An Architecture for Action, Emotion, and Social Behavior

Joseph Bates A. Bryan Loyall W. Scott Reilly

May 1992
CMU-CS-92-144

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

This report will appear in revised form in the Proceedings of the Fourth European Workshop on Modeling Autonomous Agents in a Multi-Agent World, S.Martino al Cimino, Italy, July 1992. The proceedings will be a volume in the Decentralized AI Series published by Elsevier/North Holland.

Keywords: artificial intelligence, agent architecture, autonomous agents, emotion, goals, reactivity, interactive media, virtual reality, Oz

Abstract

The Oz project at Carnegie Mellon is studying the construction of artistically effective simulated worlds. Such worlds typically include several agents, which must exhibit broad behavior. To meet this need, we are developing an agent architecture, called Tok, that presently supports reactivity, goals, emotions, and social behavior. Here we briefly introduce the requirements of our application, summarize the Tok architecture, and describe a particular agent we have constructed that exhibits the desired social behavior.

1 The Oz Project and Broad Agents

The Oz project at Carnegie Mellon University is developing technology for artistically interesting simulated worlds [3]. We want to let human users participate in dramatically effective worlds that include moderately competent, emotional agents. We work with artists in the CMU Drama and English Departments, to help focus our technology on genuine artistic needs.

An Oz world has four primary components. There is a simulated physical environment, a set of automated agents which help populate the world, a user interface to allow one or more people to participate in the world [14], and a planner concerned with the long term structure of the user's experience [2].

One of the keys to an artistically engaging experience is for the user to be able to "suspend disbelief". That is, the user must be able to imagine that the world portrayed is real, without being jarred out of this belief by the world's behavior. The automated agents, in particular, must not be blatantly unreal. Thus, part of our effort is aimed at producing agents with a broad set of capabilities, including goal-directed reactive behavior, emotional state and behavior, social knowledge and behavior, and some natural language abilities. For our purpose, each of these capacities can be as limited as is necessary to allow us to build broad, integrated agents [4].

Oz worlds can be simpler than the real world, but they must retain sufficient complexity to serve as interesting artistic vehicles. The complexity level seems to be somewhat higher, but not exceptionally higher, than typical AI micro-worlds. Despite these simplifications, we find that our agents must deal with imprecise and erroneous perceptions, with the need to respond rapidly, and with a general inability to fully model the agent-rich world they inhabit. Thus, we suspect that some of our experience with broad agents in Oz may transfer to the domain of social, real-world robots.

Building broad agents is a little studied area. Much work has been done on building reactive systems [1, 6, 7, 10, 11, 22], natural language systems (which we do not discuss here), and even emotion systems [9, 18, 20]. There has been growing interest in integrating action and learning (see [16]) and some very interesting work on broader integration [23, 19]. However, we are aware of no other efforts to integrate the particularly wide range of capabilities needed in the Oz domain. Here we present our efforts, focusing on the structure of a particular agent designed to exhibit goal-directed reactive behavior, emotion, and some social behavior. Further discussion of the integration issues can be found in [5].

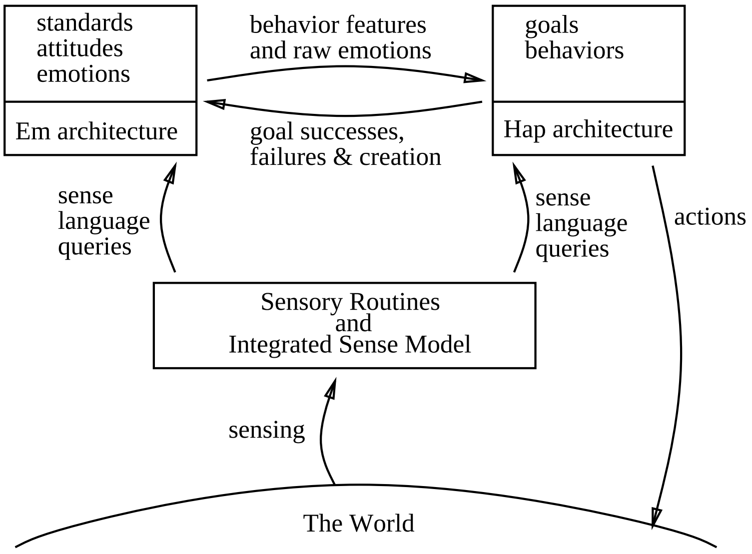

Figure 1: Tok Architecture

2 Tok and Lyotard

Through analysis of our task domain, we have concluded that the primary capabilities we want in our initial Oz agents are perception, reactivity, goal-directed behavior, emotion, social behavior, natural language analysis, and natural language generation. Our agent architecture, Tok, assigns these tasks to several communicating components. Perception, while partially task specific, is also in part handled by a pair of systems called the Sensory Routines and the Integrated Sense Model. Reactivity and goal-directed behavior are handled by Hap [17]. Emotion and social relationships are the domain of Em [21]. Language analysis and generation are performed by Gump and Glinda, respectively [13, 14]. Figure 1 shows how these components, excluding Glinda and Gump, are connected to form Tok.

In the remainder of this section we discuss the components of Tok and their integration. We illustrate using an existing Tok agent, a simulated house cat named "Lyotard", which exercises most of the capabilities of the architecture. Our goal in developing Lyotard was to build a creature that could believably pass for a cat in an Oz micro-world.

Table 1 lists the emotions and behaviors from our original informal design document for Lyotard. The emotions are those naturally available in the current version of Em, though in the end we did not use all of them. The behaviors were developed over several hours of brainstorming by several cat owners in our group. The behavioral features are used to modify details determined while Hap is performing particular

Emotions	Behaviors		Features
hope†	wanting to be pet *or brushed*	purring	curious
fear	cleaning self	arch back	content
happy	wanting to go out/in	hiss	aggressive
sad	eating	swat	ignoring
pride	wanting to eat	bite	friendly
shame	getting object (using human	escape/run away	*proud*
admiration	or other tool)	have fun	energetic
reproach	searching for something	*pouncing on creatures*	
gratification	*carrying mouse*	chasing ball/*creatures*	
remorse	playing with ball	rubbing against	
gratitude	playing with mouse	licking	
anger	*crazy hour*	watching/staring at	
love	hiding (anger/fear)	sitting on a sunny ledge	
hate	pushing things around		

†*italicized items were not included in final implementation*

Table 1: Original Lyotard Task

behaviors. They are usually derived from Lyotard's emotional state, though they also can be directly adjusted by behaviors.

2.1 The Simulated World

We are developing versions of Tok for several distinct simulation environments. Here we describe Tok within an "interactive fiction" system, where space is discrete and topological. We are also embedding Tok in an animated real-time world, where space is more continuous and geometric. This variation is briefly discussed in the conclusion, below.

The interactive fiction physical world is a very simple object-oriented simulation in which agents perform actions by invoking methods on appropriate sets of objects. These methods may alter the world, propagate sense data, and succeed or fail. Objects are connected to each other via topological relations, for example Lyotard could be *on* the table which is *in* the room. We have found this model more than adequate to express artistically interesting physical environments.

Agents sense the world via sense data objects which propagate from the item sensed through the world to the agents. These sense data convey the properties of objects, relationships between objects, and events such as the room becoming dark or Lyotard pouncing on his toy mouse. Each sense datum describes the thing sensed as a collection of property/value pairs. Unique names are not used to identify objects; agents must derive identity from other properties. Sense data can be transformed as they travel. For example speech behind a closed door can be muffled so that the words are unintelligible but the voice is recognizable or a white shirt can appear blue when seen through blue tinted glass. In general, the sense data available to an agent can be incomplete, incorrect, or absent.

2.2 Perception (Sensory Routines and Integrated Sense Model)

In the interactive fiction world, each Tok agent runs by executing a three step loop: sense, think, act. First, raw sense data is extracted from the world and recorded by the Sensory Routines. Because the world is simple, most of the perceivable world state can be determined and recorded using task independent mechanisms. The relationships between objects are represented as links, thus creating a topological graph of the newly encountered world fragment. The new data is marked with the agent's internal notion of time, and the older graphs are retained. When Hap behaviors execute, this low level memory of raw sense data can be queried for information such as "have I seen food in the kitchen in the last ten minutes?".

After the raw data are recorded in the Sensory Routines, an attempt is made to merge them into the Integrated Sense Model (ISM), which maintains the agent's best guess about the physical structure of the whole world. This requires inference, including merging sense data from different modalities, such as sight and sound, if they seem to be related, and merging new and past perceptions of seemingly identical objects. The process uses whatever (partial) property/value pairs are available in the sense data as well as topological information. Some higher-level inferences, such as deciding which of the visible objects can also be reached, are made.

Lyotard starts with an empty ISM and with no fragments in the sensory routines. As he interacts with the world these perception systems collect information. By exploring the environment he visually determines how space is connected and how objects are placed in the world. This allows him, for instance, to make a good guess later about the location of his favorite toy mouse or various soft places to sit. By executing actions which result in touching objects, he collects tactile information via the tactile sensory routine. For example, by sitting on an object which visually appeared soft, Lyotard's tactile sensory routine perceives and records the actual softness of the object. If the object is not soft, Lyotard's ISM representation of the object would change.

The continuously updated information in the sensory routines and the longer term, approximate model in the ISM are routinely queried when choosing actions or updating the emotional state of Lyotard.

2.3 Action (Hap)

Hap is Tok's goal-directed, reactive action engine [17]. It continuously chooses the agent's next action based on perception, current goals, emotional state, behavioral features and other aspects of internal state. Goals in Hap contain an atomic name and a set of parameters which are instantiated when the goal becomes active, for example (`goto` <object>). Goals do not characterize world states to accomplish, and Hap does no explicit planning. Instead, sets of actions (which for nostalgic reasons we call "plans") are chosen from an unchanging plan memory which may contain one or more plans for each goal. These plans are either ordered or unordered collections of subgoals and actions which can be used to accomplish the invoking goal. For example one plan for the above `goto` goal is the sequence: `goto-floor` of the current room, `goto-room` of the room containing <object>, `goto-object-in-room` of the <object>. Plans have testable *preconditions* which are true when the plan could apply in the current state of the world. Multiple plans can be written for a given goal, with Hap choosing between the plans at execution time. If a plan fails, Hap will attempt any alternate plans for the given goal, and thus perform a kind of backtracking search in the real world.

Hap stores all active goals and plans in a structure called the active plan tree (APT). This is a tree of alternating layers of goals and plans that represents Hap's current execution state. The APT may be thought of as an AND-OR tree, where the goals are OR nodes and the plans are AND nodes. The APT expands and contracts as goals and plans succeed and fail.

There are various annotations in the APT to support reactivity and the management of multiple top-level goals. Two important annotations are *context conditions* and *success tests*. Both of these are arbitrary testable expressions over the perceived state of the world and other aspects of internal state. Success tests are associated with each goal in the APT. When a success test is true, its associated goal is deemed to have been accomplished and thus no longer needs to be pursued. For example, in Lyotard the first step of the `goto` plan described above has a success test associated with it to determine if the agent is already on the floor of the room. This success test may allow Lyotard to skip the subgoal. Also, if Lyotard is in the process of going to the floor when some external factor, such as a human, causes him to arrive on the floor before the subgoal completes, the success test would enable him to recognize that his goal has succeeded and stop pursuing it.

Similarly, context conditions are associated with plans in the active plan tree. When a context condition becomes false its associated plan is deemed no longer applicable in the current state of the world. That plan fails and a new plan must be chosen to accomplish the invoking goal. For the `goto` plan, an appropriate context condition might be that the object of the `goto` goal appear to remain reachable. If that context condition failed, Lyotard would try other plans for going to his target, perhaps including finding a human to help out.

Figure 2 shows the concrete expression of a small plan that includes some of these annotations.

Every instance of a goal has a *priority* number used when choosing a goal to execute and an *importance* number used by Em when considering the significance of

```
(sequential-production goto (target)
       (precondition
        (and (can-see (a location ?l-me) location (node $$me))
             (know-of-in-ism (a location ?l-target) location
                             (node $$target))
             (know-of-in-ism (node $$target) reachable (node $$me))))
       (context-condition
        (and (can-see (a location ?l-me) location (node $$me))
             (know-of-in-ism (node $$target) reachable (node $$me))))
  (with (success-test (or (can-see (a location) containing (node $$me))
                          (can-see (node $$l-target) location
                                   (node $$me))))
    (subgoal goto-floor $$l-me))
  (with (success-test (can-see (node $$l-target) location (node $$me)))
    (subgoal goto-room $$l-target))
  (with (success-test (or (can-see (node $$target) containing
                                   (node $$me))
                          (can-see (node $$target) supporting
                                   (node $$me))))
    (subgoal goto-object-in-room $$target)))
```

Figure 2: Example Hap Plan in Lyotard

the goal. These annotations are assigned to instances of goals rather than to types of goals, because identical goals could have different priority or emotional importance depending on the context in which they arise. In Lyotard, going to the kitchen to get food has a higher priority than going somewhere in pursuit of an exploration goal.

After sense data is processed, Hap begins execution by modifying the APT based on changes in the world. For every goal and plan in the APT, the associated success test or context condition is evaluated. Goals whose success test is true and plans whose context condition is false are removed. Next one of the leaf goals is chosen. This choice is made by a goal arbiter which prefers high priority goals and prefers continuing a line of expansion among goals of equal priority. If the chosen goal is a primitive action, it is executed. Otherwise it is a subgoal, in which case the plan library is indexed and the plan arbiter chooses a plan for this new goal from those whose preconditions are true. The plan arbiter will not choose plans which have already failed to achieve this goal instance, and prefers more specific plans over less specific ones (a measure of specificity is encoded with each plan). After either executing the primitive act or expanding the chosen subgoal, the execution loop repeats.

To date we have found Hap's mechanisms adequately flexible for our needs. However, additional organizing principles are needed to make them easily usable. In Lyotard we cluster related goals and plans into conceptual structures that we call *behaviors*. Each behavior represents a recognizable, internally coherent unit of action. These behaviors are usually activated by a single goal, which can be created in the pursuit of another goal or by a top-level demon.

As mentioned earlier, Lyotard's behaviors are shown in Table 1. An example behavior is wanting-to-be-pet, which represents plans such as finding a person and then purring or rubbing against their leg, or otherwise relaxing in a comfortable place with the expectation that a human should sense the Lyotard's desire and pet him. When the behavior is active, Lyotard displays coherent action toward this end. Section 3 provides examples of additional behaviors.

2.4 Emotion and Social Relationships (Em)

Em models emotional and certain social aspects of the agent. It is based on ideas of Ortony et al. [20]. Like that work, Em develops emotions from a cognitive base: external events are compared with goals, actions are compared with standards, and objects are compared with attitudes. Most of Em's possible emotions are shown in Table 1. We describe them very briefly here, but see [21] for details.

As Hap runs, goals are created, succeed, and fail. As these events occur, Hap informs Em, and Em in turn uses this information to generate many of its emotions. Happiness and sadness occur when the agent's goals succeed or fail. The degree of happiness or sadness depends on the importance of the goal to the agent, which is provided by the agent builder. Lyotard feels a greater degree of happiness when he succeeds in satisfying an active eating goal than when he satisfies an active relaxation goal because the builder of Lyotard labeled the former as more important.

Hope and fear occur when Em believes that there is some chance of an active goal succeeding or failing. For example, Lyotard feels hope when he sees a human about to feed him. The amount of hope or fear is determined by a function of the goal's importance and the believed likelihood of success or failure.

Pride, shame, reproach, and admiration arise when an action is either approved or disapproved. These judgments are made according to the agent's *standards*, which represent moral beliefs and personal standards of performance. Pride and shame occur when the agent itself performs the action; admiration and reproach develop in response to others' actions. Lyotard uses only the most primitive standards, do-not-cause-my-goals-to-fail and help-my-goals-to-succeed, so he will feel reproach toward an agent who shoves him from his soft chair as this causes the failure of his relaxation goal.

Anger, gratitude, remorse and gratification arise from combinations of other emotions. An agent shoving Lyotard from his chair not only causes reproach toward the agent, but also causes sadness in Lyotard due to the failure of Lyotard's relaxation goal. The sadness and reproach combine to produce the composite emotion of anger toward the agent. Similarly, gratitude is a composite of happiness and admiration, remorse is sadness and shame, and gratification is happiness and pride.

Em's final two emotions, love and hate, arise from noticing objects toward which the agent has positive or negative *attitudes*. In Lyotard we use attitudes to help model the human-cat social relationship. Lyotard initially dislikes the user, a negative attitude, and this attitude varies as the user does things to make Lyotard angry or grateful. As this attitude changes, so will the degree of his emotion of love or hate, when the human is nearby.

Emotions (but not attitudes) should fade with time, and Em models this decay. An agent will feel love when close to someone liked. This will fade if the other agent leaves, but the attitude toward that agent will remain relatively stable.

2.5 Behavioral Features

Behavioral features modulate the activity of Hap. They are adjusted by Hap or Em to vary the ways in which Hap achieves its goals. Em adjusts the features to express emotional influences on behavior. It continuously evaluates a set of functions that control certain features based on the agent's emotional state. Hap modifies the features when it wants to force a style of action. For example, it may decide to act friendly to get what it wants, even if the agent isn't feeling especially friendly.

Features may influence several aspects of Hap's execution. They may trigger demons that create new top-level goals. They may occur in the preconditions, success tests, and context conditions of plans, and so influence how Hap chooses to achieve its goals. Finally, they may affect the precise style in which an action is performed.

Lyotard's behavioral features are listed in Table 1. One such feature is `aggressive` which arises whenever Lyotard is either angry or mildly afraid (which might be considered bravado). The aggressive feature may affect Hap by giving rise to a new goal, such as bite-human, by influencing the choice of plan for a goal, such as nipping instead of meowing to attract attention, or by modifying the style of an action, such as swatting a toy mouse a little more emphatically than usual.

We have no structured set of features, and know of no source that suggests one. Besides those in Lyotard, we have seen the following suggested: curious, belligerent, persistent, depressed, patient [8]; timid, reckless, quiet, arrogant [12]. The feature mechanism, while very *ad hoc*, appears to provide a useful degree of abstraction in the interface between emotion and behavior.

3 The Behavior of Lyotard

To our knowledge, whether an agent's behavior produces a successful suspension of disbelief can be determined only empirically. The agent must be embedded in a world, and a variety of users must report their subjective experience with the agent. For us this evaluation is an on-going effort, which we will attempt to report in the literature [15] and to convey by demonstration.

```
Lyotard:  (*GO-TO "the bedroom").
          (*GO-TO "the sunroom").
          (*GO-TO "the spare room").
          (*JUMP-ON "the chair").
          (*SIT-DOWN).
          (*LICK "Lyotard").
          (*LICK "Lyotard").

Player:   (*GO-TO "the spare room").
Lyotard:  (*JUMP-OFF "the chair").
          (*RUN-TO "the sunroom").

Player:   (*GO-TO "the sunroom").
Lyotard:  (*LOOKAROUND NERVOUSLY).
Player:   (*PET "Lyotard").
Lyotard:  (*BITE "Player").
          (*RUN-TO "the diningroom").

Player:   (*GO-TO "the spare room").
Lyotard:  (*LOOKAROUND NERVOUSLY).
          (*GO-TO "the sunroom").
          (*POUNCE-ON "the superball").
          (*LOOKAT "the superball").
          (*NUDGE "the superball").
          (*POUNCE-ON "the superball").
          (*POUNCE-ON "the superball").

Lyotard:  (*GO-TO "the diningroom").
Lyotard:  (*GO-TO "the kitchen").
          (*MEOW).
Player:   (*GO-TO "the sunroom").
Lyotard:  (*MEOW).
Player:   (*GO-TO "the diningroom").
Lyotard:  (*WAIT).
Player:   (*TAKE "the glass jar").

Lyotard:  (*GO-TO "the diningroom").
Player:   (*GO-TO "the kitchen").
Lyotard:  (*JUMP-ON "the table").

Lyotard:  (*JUMP-OFF "the table").
          (*GO-TO "the kitchen").
          (*MEOW).
Player:   (*POUR "the glass jar" IN
                 "the kitty bowl").

Lyotard:  (*EAT "the sardine").
          (*EAT "the sardine").
          (*EAT "the sardine").
          (*EAT "the sardine").
          (*EAT "the sardine").

Player:   (*PET "Lyotard").
Lyotard:  (*CLOSE-EYES LAZILY).
Player:   (*TAKE "Lyotard").
Lyotard:  (*CLOSE-EYES LAZILY).
```

Table 2: Section of an interaction with Lyotard

In an attempt to provide the reader of this non-interactive text with some sense of Lyotard's behavior, we present in Table 2 a small excerpt of a session with Lyotard. In this session a human user interacted with Lyotard in a simulated six room house. Because we are interested in the actions of the agents, the figure contains debugging output showing the actions of each agent from an omniscient perspective. The normal output from the system to the human user has been omitted: English descriptions of what the human perceives, prompts for the human's action, etc. Blank lines have also been included to improve clarity.

Just prior to the beginning of this excerpt, Lyotard had successfully finished an exploration goal. This success was passed on to Em which made Lyotard mildly happy. This happy emotion led to the content feature being set. Hap then noticed this feature as active and decided to pursue a behavior to find a comfortable place to sit. This decision was due to the presence of a high-level amusement goal and the content feature. Other behaviors were under consideration both in pursuit of the amusement goal and in pursuit of Lyotard's other active high-level goals.

In finding a comfortable place to sit, Lyotard (using the ISM) remembers places that he believes to be comfortable and chooses one of them, a particular chair in the spare room. He then goes there, jumps on the chair, sits down, and starts cleaning himself for a while.

At this point, the human user whom Lyotard dislikes walks into the room. The dislike attitude, part of the human-cat social relationship in Em, gives rise to an emotion of mild hate toward the user. Further, Em notices that one of Lyotard's goals, not-being-hurt, is threatened by the disliked user's proximity. This prospect of a goal failure generates fear in Lyotard. The fear and hate combine to generate a strong aggressive feature and to diminish the previous content feature. In this case, Hap also has access to the fear emotion itself to determine why Lyotard is feeling aggressive. The fear emotion and proximity of its cause combine in Hap to give rise to an avoid-harm goal, while the aggressive feature gives rise to a goal to threaten the user. In this case the avoid-harm goal wins out, creating a subsidiary escape/run-away behavior that leads Lyotard to jump off the chair and run out of the room. Since Lyotard is no longer on the chair, the plan he was executing in pursuit of his relaxation goal no longer makes sense. This is recognized by the appropriate context condition evaluating to false, which causes the plan to be removed from the APT.

At this point some time passes (not shown in the trace), during which Lyotard does not see the user. This causes the success test of the escape/run-away goal to fire and thus the goal to be removed from the APT. However, when the user follows Lyotard into the sunroom, these goals are again generated. As the user then tries to pet Lyotard, Lyotard sees the action, and notices that the actor trying to touch him is one toward whom he feels mild hate. This combination generates a third goal, respond-negatively-to-contact. Lyotard responds to this rather than to either of the first two goals or any of his other goals because we annotated it as having a higher priority than the others due to its immediacy. Further refinement of this goal through a series of plan choices leads to Lyotard biting the player.

As the player leaves Lyotard alone, the emotions engendered by the player start to decay, and Lyotard again pursues his amusement goal. This time he is no longer content, which is one of several changes to his emotional state, so a slightly different set of amusement choices are available. He chooses to play with one of his toys, and so goes to find his superball.

As the simulation has progressed, Lyotard's body has been getting more hungry. At this point his hunger crosses a threshold so that his mind notices it as a feeling of hunger. This triggers a feeding goal causing him to go to his bowl, but it is empty so he complains by meowing. After a while, he gives up on this technique for getting food, so he tries another technique; he goes looking for food himself. He remembers places where he has seen food that was reachable, and goes to one of them, passing by the user in the process. At this point he again feels fear and aggression, but he ignores these feelings because dealing with the hunger is more important to him. As he reaches the location he expected to find the food, he notices that it is gone (taken by the user when Lyotard couldn't see him), so Lyotard again considers other techniques to get food. He could try to find a human and suggest he be fed, but instead he chooses to try his bowl again. This time the human feeds him, and Lyotard eats. As he eats he feels happy because his emotionally important goal of eating is succeeding, and he also feels gratitude toward the user, because he believes the user helped to satisfy this goal. This gratitude in turn gradually influences Lyotard's attitude toward the user from dislike to neutral.

Now when the user pets Lyotard, Lyotard responds favorably to the action by closing his eyes lazily. Lyotard wants to be pet because he no longer dislikes or fears the user. Thus, being pet causes a goal success which causes happiness, and because the goal success was attributed to the user, increases gratitude toward the user. The result is that Lyotard now strongly likes the player.

The trace we have shown was produced by the interactive fiction version of Oz, which is written in Common Lisp. Of the 50,000 lines of code that comprise Oz, the Tok architecture is roughly 7500 lines. Lyotard is an additional 2000 lines of code. On an HP Snake (55 MIPS), each Tok agent takes roughly two seconds for processing between acts. (Most of this time is spent sensing, which suggests that even in the interactive fiction domain it may be desirable to use task specific selective perception.)

4 Conclusion and Future Work

We have described Tok, an architecture that integrates mechanisms for perception, reactivity, goals, emotion, and some social knowledge. Lyotard, a particular small agent, has been built in Tok and exhibits, we believe, interesting behavior.

We are engaged in several efforts to extend Tok. First, Gump and Glinda, our natural language components, are attached to Tok only as independent Lisp modules invocable from Hap rules. It would be best if they were expressed as complex behaviors written directly in Hap. We have increasingly observed similarities in the mechanisms of Hap and Glinda, and are exploring the possibilities of merging them fully.

Second, since the Oz physical world and agent models are computer simulated, we have the opportunity to embed (possibly imprecise) copies inside Tok for use by an envisionment engine. This might allow Tok, for instance, to consider possible re-orderings of steps in behaviors, to model and consider the internal states of other agents, and generally to make decisions based on a modicum of foresight.

Finally, we are in the process of building a real time, multi-agent, animated Oz world. This is imposing hard timing constraints and genuine parallelism on Tok, and is causing substantial changes to the implementation and smaller changes to the architecture. We expect to show a video tape of this system at the workshop.

It has been suggested to us that it may be impossible to build broad, shallow agents. Perhaps breadth can only arise when each component is itself modeled sufficiently deeply. In contrast to the case with broad, deep agents (such as people), we have no *a priori* proof of the existence of broad, shallow agents. However, at least in the Oz domain, where sustained suspension of disbelief is the criteria for success, we suspect that broad, shallow agents may be possible. This work is an experimental effort to judge the issue.

5 Acknowledgments

This research was supported in part by Fujitsu Laboratories, Ltd. We thank Phoebe Sengers, Peter Weyhrauch, and Mark Kantrowitz for their broad and deep assistance.

References

[1] Philip E. Agre and David Chapman. Pengi: An implementation of a theory of activity. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, July 1987.

[2] Joseph Bates. Computational drama in Oz. In *Working Notes of the AAAI-90 Workshop on Interactive Fiction and Synthetic Realities*, Boston, MA, July 1990.

[3] Joseph Bates. Virtual reality, art, and entertainment. *PRESENCE: Teleoperators and Virtual Environments*, 1(1):133–138, 1992.

[4] Joseph Bates, A. Bryan Loyall, and W. Scott Reilly. Broad agents. In *Proceedings of AAAI Spring Symposium on Integrated Intelligent Architectures*, Stanford, CA, March 1991. Available in *SIGART Bulletin*, Volume 2, Number 4, August 1991, pp. 38-40.

[5] Joseph Bates, A. Bryan Loyall, and W. Scott Reilly. Integrating reactivity, goals, and emotion in a broad agent. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, Bloomington, IN, July 1992.

[6] Rodney Brooks. Intelligence without representation. In *Proceedings of the Workshop on the Foundations of Artificial Intelligence*, June 1987.

[7] Rodney Brooks. Integrated systems based on behaviors. In *Proceedings of AAAI Spring Symposium on Integrated Intelligent Architectures*, Stanford University, March 1991. Available in *SIGART Bulletin*, Volume 2, Number 4, August 1991.

[8] Jaime Carbonell. Computer models of human personality traits. Technical Report CMU-CS-79-154, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, November 1979.

[9] Michael Dyer. *In-Depth Understanding.* The MIT Press, Cambridge, MA, 1983.

[10] James R. Firby. *Adaptive Execution in Complex Dynamic Worlds.* PhD thesis, Department of Computer Science, Yale University, 1989.

[11] Michael P. Georgeff, Amy L. Lansky, and Marcel J. Schoppers. Reasoning and planning in dynamic domains: An experiment with a mobile robot. Technical Report 380, Artificial Intelligence Center, SRI International, Menlo Park, CA, 1987.

[12] Eduard Hovy. *Generating Natural Language under Pragmatic Constraints.* Lawrence Erlbaum Associates, Hillsdale, NJ, 1988.

[13] Mark Kantrowitz. Glinda: Natural language text generation in the Oz interactive fiction project. Technical Report CMU-CS-90-158, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1990.

[14] Mark Kantrowitz and Joseph Bates. Integrated natural language generation systems. In R. Dale, E. Hovy, D. Rosner, and O. Stock, editors, *Aspects of Automated Natural Language Generation*, volume 587 of *Lecture Notes in Artificial Intelligence*, pages 13–28. Springer-Verlag, 1992. (This is the Proceedings of the Sixth International Workshop on Natural Language Generation, Trento, Italy, April 1992.).

[15] Margaret Kelso, Joseph Bates, and Peter Weyhrauch. Live simulation of a model for dramatic virtual worlds. Working Paper, Oz Project, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA., May 1991.

[16] John Laird, editor. *Proceedings of AAAI Spring Symposium on Integrated Intelligent Architectures*, March 1991. Available in *SIGART Bulletin*, Volume 2, Number 4, August 1991.

[17] A. Bryan Loyall and Joseph Bates. Hap: A reactive, adaptive architecture for agents. Technical Report CMU-CS-91-147, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, June 1991.

[18] Erik T. Mueller. *Daydreaming in Humans and Machines.* Ablex Publishing Corporation, 1990.

[19] Allen Newell. *Unified Theories of Cognition.* Harvard University Press, Cambridge, MA, 1990.

[20] A. Ortony, G. Clore, and A. Collins. *The Cognitive Structure of Emotions.* Cambridge University Press, 1988.

[21] W. Scott Reilly and Joseph Bates. Building emotional agents. Technical Report CMU-CS-92-143, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, May 1992.

[22] Reid Simmons. Concurrent planning and execution for a walking robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Sacramento, CA, 1991.

[23] S. Vere and T. Bickmore. A basic agent. *Computational Intelligence*, 6:41–60, 1990.