proceedings of AAAI-88*: Morgan Kaufman.

[Kaelbling & Rosenschein, 1990] Kaelbling, L. & Rosenschein, S. (1990). Action and planning in embedded agents. In P. Maes (Ed.), *Designing Autonomous Agents* (pp. 35–48). MIT Press.

[Laird et al., 1991] Laird, J., Huka, M., Yager, E., & Tucker, C. (1991). Robo-soar: An integration of external interaction, planning, and learning, using soar. In *Robotics and Autonomous Systems.*

[Laird et al., 1987] Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence*, 33, 1–64.

[Lakoff, 1987] Lakoff, G. (1987). *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind.* Chicago, IL: University of Chicago Press.

[Langley et al., 1991] Langley, P., McKusick, K., & Allen, J. (1991). A design for the icarus architecture. In *ACM SIGART Bulletin* (pp. 104–109). ACM publications.

[Maes, 1991] Maes, P. (1991). Action selection. In *Proceedings of Cognitive Science Society Conference.*

[McClelland et al., 1986] McClelland, J. L., Rumelhart, D. E., & Hinton, G. E. (1986). The appeal of parallel distributed processing. In D. Rumelhart, J. McClelland, & the PDP Research Group (Eds.), *Parallel Distributed Processing* chapter 1, (pp. 3–44). MIT Press, Cambridge MA.

[McDermott, 1991] McDermott, D. (1991). *Robot Planning.* Technical Report CS-861, Yale University.

[Payton, 1986] Payton, D. (1986). An architecture for reflexive autonomous vehicle control. In *Proceedings of Robotics Automation* (pp. 1838–1845).: IEEE.

[Pollock, 1989] Pollock, J. (1989). *How to Build a Person.* MIT Press.

[Pollock, 1992] Pollock, J. (1992). New foundations for practical reasoning. In *Minds and Machines.*

[Rapaport, 1988] Rapaport, W. J. (1988). Syntactic semantics: Foundations of computational natural-language understanding. In J. H. Fetzer (Ed.), *Aspects of Artificial Intelligence* (pp. 81–131). Kluwer Academic, New York NY.

[Regan & Beverly, 1978] Regan, D. & Beverly, K. (1978). Looming detectors in the human visual pathways. In *Vision Research 18* (pp. 209–212).

[Rosenblatt & Payton, 1989] Rosenblatt, J. K. & Payton, D. (1989). A fine-grained alternative to the subsumption architecture for mobile robot control. In *Proceedings of Internations Joint Conference on Neural Networks.*

[Russell, 1991] Russell, S. (1991). An architecture for bounded rationality. In *ACM SIGART Bulletin* (pp. 146–150). ACM publications.

[Schoppers, 1987] Schoppers, M. J. (1987). Universal plans for unpredictable environments. In *Proceedings 10th IJCAI* (pp. 1039–1046).

[Shapiro, 1990] Shapiro, S. C. (1990). Cables, paths, and 'subconscious' reasoning in propositional semantic networks. In *Principles of Semantic Networks.* Morgan Kaufman.

[Shapiro et al., 1988] Shapiro, S. C., Kumar, D., & Ali, S. S. (1988). A propositional network approach to plans and plan recognition. In *Proceedings of the AAAI-88 Workshop on Plan Recognition*: Morgan Kaufmann.

[Shapiro & Rapaport, 1987] Shapiro, S. C. & Rapaport, W. J. (1987). Sneps considered as a fully intensional propositional semantic network. In N. Cercone & G. McGalla (Eds.), *The knowledge frontier: essays in the representation of knowledge* (pp. 262–315). Springer, New York.

[Simmons, 1990] Simmons, R. (1990). An architecture for coordinating planning, sensing, and action. In *Proceedings the DARPA workshop* (pp. 292–297).

[Suchman, 1988] Suchman, L. A. (1988). *Plans and Situated Actions: The Problem of Human Machine Communication.* Cambridge University Press.

[Whitehead, 1992] Whitehead, S. (1992). *Reinforcement Learning for the Adaptive Control of Perception and Action.* PhD thesis, University of Rochester. Technical Report 406.

[Winston, 1975] Winston, P. H. (1985 (orig. 1975)). Learning structural descriptions from examples. In R. J. Brachman & H. J. Levesque (Eds.), *Readings in Knowledge Representation* (pp. 141–168). Morgan Kaufmann, San Mateo CA.

TECHNICAL REPORTS

Following are recent technical reports published by the Department of Computer Science, University at Buffalo. Reports may be ordered from:

Library Committee
Department of Computer Science
University at Buffalo (SUNY)
226 Bell Hall
Buffalo, NY 14260
USA

Prices are given as (USA and Canada/Other Countries). Payment should be enclosed with your order (unless we have a free exchange agreement with your institution, in which case the reports are free). Please make checks payable to "Department of Computer Science".

If your institution is interested in an exchange agreement, please write to the above address for information.

91-01 D.B. Sher, D. Milun and P. Mulhaupt: *Trainable Low-level Image Analysis Algorithms* ($2.00)

91-02 G. Succi: *Set Representations in a Subset-Equational Language* ($2.00)

91-03 A. Yuhan: *Dynamic Computation of Spatial Reference Frames In Narrative Underestanding* ($2.00)

91-04 S. Homer, A. Selman: *Complexity Theory* ($2.00)

90-24 X. He: *On Finding the Rectangular Duals of Planar Triangular Graphs, Revised* ($2.00)

91-05 X. He: *Efficient Parallel Algorithms for Two Graph Layout Problems* ($2.00)

91-06 S. Chakravarty: *Robust Tests for Stuck-open Faults: A Testable Design and a* Processing Step" ($2.00)

91-07 D. Sher, S. Revankar, S. Rosenthal: *Computer Methods in Quantitation of Cardiac Wall Parameters from Two Dimensional Echocardiograms: A Survey* ($2.00)

91-08 D. Milun, D. Sher: *Improving Sampled Probability Distributions for Markov Random Fields* ($2.00)

91-09 J. Haas, B. Jayaraman: *Automatic Synthesis of Semantics for Context-free Grammars* ($2.00)

91-10 D. Sher: *Deriving the Distribution of Edge Neighborhoods from a Library of Occluding Objects* ($2.00)

91-11 B. Jayaraman: *The SuRE Programming Framework* ($2.00)

91-12 A. Selman: *A Taxonomy of Complexity Class of Functions* ($2.00)

91-13 S. Shapiro, H. Chalupsky, H. Chou: *Connecting ARC/INFO*and SNACTor Project Report* ($2.00)

91-14 A. Shende: *Digital Analog Simulation of Uniform Motion in Representations of Physcial N-Space by Lattice-work MIMD Computer Architectures* ($10.00)