A Mind Model for Multimodal Communicative Creatures & Humanoids

Kristinn R. Thórisson

The Media Laboratory☎

Massachusetts Institute of Technology

kris@media.mit.edu http://xenia.media.mit.edu/~kris

Published in
INTERNATIONAL JOURNAL OF APPLIED ARTIFICIAL INTELLIGENCE
1999, Vol. 13 (4-5), 449-486.

☎ Now at Soliloquy Inc., 255 Park Avenue South, New York, NY

Abstract

This paper presents a computational model of real-time task-oriented dialogue skills. The architecture, termed *Ymir*, bridges between multimodal perception and multimodal action and supports the creation of autonomous computer characters that afford full-duplex, real-time face-to-face interaction with a human. Ymir has been prototyped in software, and a humanoid created, called *Gandalf*, capable of fluid multimodal dialogue. Ymir demonstrates several new ideas in the creation of communicative computer agents, including *perceptual integration of multimodal events*, *distributed planning and decision making*, an *explicit handling of real-time*, and *layered input analysis* and *motor control* with *human characteristics*. This paper describes the architecture and explains its main elements. Examples of implementation and performance are given, and the architecture's limitations and possibilities are discussed.

Keywords: Architectures of mind, autonomous agents, human-humanoid interaction, real-time, face-to-face communication.

Introduction

We humans are naturally endowed with multimodal input/output capabilities. Multimodal interactions happen between people most every day: we exchange glances, gesture to each other, speak and make facial expressions. The purpose of these interactions is usually to communicate certain information to, and receive information from others. Most of the time such actions are performed in the context of a task or a particular goal in a meaningful situation. In other words, we perform situated communication on a regular basis.

In contrast to research which is focused on various subsets of human communication, the work described in this paper focuses on the full range of human multimodal interaction. The discussion will therefore center on agents with human-like communication skills—computer agents that bear a resemblance to humans in *appearance* and *skills* as far as their communicative apparti are concerned, as opposed to insect-like communication or artificial communication protocols. This distinction is important because so much of our face-to-face communication is based on assumptions about the skills of the participants (i.e. competence and level of intelligence), as well as their visual appearance and spatial representation. One can of course *imagine* communication with alien creatures that are very different from humans. Still, to be able to interact successfully with such aliens we would have to assume certain human-like constants in the way they use language, gesture and body language, and the way they perceive and interpret the world.

Artificial agents in the current context can be thought of as {1} having a body, {2} having task-related knowledge, and {3} having goals, usually conveyed to them by their users, and {4} having communicative skills appropriate for the tasks they perform. A vacuum-cleaning robot is a good example of a situated agent with a body, knowledge about a task, and a specific goal to serve its user. How users convey their wishes and intent to the agent is an issue of human-computer interface design. For example, Chin (1991) describes an agent that gives users advice about UNIX commands during interactive sessions. This system is a text-based natural language system using a keyboard as the input device, and written English as the language of communication. Such agents rely on the traditional interaction hardware of keyboard, mouse and monitor to interact with their users. In contrast, the model here is *face-to-face* interaction between humans: we want to communicate naturally with the virtual character, keeping the communication channel as broad as possible. The term "face-to-face" not only to refers to the presence of faces but in general to embodied, co-present, co-temporal, non-mediated communication. This means that the interaction is multimodal. There are several tasks where human-like interaction skills can make the task of "programming" agents more straight-forward: If our vacuum-cleaning robot has multimodal perception and multimodal interpretive skills we can simply point into a corner and tell it to "Vacuum that corner tomorrow." In this case the communicative vacuum cleaner would have very human communication skills. Laurel (1992) lists some other chores that agents with such communicative skills

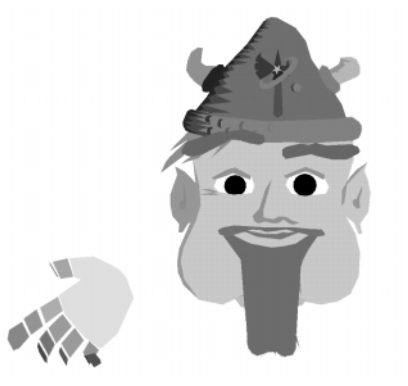


FIGURE 1. The prototype humanoid Gandalf's face and hand, comprising a total of 23 df.

might do well: coaching, tutoring, providing help, following orders, reminding, advising and entertaining, e.g. playing against, playing with and performing.

We already know a lot about how to represent the topic knowledge needed for many tasks. What has been missing is a general architecture that can integrate the critical pieces of multimodal real-time dialogue in a computer character with one or more of the above skills.

This paper describes Ymir,[1] a model well suited for creating autonomous creatures capable of human-like communication with real users. A prototype agent called Gandalf, created in the Ymir architecture, will also be described (FIGURE 1). Ymir does essentially what Fehling et al. (1988) call resource-bounded problem solving. The problem is *task-oriented dialogue*; the resources are *time*, *information* and *computational power*. Dialogue has the additional quality of encompassing many features inherent in other tasks—dialogue planning, gaze control, turn-taking, multimodal actions all have an equivalent in complex tasks where actions on several levels of detail, such as movement of arms, sensors and body have to be coordinated to meet high-level goals. On the practical side, Ymir is intended to be used for creating synthetic characters, softbots, even robots, whose purpose in life is to receive commands from humans through face-to-face interaction, ask questions when appropriate, but otherwise do the job as best their knowledge allows them to. On the theoretical side, Ymir could be used to test theories about human discourse, because it provides the possibility to turn certain dialogue actions on and off at will—something that was impossible to do before, even with a skilled actor. This would for example be very useful in testing theories of multimodal miscommunication (cf. Traum & Dillenbourg, 1996), the grounding process (cf. Clark, 1992) and collaboration principles in dialogue (Grice, 1989). The current version of Gandalf has already shown the value of the system for this purpose (Cassell & Thórisson, 1998).

Isolated parts of Ymir have been presented elsewhere (Thórisson 1998, 1997) and the theoretical underpinnings and assumptions of the model can be found in Thórisson (1996, 1995). Here I give a comprehensive overview of the model, give examples of an implementation and discuss how the model can be extended to a broader range of behaviors. But first we will look at related work and take a brief look at face-to-face interaction the way it has most often been conducted in the last few millennia: between people.

1 The name "Ymir" comes from the Icelandic Sagas, written around 1300 A.D. Ymir was a giant that the Nordic gods killed and whose carcass they subsequently used to create the Heavens and Earth out of. One of the creatures that sprung to life in the newly formed Earth was *Gandalf.* Ymir is pronounced *e-mir*, with the accent on the first syllable.

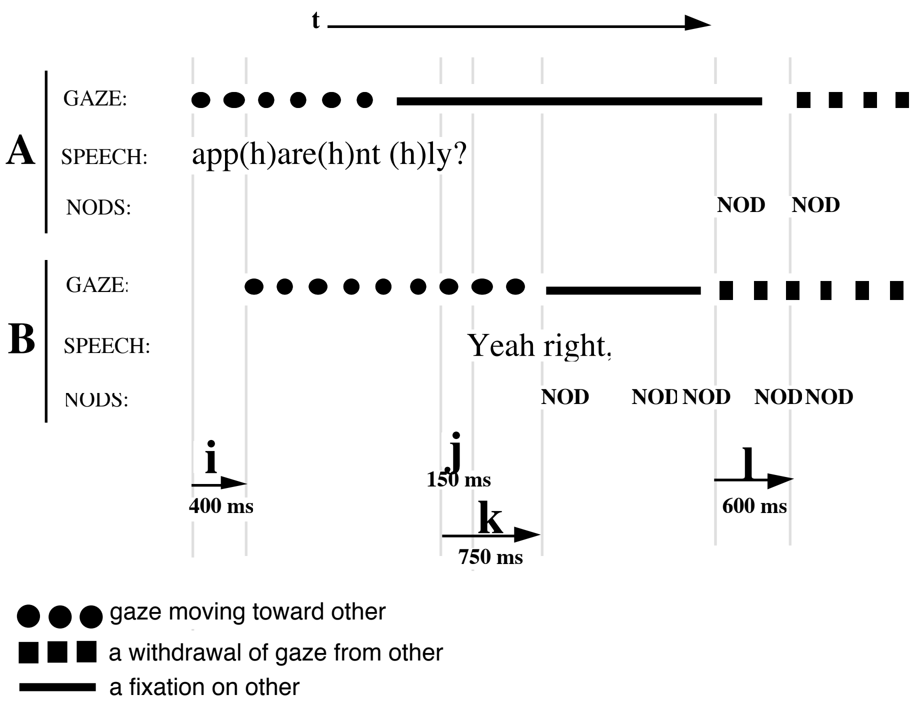
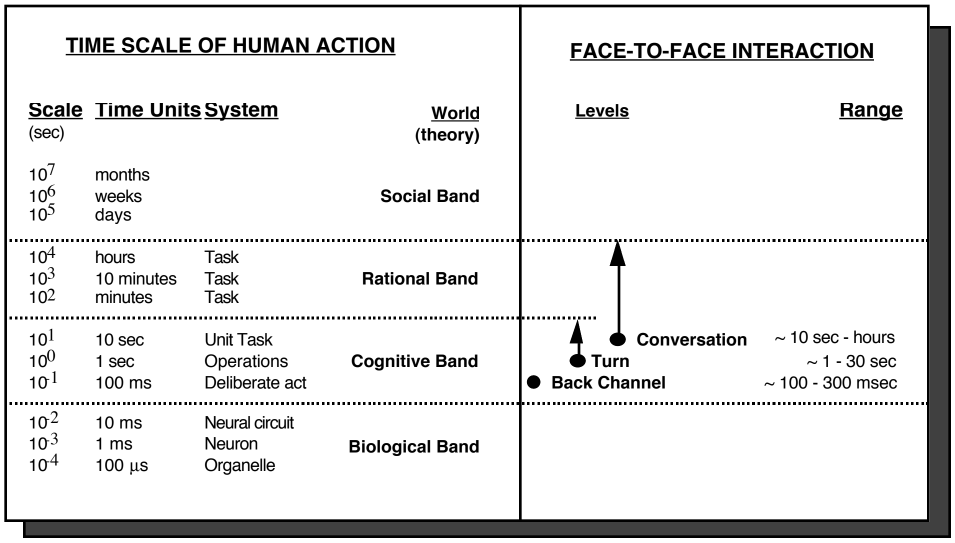


FIGURE 2. Transcript spanning 3 seconds of a typical two-person conversation (A and B), showing the timing of speech, gaze and head nods for each conversant (adapted from Goodwin (1981)). "A brings her gaze to the recipient [B]. B reacts to this by immediately bringing her own gaze to A. The two nod together and then ... withdraw [gaze] from each other, occupying that withdrawal with a series of nods" (Goodwin 1981, p. 119). Intervals i, j, k and l represent (approx.) reaction times of the listener (B) to the speaker's (A) multimodal actions; these all happen in under 1 second: Interval i is a delay in reciprocal gaze behavior; j is the time it takes B to realize that she can respond without interrupting—i.e. a turn transition; k is the time available to B for scheduling head nods in response to the same event; l is the time it takes B to reciprocate A's gaze withdrawal. Question mark indicates rising intonation.

Multimodal Face-to-Face Interaction

FIGURE 2 shows three seconds of face-to-face interaction between two participants (Goodwin, 1981). While only plotting three information types, head nods, speech content and gaze, the complexity of the open-loop coordination between the conversants is quite evident in this chart. And this is not the whole story: Other sources of information, such as intonation (Pierrehumbert & Hirschberg, 1990), gesture (Rimé & Schiaratura, 1991, Goodwin, 1986) and facial expression (Ekman & Friesen, 1978) enter constantly into the dialogue, increasing the flexibility of conversants to express themselves, and increasing the number of pieces of information that the listeners need to "put back together" in their minds. However, the additional (and often redundant) data makes it easier to follow the dialogue and thus makes it more robust (Bolt, 1987). The *communicative encounter* can be decomposed into *turns*, the turns into *multimodal actions*, the multimodal actions into *motor movements*. Each component in the hierarchy has timing relationships to other components. When looked at from a real-time perspective, these



TIME SCALE OF HUMAN ACTION				FACE-TO-FACE INTERACTION	
Scale (sec)	Time Units	System	World (theory)	Levels	Range
10^7	months				
10^6	weeks		Social Band		
10^5	days				
10^4	hours	Task			
10^3	10 minutes	Task	Rational Band		
10^2	minutes	Task			
10^1	10 sec	Unit Task		● Conversation	~ 10 sec - hours
10^0	1 sec	Operations	Cognitive Band	● Turn	~ 1 - 30 sec
10^{-1}	100 ms	Deliberate act		● Back Channel	~ 100 - 300 msec
10^{-2}	10 ms	Neural circuit			
10^{-3}	1 ms	Neuron	Biological Band		
10^{-4}	100 µs	Organelle			

FIGURE 3. Comparison between the timing in face-to-face interaction and the time scales of human action as classified by Newell (1990) (from Thórisson (1994)).

actions and plans turn out to span several orders of magnitude of time (FIGURE 3). This structure needs to be captured in any system attempting to model real-time dialogue.

Related Work

Cassell et al. (1994) describe a hybrid architecture for automatic speech and gesture generation for two graphical characters. The characters interact with each other using speech, gaze, intonation, head, face and manual gesture. The system employs what the authors call Parallel Transition Networks, in which synchronization between gestures and speech is accomplished as simultaneously executing finite state machines. The system is not real time, and lacks the necessary perceptual constructs for dialogue with real human users. But it is a significant step in tying together multimodal action generation at the planning level, from the phoneme level up to the phrase and full utterance.

Hap is an architecture for creating broad agents with goals, emotions, planning and perceptuo-motor capabilities (Loyall & Bates, 1992, Bates, Loyall & Reilly, 1994) and has been used for both text-based and graphical worlds. It addresses flexibility of plan execution and goal-directed planning, as well as real-time natural language generation. The target problem that the Hap architecture is aimed at addressing is more limited than Ymir; Ymir being more of a meta-structure (it can accommodate the Hap method of planning). Another rather important difference lies in the complexity of the kind of perception and motor output it addresses: Like Cassell's et al. (1994) work, Hap is directed at simulated characters that interact with each other inside simulated worlds. Ymir deals primarily with dialogue between synthetic characters and *real humans*, using the richness of multimodal information available in face-to-face conversation, of which virtual worlds are a subset.

A movement that has been called *behavior-based A.I.* (c.f. Maes, 1990) is driven by an interest in modelling animal intelligence. A good example of this approach is Blumberg's ethology-inspired architecture (Blumberg, 1996, Blumberg & Galyean, 1996, Maes 1989). This architecture, as many other behavior-based approaches (Wilson, 1991, Agre & Chapman, 1987) is very good for effective, fast decision making and motor control, and allows some learning. The main drawbacks of such architectures is their lack of methods to deal with actions that span more than one or two orders of magnitude in time, as well as a lack of addressing human-specific skills such as language. Long-term planning is difficult, if not impossible, to implement directly within these systems.

The NASREM architecture (Albus et al., 1987) incorporates knowledge gleaned from animal research on sensory-motor capabilities and integrates this into a comprehensive scheme for autonomous and tele-robot control. The system contains multiple levels of processing, each level containing the three components of sensory processing, world modeling and task decomposition. A global data storage is accessible from any level, but sensory modules also receive information from the level below, and task modules receive input from the level above. There are five levels all together: *mission*—information relating to a full mission, *service*—information related to parts of a mission, *task*—sub-components of a service, *elemental move*—transition from symbolic commands of movements to spatially-defined commands, *primitive move*—generates smooth trajectories and *servo*—simple hardware control. Although the layered approach of this model sounds promising for achieving the best of both worlds—fast responses to time-constrained events and slower responses to less time-constrained events—it has been criticized for trying too hard to encompass all possible systems, and thus losing its descriptive power (Thorpe, 1992). Ymir uses three layers to accomodate the *mission*, *service* and *task*-level behaviors; an Action Scheduler takes care of *elemental moves* and the *primitive moves*. The *servo* process is assumed to be part of a "dumb" motor control system, a fixed-frames-per-second animation system in the case of graphics.

The Soar architecture (Laird & Rosenbloom, 1995) has, in it's most recent incarnations, also defined layers of processing, where layers higher up do more flexible processing but are slower than those lower down. The layers of Soar are defined around a production-rule structure—a type of knowledge representation. In contrast, the layers in Ymir are defined according to the real-world demands made on the system. This links Ymir more tightly to real-world performance, allows more flexibility when implementing various mechanisms at each level. It assumes less about the particular types of processes that accomplish mental tasks and says instead more about how they need to perform and what they need to communicate about.

Many of the systems developed for autonomous robots, animation and agent creation address pieces of the multimodal pie—action generation, sensory fusion, speech and planning—but lack a single framework for the integration of all of these, making it very difficult to modify any one of them to apply to communicative humanoids. A strong dichotomy exists in many of the prior systems between language capability and action generation/coordination. In humans these are obviously seamlessly integrated most of the time. A few, like Cassell et al.'s system (1994), integrate both in a consistent way, but miss out on the full perception-action loop. Hap is a relatively broad architecture that integrates planning, emotion and natural language in a concise way, but it also has a weakness in simple perception and motor systems, making it difficult to use with the richer set of output mechanisms and input data found in the multimodal behavior of real users. Finally, a last bit of connective tissue has been sorely missing between higher-level mental function and the very low-level motor processing and coordination. In behavior-based systems, interfaces between action control modules are defined at a relatively low level—and their greatest problem by far is adding high-level cognitive skills such as natural language capabilities.

Task oriented face-to-face dialogue is an interesting problem for A.I. research because it brings together in one system many problems that have been addressed in isolation, such as speech and prosody recognition and generation (Prevost & Steedman, 1994, Pierrehumbert & Hirschberg, 1990), knowledge representation, graphical puppet animation (Badler et al., 1993), robotics (Brooks & Stein, 1993), facial animation and representation (Pelachaud et al., 1996, Essa et al. 1994, Waters, 1990), facial expression recognition (Essa et al., 1994), speech acts (Searle, 1969) and discourse (Grice, 1988, Grosz & Sidner, 1986), gesture analysis and classification (Bers, 1995, Maes et al., 1995, Wexelblatt, 1994, Sparrell, 1993) and gesture generation (Cassell, 1994), as well as real-time coordination and process control (Hayes-Roth et al., 1988) and user modeling (Wahlster, 1991). A unifying problem calls for a unified approach.

The Ymir Architecture

Ymir is a broad, generative model of psychosocial dialogue skills that bridges between multimodal perception, decision and multimodal action in a coherent framework. It represents a distributed, modular approach that can be used to create autonomous characters capable of full-duplex[1] multimodal perception and action generation (Thórisson, 1998, 1996). Features from three A.I. approaches have been adopted in Ymir: *Blackboard systems* (Adler, 1992, Nii, 1989, Engelmore & Morgan, 1988, Selfridge, 1959), *Schema Theory* (Arbib, 1992) and *behavior-based* systems (Maes, 1990). However, Ymir goes beyond any one of these in the number of communication modalities and performance criteria it addresses. The goals behind the architecture, all of which have been addressed in the model, can be summarized as:

1. Multimodal input and output has to be supported, with no artificial communication protocols or rules.
2. All data types and data combinations found in human face-to-face communication should be accommodated (analog, spatial, symbolic).
3. Incremental, real-time interpretation of perceptual input has to be directly supported.
4. Incremental output generation has to be directly supported.
5. Real-time interpretation has to happen in parallel with real-time response generation, providing seamlessness in the interaction.
6. Time should be an explicit part of the architecture's structure and internal data.

Given the complexities of integrating numerous multimodal capabilities in a single system, two practical features that make the architecture more useful as a research tool are:

7. The architecture should allow incremental expansion of a character's abilities, and
8. The architecture should allow the possibility for testing various computational methods within each of its elements.

A character created in this architecture should have the following abilities:

9. it should be capable of fluid, dynamic, face-to-face dialogue with human users, and
10. a character should be able to fluently combine its *communication* and *task skills*, like humans do in task-oriented dialogue.

Two critical abilities that affect the way the whole system is designed, and allow us to meet the requirements in 9 & 10:

11. The character should be able to cancel any of its own actions at a moment's notice, based on a verbal or other perceptual cue.[2]
12. While some behaviors may be highly autonomous most of the time (for example where the next fixation falls), the agent should be able to follow verbal instructions to modify that behavior (e.g. "stop staring"), requiring control links from symbolic processes to lower-level behaviors.

Following Nii (1989) we can describe a computational system at three levels: The *model* is the least specific, showing the ideology behind the approach, the *framework* is more specific, detailing the pieces of the system and their interconnections, and the *specification* being the most detailed one, showing how to implement the particular system. This article focuses on the first two.

1 Full-duplex in this context means that the interaction is open-loop, i.e. the exchange of information is not step-lock.

2 Even in highly automated tasks performed by humans, such as touch typing, people can cancel motor sequences within 90 ms of a perceptual cue (Kosslyn & Koenig, 1992). This means that the fastest perception-action loop in our model should be no longer than 90 ms, quite a strict requirement for a complex system like multimodal dialogue.

The Dialogue Model

The model of multimodal interaction used here can be characterized as a layered feedback-loop[1] model, and is intended to be *descriptive*—it is based on descriptive results from the psychological and linguistic literature, as well as *generative*—it specifies how a conversant can construct dialogue in real-time. The three layers in the model are based on the time-scale and function of actions found in face-to-face dialogue (FIGURE 4). At each level various sensory and action processes are running. When engaged in dialogue the set of sensory and action processes at work in each layer are mostly determined by the role of the participant at each moment: *speaker* or *listener*. The organizing principle for these roles is *turn-taking* (Goodwin, 1981, Sacks et al., 1974)—turn-taking has been incorporated into the Ymir prototype through perceptual and decision mechanisms.

The Six Elements of Ymir

The six main types of elements in Ymir are:

1. **Perception:** A set of Unimodal Perceptors, ρ_υ, and Multimodal Integrators, ρ_i.
2. **Decision:** A set of Deciders, overt Π_o, and covert Π_c.
3. **Action:** A set of Behaviors, β, and Behavior Morphologies, βm (specific motor programs).
4. **Interprocess communication:** A set of Blackboards, Φ.
5. **Knowledge:** A Dialogue Knowledge Base, κ_{DKB}, and a set of Topic Knowledge Bases, $\{\kappa_{TKB\text{-}1} ... \kappa_{TKB\text{-}n}\}$.
6. **Organization:** Four semi-independent process collections, Γ: three perception-decision layers, and an Action Scheduler.

In the following discussion, the term *modules* refers to the elements of perception (ρ_υ, ρ_i), decision (Π_o, Π_c), behaviors (β) and knowledge (κ).

The three perception-decision layers group perception and decision objects together into time-dependent groups; the Action Scheduler uses a lexicon of Behavior objects to carry out motor-level actions in small increments (FIGURE 5). Multimodal sensory data can flow in to all layers (but not all data is relevant everywhere). Ymir can accommodate any number of modules—behavior, perception, decision and knowledge. Blackboards allow communication between the modules, both within process collections and between. By and large, all events in the model are non-deterministic, that is, the results of events are not guaranteed.

In the current implementation a graphics system receives output from the Action Scheduler and controls addressable relative-positions for a number of controlpoints that link to the character's anatomy (these could also be implemented for example as stepper motors or servo controllers). The Action Scheduler runs on its own computer, the rest of the Ymir elements on another (see Hardware, below). Other viable options are: Each module could have its own execution thread, each process collection could have its own execution thread, or the whole system could be sequential.

In the prototype, Unimodal Perceptors, Multimodal Integrators and Deciders are modelled as objects with methods. The main methods are **UPDATE**, **ACTIVATE**, **DEACTIVATE**, **FIRE** and **POST**. **UPDATE** is called on each module according to its desired update rate, determined by the layer to which the object belongs. **FIRE** happens when a module's conditions are met, at which time it **POST**s its message and state, to one of the blackboards (e.g. **[User-Speaking true <Timestamp>]**) — the timestamp

1 "Feedback" in this context refers to the reciprocal nature of any speaker-listener relationship, where a participant's [P_A] multimodal action [P_A-1] is met by the other's [P_B] re-action [P_B-1]. This feedback loop can be more than one level deep; a common format is the sequence [$P_A\text{-}1 \rightarrow P_B\text{-}1 \rightarrow P_A\text{-}2$]. See e.g. Clark (1992).

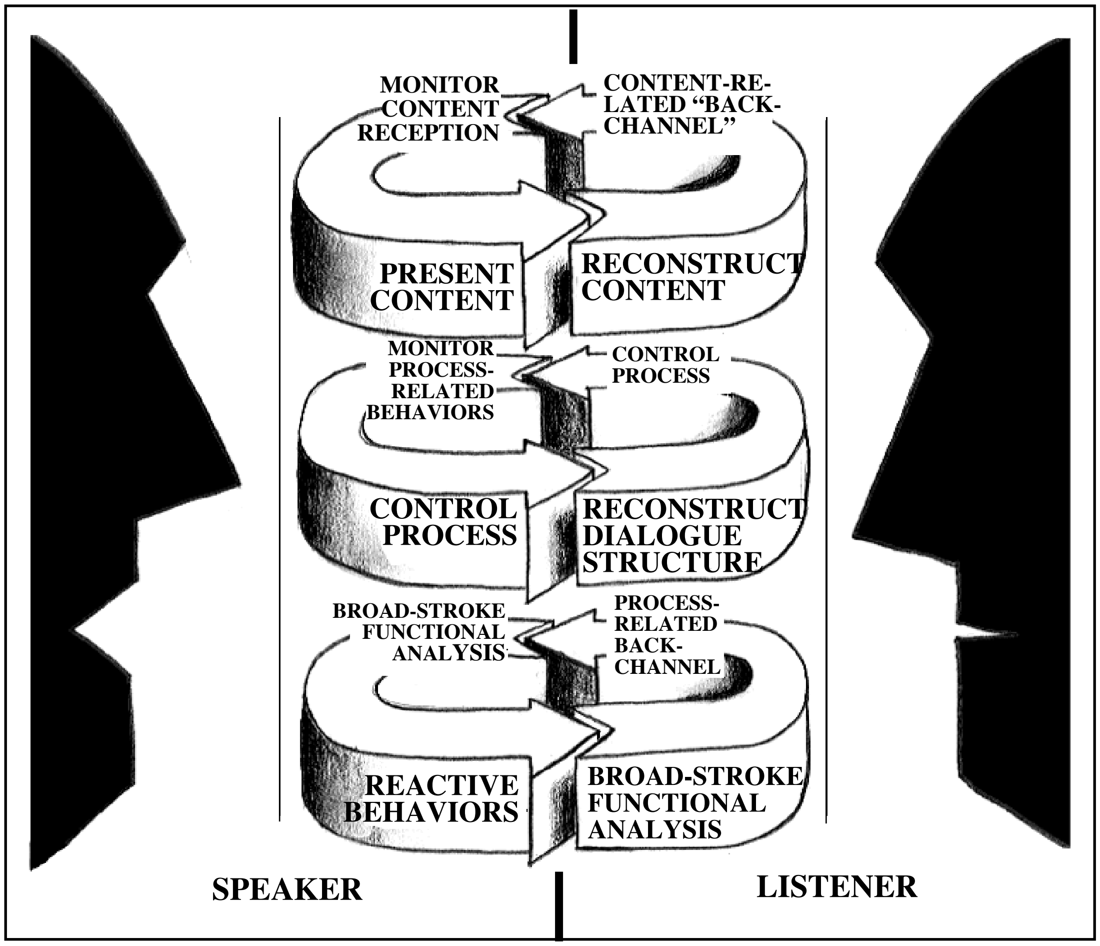

MONITOR CONTENT RECEPTION
CONTENT-RE-LATED "BACK-CHANNEL"
PRESENT CONTENT
RECONSTRUCT CONTENT
MONITOR PROCESS-RELATED BEHAVIORS
CONTROL PROCESS
CONTROL PROCESS
RECONSTRUCT DIALOGUE STRUCTURE
BROAD-STROKE FUNCTIONAL ANALYSIS
PROCESS-RELATED BACK-CHANNEL
REACTIVE BEHAVIORS
BROAD-STROKE FUNCTIONAL ANALYSIS
SPEAKER
LISTENER

FIGURE 4. The proposed three "channels", or layers, of information transfer in multimodal dialogue. At each layer semi-independent decisions are being made in real-time, arbitrated between and scheduled for execution by an independent action scheduler. In reference to FIGURE 3 and FIGURE 5, loop time for each half of a layer (i.e. for each bent arrow) in the lowest layer is in the range of 2-10 Hz, in the middle layer 1-2 Hz, and in the top layer 1 Hz and slower. Loop times are assumed symmetrical for speaker and listener. The roles of listener and speaker require different tasks to be performed in the perceptual, decision and motor processes. One of the primary process-related tasks of the of the participants is therefore to coordinate when the turns are given and taken, i.e. who is in the role of listener and who is the speaker.

allows other modules to use knowledge of real-time when determining whether and how to use this information. **ACTIVATE** and **DEACTIVATE** are used to turn perceptual and Deciders **on** and **off**, depending on the dialogue state and/or task-related actions at each moment in time.

We will now take a closer look at Ymir's elements in the following order: perceptual processing, decision making, the Behavior Lexicon and the Knowledge Bases. Then we will look at how these are grouped in the process collections, and how communication between process collections is handled through the blackboards.

Perceptual Processing

There are two types of modules in Ymir's perceptual system, Unimodal *Perceptors* (ρ_υ) and *Multimodal Integrators* (ρ_ι). *Unimodal Perceptors* encode features related to a single mode. These perceptors are considered to lie at least one level above the basic transformation provided by energy transducers, such as a retina or cochlea, or, in the case of the Gandalf system (see system description below), the user's body movement and posture,[1] and speech signal. A Unimodal Perceptor is associated with a single mode: an example of a Unimodal Perceptor would be a vocalization perceptor that turns **on** or **off** depending on whether sounds are emanating from the user's mouth, and a hand position perceptor that signals whether the speaker's hands are in or out of gesture space.[2]

The Unimodal Perceptors in the Ymir system fall into the following categories:

1. Prosodic
2. Speech
3. Positional
4. Directional

Prosodic perceptors track the intonation of the speech, pauses, volume of vocalization and related features (c.f. Todd & Brown, 1994); *speech* perceptors are a general class of processes that look at the speech *content*.[3] They could for example be sensitive to certain words that play a role in dialogue orchestration such as cue phrases (c.f. Litman, 1996). They would also track the global functional aspects of speech, such as determining whether an utterance is syntactically correct, whether it makes sense pragmatically, what the topic is, etc., as much as these can be done bottom-up using speech only. More extensive and context-sensitive analysis of these features is done elsewhere in the system, at a slower pace. *Positional* perceptors track the relative position of two or more objects, e.g. displacement of the eyebrows from a resting position or the location of a person with regard to oneself, and *directional* perceptors track the direction of objects (e.g. gaze, trunk or head).

Unimodal Perceptors have been implemented in the Gandalf prototype. They consist of the following parts: {1} the module itself, a CLOS[4] object, with {a} pointers to where to find the necessary data, {b} a pointer to the custom-made function which takes that data as arguments, {c} its own state (**true** or **false**), as well as {d} a time stamp for when the module last changed its state, and {2} a custom-made function that pre-digests the data needed for the perceptor. A module will for example take continuous intonation and digitize it into **up**, **down** and **no-change** time-stamped values. A single directional perceptor may be used to signal whether the speaker is looking at the agent or somewhere else. A position perceptor may track the position of hand relative to the user's body.

A Unimodal Perceptor detects important events in a single mode; however, if we want an artificial character to respond to multimodal events, such as "What is [pointing gesture] that!?", we need to look at the co-occurrence of two or more information channels. Processing the output provided by the Unimodal Perceptors is a collection of what are called *Multimodal Integrators* (ρ_i). The integrators aggregate information from the Unimodal Perceptors, as well as from other Multimodal Integrators, to come up with more comprehensive descriptions of the user's behavior. An example is an integrator that tries to determine whether the user is *giving the turn*, using data from speech, gesture, gaze and head direction. Another might combine information from a spatial perceptor and a speech perceptor to provide a **[Sound**

1 In the current implementation the user's body is represented geometrically. This is not a prerequisite for the architecture to work, it just happens to be the most readily available and handy representation available because of the space-sensing methods used (Bers, 1996).

2 Most of the perceptual modules and all of the Deciders in the Ymir prototype provide boolean output, which simplifies significantly construction of a large system. However, in many cases, e.g. for tracking the topic of the dialogue, such a simplification cannot be made. Boolean states of modules are by no means a requirement the architecture, simply one way of instantiating it.

3 See discussion on speech recognition in "Knowledge Bases", below.

4 Common Lisp Object System (Steele, 1990).

[type: Human-Voice], <Location>, <Timestamp>] report that can be used by the agent for orienting itself toward a person.

There are two kinds of Multimodal Integrators,

1. *static integrators* and
2. *dynamic integrators*.

Static integrators simply respond to a static situation, whereas *dynamic* integrators detect patterns over time intervals—as reported by other perceptual modules—such as a specific combination of arm and eye movements for a given interval. For instance, holding up your hand and uttering something ("ahhh") while the agent is talking may—in the context performed—constitute a wish to interrupt. This co-occurrence of actions in two modes could be detected by a static integrator sensitive to the co-presence of "hand in gesture space" and "vocalization" if the agent has the turn. A head nod (a dynamic event) and a particular vocalization ("aha") combines into a single "back channel feedback" report, detected with a single dynamic integrator. As mentioned before, perception modules communicate with each other, and to other modules, through the blackboards.

Deciders (Π)

Deciders make decisions about what to do from moment to moment by looking at the agent's up-to-the-millisecond knowledge state (found on the blackboards), which includes a representation of the outside world as well as the state of its own processing. Decisions made by overt modules (Π_o) affect the outward behavior of the agent; decisions of covert modules (Π_c) affect the processing inside the agent's mind. A special type of the covert Decider is used to keep track of states and state changes, e.g. dialogue state. This allows us deal with mutually exclusive states that the agent can take on, such as "listener" and "speaker".[1] Each Decider contains knowledge about where to look for data (which blackboards), what to do with it and how to communicate its status to other modules by posting information to the blackboards. One of the main functions of the covert Deciders is to turn on and off the right kinds of perceptual modules, depending on the dialogue state. Deciders can also turn other Deciders **on** and **off**.

An example of an overt Decider is shown in FIGURE 6. The conditions in **Fire-Conditions** have to be met for the module to turn **true**. When a Decider's conditions are met, it fires and sends out a *Behavior Request*; its **State** turns back to **false** (the **true** state is transient). Then it waits for the conditions in **Reset-Conditions** to be met in order to be able to send the Behavior Request again. The value in the slot **Expected-Lifetime** determines how long its Behavior Request, once sent, can wait without being executed. In the current implementation, a request that takes longer to process than its **Expected-Lifetime** specifies is simply cancelled without ever being executed. Reports on which Requests are (and which are not) executed are posted to the Motor Feedback Blackboard (see below).

The fate of the Behavior Requests is determined in the *Action Scheduler* (AS)—the agent's "cerebellum". We will look at the AS process collection below, but first we will examine the behaviors which are requested.

Behaviors (β): The Behavior Lexicon

If a Decider has sent out the Behavior Request **Smile**, how should this relatively high-level behavior be executed, given the state of the dialogue, motors in the face, and currently executing plans that might

1 Notice that this should be modeled as a decision task, not a perceptual task; changing your role from "listener" to "speaker" is a decision, not a perception.

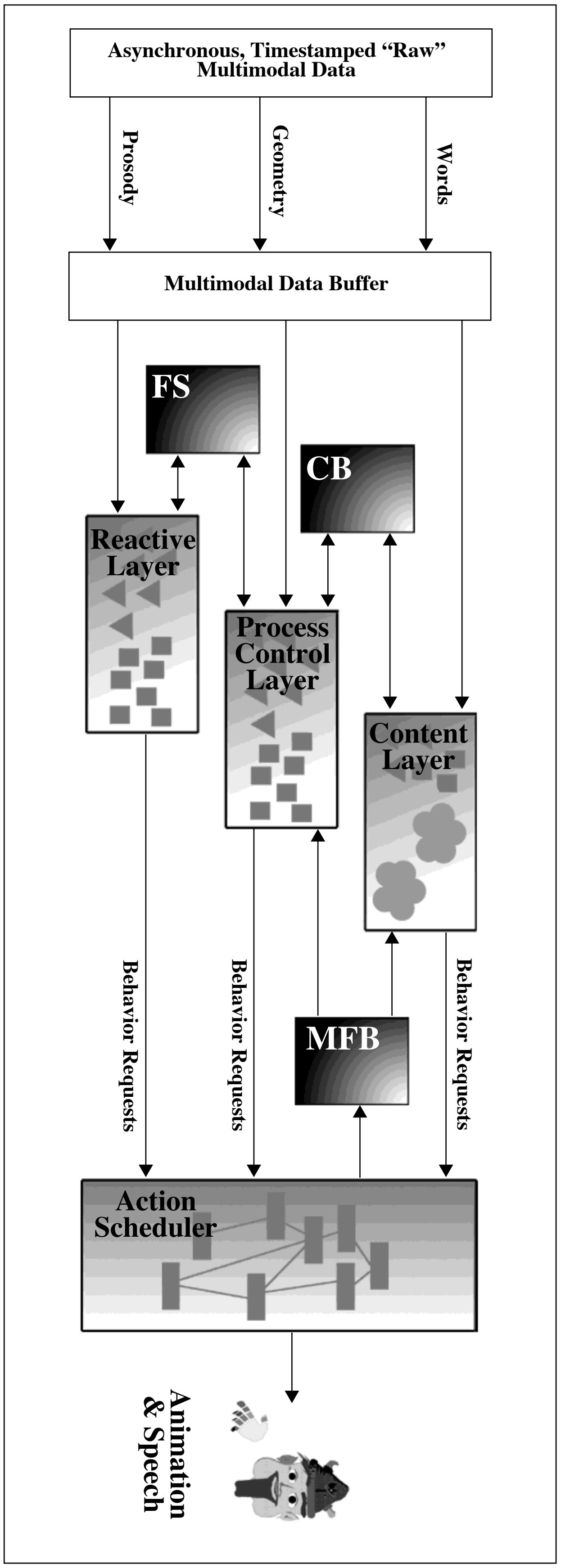


FIGURE 5. Data flow into and between the three layers (RL, PCL & CL), Action Scheduler (AS) and the three limited-access blackboards (FS, CB & MFB) of the Ymir architecture. The layers contain perception modules (prisms) and Deciders (squares), and correspond to the three layers of information transfer shown in FIGURE 4. Process-related back-channel feedback and low-level functional analysis happens in the RL; process control, reconstruction of dialogue structure and monitoring of process-related behaviors happens in the PCL; content-related back-channel, content reception monitoring, content presentation and content interpretation happens in the CL. Knowledge Bases in the CL are indicated by "blobs"; the Behavior Lexicon in the AS is shown as a network of nodes. For examples of data transmitted between layers, AS and blackboards see FIGURE 9.

```
DECISION-MODULE: Show-I-Know-User-Is-Addressing-Me
TYPE: overt-decision-module
EXPECTED-LIFETIME: 700 ms
BEHAVIOR-REQUEST: (Turn-To ‘user)
FIRE-CONDITIONS: ((TKB-exe-world-act false) (User-Looking-At-Me true)
                  (User-Facing-Me true) (User-Speaking true))
RESET-CONDITIONS: (User-Gives-Turn true)
STATE: false
TIMESTAMP: 2523423
```

FIGURE 6. Example of an overt Decider. The module makes a request to the motor system to turn the agent’s head toward the user (**Turn-To ‘user**). There is one request per module. The necessary pre-conditions for this module firing are that no Topic Knowledge Base is executing a task, meaning that the agent should not be currently executing a requested action in the topic domain (**TKB-exe-world-act false**), that the user is speaking (**User-Speaking true**), facing him (**User-Facing-Me true**) and looking at him (**User-Looking-At-Me true**). Once the module has fired it can not make its Behavior Request again until the user gives the agent the turn (**User-Gives-Turn true**), in which case the module’s **STATE** changes from **true** to **false**. This module belongs to the Process Control Layer (see below). In covert modules the **BEHAVIOR-REQUEST** slot is contains a function call to some internal task. Logic gates (AND, OR, etc.) can be used to combine pre-conditons for a more flexible and complex decision making.

overlap with it? Behavior Requests are an “intention” to perform a specific act; that act can be executed in many ways by selecting from a set of Behavior Morphologies available for that act.

Behavior Morphologies (βm) are chosen from a library of alternatives, called a *Behavior Lexicon*. The approach taken here to motor representation is in some ways similar to Rosenbaum et al. (1992): The idea of stored postures is used in the Behavior Lexicon, as is the idea of hierarchical storage of increasingly smaller units. This method for representing behavior leads to a database where functional and morphological definitions co-exist in the same space, with no distinct division lines between the two classes. An example of what such a database may look like is given in FIGURE 7. The behavior **Show-Im-Taking-Turn** is functionally indexed (according to its function: to show that the agent is taking the speaking turn), whereas **Turn-Head-Away-From-User** is indexed morphologically (according to the way it looks). A more obvious example of this distinction is the two behaviors **Show-I-Agree** (functional), and **Nod** (morphological), which in some cases can result in the same action, but often don’t. Other examples of behaviors in Gandalf’s Lexicon are **Look-At-User**, **Blink**, **Hesitate**, **Look-Puzzled** and **Greet**. All of these are relatively high-level descriptions of behaviors—branches in a tree of behaviors where the leafs are facial, hand or body motors. In a fully-realized Behavior Lexicon, most of these can be realized in multiple ways at the motor level.

Knowledge Bases

Knowledge Bases contain task knowledge and declarative knowledge about a particular topic. They contain information on how to do high-level interpretation of a person’s multimodal acts in the context of the specific topics they represent, how to generate responses to those acts, and how to communicate their processing status to other parts of the system. The Dialogue Knowledge Base (DKB) contains a central part of the system’s psychosocial knowledge: Any knowledge that has to do with dialogue—such as participants, their body parts, instruments used in interaction (mouths, hands, eyes, etc.), greetings, good-byes, etc.—rightfully belongs in the DKB and is considered a topic (albeit a meta-topic) in and of itself. Memory is distributed: History about the task, be it excavation or moon landings, is stored in the relevant Topic Knowledge Base (TKB); history about the interaction, and references to the interaction (“Go back to when I told you to ...”), are stored and treated in the DKB. The DKB also contains

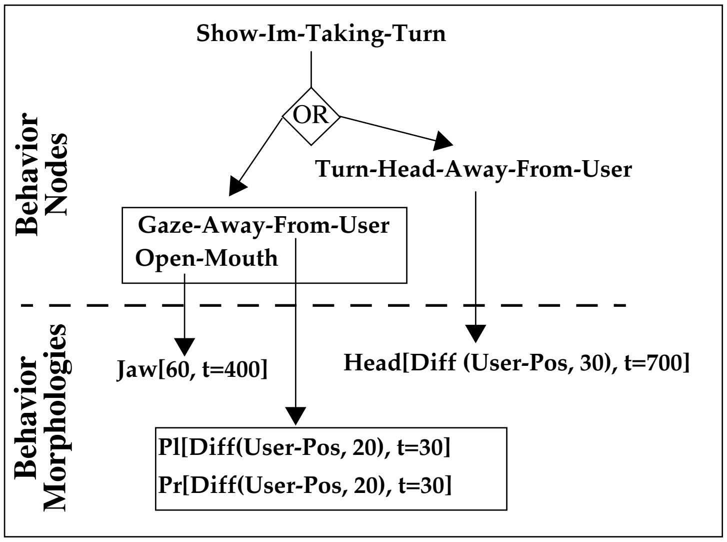


FIGURE 7. The behavior **Show-Taking-Turn** has two possible instantiations, **Turn-Head-Away-From-User** and the parallel pair [**Gaze-Away-From-User, Open-Mouth**] (parallel actions are in boxes). Each of these options point to low-level motor commands which specify angular degrees and time in milliseconds for particular muscle movements. The function **Diff** returns a setting that is guaranteed to not include a given variable (in this case the user's head) in the agent's line of sight. Access to a spatial knowlege base provides necessary data to execute behaviors containing wildcards such as the user's head (see FIGURE 8). In the current implementation, new execution time may be specified for any action—the default times for the motors are then overridden. Notice that **Show-Taking-Turn** is defined functionally, while the other behaviors are morphologically defined. (Head = agent's head, User-Pos = user's head position, P = pupil (left and right), Jaw = agent's jaw motor, t = time in milliseconds, other numbers represent angular degrees—and relative position along range of motion in the case of motor Jaw.)

knowledge about its knowledge, i.e. it has a record of all the TKBs available to the agent. This way references to discussions about more than one topic ("Remember when we were talking about Mars?") can be treated in one place, along with other meta-references to past dialogue.

Upon receiving a particular multimodal action as prepared by the perceptual processes, the DKB selects the most relevant TKB for interpreting that act. To take an example, if the agent knows the two topics of music and computer graphics, and hears the utterance "Turn the blue box sideways" the DKB recognizes that this utterance refers to the *computer graphics topic* and notifies the computer graphics TKB, which will in turn interpret the input in that context and compose some actions that can satisfy the request. If the topic of an utterance cannot be reliably predicted by simple methods, e.g. keyword indexing (c.f. Litman 1996) and expectations about topical continuity, the DKB may opt for sending it to multiple TKBs and compare the confidence score that they return with the parse. The win in this modularization of the knowledge is not that it reflects some deep feature of human mental functioning, but rather that it allows for the one-time creation of dialogue knowledge ("Look over here", "Listen to me"), with domain-dependent knowledge being added by the agent's designer (or by the system at run-time) in a plug-and-play fashion.[1]

Whether we use one, two, or many speech recognizers and/or recognition schemes in Ymir, they need to adhere to the same requirements as the perceptual processes, i.e. be *incremental* and post *intermediate*

1 I would like to thank Richard A. Bolt for pointing me to the issue of modularity here; see also Walker & Whittaker (1990).

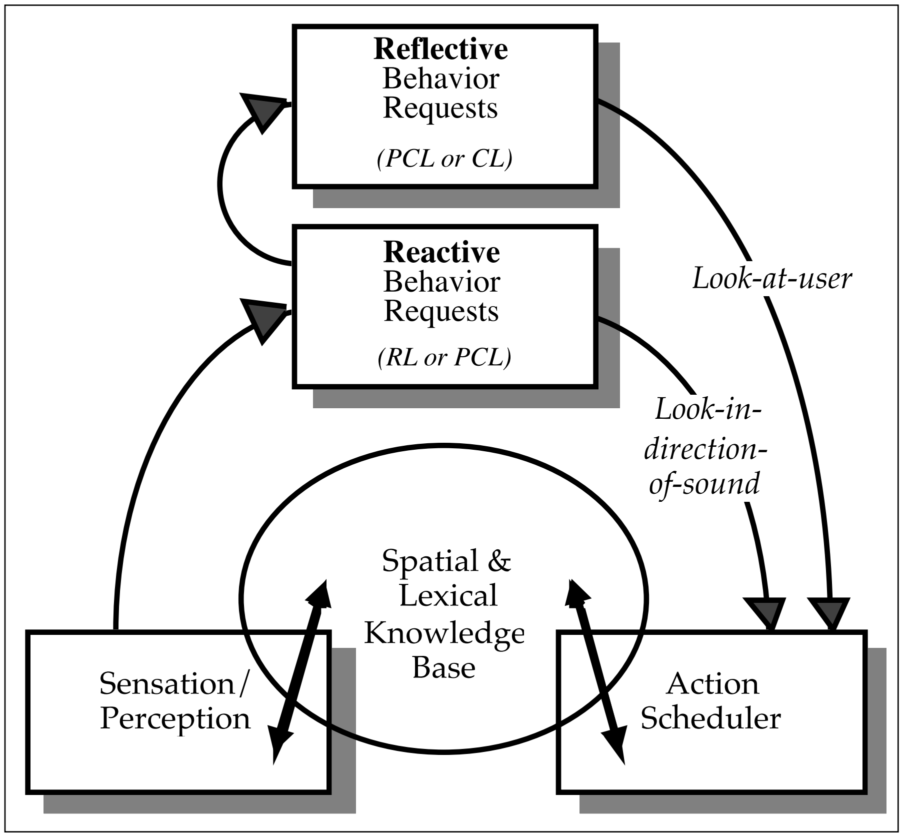


FIGURE 8. The Action Scheduler has access to a spatial knowledge base that is kept updated by the sensory and perceptual mechanisms. Examples of messages sent to the AS from the Reactive and Process Control layers are **Look-At-User** and **Look-in-Dir-of-Sound[X]**.

results for other processes to read and act on. These are used by the multimodal interpretation schemes that piece together speech, gesture, intonation and other prosody data incrementally, but also modules doing process control, which need to know the progress of the interpretation to be able to give feedback incrementally.

To allow an agent to move in relation to surrounding objects such as a person or a task area, the Action Scheduler needs access to a spatial knowledge base. A common spatial knowledge base, that is fed with information from the perceptual processes (FIGURE 8), allows the Action Scheduler to access spatial data needed for executing behaviors that require a reference point in the real world. Examples of such actions are **Look-at-User** and **Turn-to-Area-[X]**, where **[X]** is a variable attached to some external object or reference frame.

Summary: Perceptual, Decision, Behavior, & Knowledge Base Modules

Perceptual modules (ρ_υ, ρ_i) receive and prepare input data to be used as the basis for decisions to act, either covertly (internal actions, Π_c) or overtly (initiating visible external behaviors, Π_o). The Deciders can read mental and world states—provided by perceptual processes, as well as process reports from one or more Knowledge Bases (κ)—and issue behavior requests (βr) which are to be executed as a particular behavior morphology (motor program, βm).

We are now ready to discuss how these elements are grouped together, and how time is addressed, in the Process Collections.

Process Collections

Unlike some of the cognitive psychology models following the strictly pipelined, flat structure *perceive→decide→act*, the modules in Ymir are distributed into four process collections,

{1} a *Reactive* layer (RL),

{2} a *Process Control* layer (PCL), a

{3} *Content* layer (CL) and an

{4} *Action Scheduler* (AS).

The elements discussed in the previous sections are distributed in these collections in the following way:

$$\Gamma_{(RL)} = \{\rho u, \rho i, \Pi o\}$$

$$\Gamma_{(PCL)} = \{\rho u, \rho i, \Pi o, \Pi c\}$$

$$\Gamma_{(CL)} = \{\rho u, \rho i, \Pi o, \Pi c, \kappa\}$$

$$\Gamma_{(AS)} = \{\beta, \beta m\}$$

Processes in the RL are "simple" in that they do initial processing of sensory data and make reactive decisions based on this, e.g. looking away briefly before starting to speak (Goodwin 1981); these decisions and actions are on time scales Newell (1990) calls "deliberate act" (FIGURE 3). Processes in the PCL do more complex computations, but are mostly tied to controlling the process of the dialogue, e.g. deciding when to take turns speaking. They belong to Newell's time scale of "operations". Knowledge bases in the CL interpret input about a specific domain and produce actions that are applicable in response to the content of that input. These fall under Newell's "unit task" and "task" time scales.

The AS produces specific motor morphologies. Processes in each collection communicate with processes in other collections through blackboards. Top-down control is achieved through the covert Deciders: Deciders in the PCL and CL can control activity in the layers below, consisting mostly of turning perceptual modules on and off, and, sometimes, other Deciders. This control helps guide the lower-level processing and serves as a way to direct the attention of the character. It is always deterministic, it does not go through the blackboards.

The processes within each collection share a similar time-specificity. The collections are separated by different computational expense, primary data types and processing speeds. Using time stamping, processing delays and transmission delays are logged and treated like any other data in the system.

Reactive Layer (RL)

Processes in the Reactive Layer operate on relatively simple data. Here the Unimodal Perceptors give descriptions of hand movement (for instance whether either left or right hand are in gesture space), gaze direction (e.g. is the user looking at the agent or the workspace), prosody (e.g. significant pauses) and body posture (e.g. is the user turned away from the agent or not). Most of these modules are continuously active since their output is very basic and of interest to many high-level processes. Multimodal Integrators in the RL compute intermediate, multimodal "functional sketches" of the user's behavior. A functional sketch is computation that sketches out, in broad strokes, the dialogue *function* of an action. For example, when someone brings up an arm with the index finger extended and says "Look!", the *function* of that arm movement and utterance is to direct the listener's attention in a specific spatial direction. The perceptual modules in the RL try to describe this multimodal act in the most general way, by labeling its elements and then classifying it, first as a *communicative act* (based on collective evidence from many modes), then perhaps the arm movement as a *deictic* or other type of gesture (based e.g. on evidence from hand posture, hand position and gaze). Deciders in the Reactive Layer search for conditions in the Functional Sketchboard, for example to make a decision whether to look in the direction the speaker appears to be pointing. Other examples of actions initiated, or requested, from this layer are **Look-At-Person**, and **Show-Im-Taking-Turn** (often done through a very subtle movement of eyebrows, head and/or gaze; Goodwin, 1996). Reactive Deciders use data in the Functional Sketchboard to issue actions with a relatively high speed/accuracy trade-off; i.e. as long as the actions are quick it doesn't matter that they are less than 100% accurate—their correct *timing* is more important than their *exact type*.

Computation at this lowest level is assumed to be "immediate", i.e. without delay. This simplification can be made by using computing mechanisms that are significantly faster than the fastest response needed in human communication, ideally a fraction of 100 msec. A full cycle, from perception to action, through the RL is 2-10 Hz.

Process Control Layer (PCL)

The role of processes in the PCL is mostly to control global aspects of dialogue: to turn the correct kinds of internal processes on and off, recognize the global context of dialogue and manage communicative behavior of the agent. Perceptual modules in the PCL compute features that take on average 0.5 - 1 second to compute. Features and feature combinations related to dialogue state are computed here, as well as features of gestures and process control-related speech.

Deciders in the PCL deal with issues such as *when* a question should be answered; what to do when information is missing, when to greet; etc. Examples of the slightly more complex behaviors requested from the PCL are **Indicate-Response-Delay**, which might be requested if the agent's speech recognition takes longer to compute than is the social norm, and **Express-Confusion**, for instance if the TKB or DBK fails to produce pragmatically complete structures from the received input. Deciders in this layer also control covert actions related to the dialogue such as starting to listen, making predictions about the knowledge needed in a particular interaction; making predictions about what to expect next; managing multi-turn information exchange, etc. State-tracking Deciders related to short-lived tasks, such as turn-taking, belong to this layer.

Deciders in the PCL can control (turn on and off; change thresholds in) the processes in the Reactive Layer. This feature is essential since many of the lower level processes don't have global enough information to decide when they should be active and when not. The Process Control Layer's Deciders can look for conditions in both the Functional Sketchboard (data from the RL and PCL) and the Content Blackboard (data from the PCL and CL).

Typically, a full cycle (from perception to action) through the PCL is 1-2 Hz.

Content Layer (CL)

The role of the CL is to host the processes that make sense of the *content* of the input and generate acceptable responses based on this. For example, upon hearing the words "Delete that [pointing gesture] blue box", the appropriate Topic Knowledge Base locates the "blue box" by analyzing the user's gesture or gaze around the time she said the words, finds the I.D. of the blue box, finds the correct command to remove items, and applies that command to the I.D. of the object. In addition to having its own perceptual processes and Deciders, data processed at lower levels in the system, by the perceptual modules, trickles up to the CL to guide top-down analysis. For instance, if perceptual modules in the RL and PCL had labeled a particular segment of arm motion as "deictic", perceptual modules in the CL could analyze it further to find an exact direction of pointing. If the label from the lower layers was simply "communicative gesture" and the accompanying speech was "Tilt it", perceptual modules in the

CL could extract a direction of rotation from the gesture. This makes it possible to drive analysis not only from speech but also from any other relevant multimodal event, or from their combination.

As we have mentioned, dialogue knowledge is separated from topic knowledge. This has certain advantages, for example when interpreting dialogue-related gesture. We can put the recognition of gesture related to dialogue into the DKB without having to replicate these in each TKB. The gestures whose recognition and generation belongs rightfully in the DKB are {1} *emblematic gestures* related to the dialogue (e.g. holding up a hand with palm forward to signal "stop speaking"), {2} *deictic gestures* involving objects in the Dialogue Knowledge Base such as the user, {3} *beats* and {4} *butterworths* (e.g. waving a hand around while searching for a word or a phrase). Iconic, pantomimic and deictic gestures related to the topic of discussion cannot be interpreted without reference to knowledge of the topic. These should be recognized, interpreted and responded to in the appropriate TKB.

A full cycle time (from perception to action) through the CL typically starts at 1 second and goes up to virtual infinity (one can defer *content* processing through clever control of dialogue *process* by saying "I'll get back to you on that next year", or "next decade", or "next century").

The Action Scheduler

The AS's role is to receive Behavior Requests (βr) from the Deciders (FIGURE 9). The AS prioritizes the behavior requests, and chooses specific morphologies (βm) for them, thus determining how each requested action looks at the motor level. Since the choice of morphology is done at run-time, it can be based on any relevant information accessible to the AS, the most obvious being the current status of the motor system (in the case of Gandalf this includes a face, a voice and a hand). By divorcing the *decision to act* from its *form* in this way, the exact morphology can be fitted more to the current state of the agent's actions. This increases the system's reactivity while still allowing long-term plans to be executed.

As mentioned above, behaviors are indexed at multiple levels of abstraction in the Behavior Lexicon: from high (e.g. **Smile**), to medium (e.g. **Pull-Corners-of-Mouth-Up**), to low (e.g. **Move-Motor-x-to-Position-y**). Obviously, there must be a number of ways a person could smile, fewer ways in which one could pull both corners of mouth up, and only a handful of ways to move a muscle to a particular location. We can make a tree, where particular morphologies are given as the tree's leafs (FIGURE 7). As we travel up the tree, the flexibility for various implementations of a particular act, like **Smile**, or **Show-Im-Taking-Turn**, increases. Of course, with more options, it takes longer to choose the best one. Overt Deciders in the RL generally request highly specific actions; those in the PCL usually make a more general specification, and those in the CL compose their own multimodal action sequences, whose execution is still handled by the AS.

To choose between βm options, the AS uses an anytime algorithm (Dean, 1987). We need this kind of algorithm if we want the execution of multiple, possibly conflicting behaviors to be executed in the most efficient manner: Being able to execute a given behavior morphology *Right Now*, even if it may not be the best choice, makes the system much more responsive to its environment. Being sensitive to internal and external states is of course a necessary ingredient in making a character seem lively and intelligent.

To take an example of how scheduling in the AS works: if it receives a request for the behavior **Acknowledge**, it can use the current dialogue state and the amount of load on the various degrees of freedom of its own motor system to choose a way to express this (Thórisson, 1997). The usual method might be to say "yes", but if the user is speaking, perhaps a nod would be more appropriate; however, if the agent's head is already moving, it might choose to give verbal content feedback anyway.

There are many ways to implement such a scheduling algorithm; the key requirements are that it be time- and context-sensitive. In the current prototype the Action Scheduler is sensitive to the current state of the "motors" of the body; given two or more choices it chooses motor programs that don't cause conflicts with currently executing behaviors. A more extensive context-sensitivity could include an ability to negotiate with a planner for the best action solution given the current goal. Time-sensitivity is achieved by {1} using time-outs for each decision (Expected-Lifetime, FIGURE 6) and {2} prioritization based on which layer requests the behavior: behavior requests from the RL take highest priority, CL-initiated behaviors take lowest priority, with PCL-initiated behaviors in the middle.

Summary of Process Collections

The purpose of the layers is to provide a hierarchy for {1} the complexity and incremental nature of perception and interpretation, {2} decision making prioritization and {3} top-down control and bottom-up analysis. The RL contains fast processes that describe the environment in broad strokes and make decisions about behaviors on a short time scale (e.g. blinking, gaze). Processes in the PCL do more sophisticated analysis of input (partly based on analysis from the RL), and initiate task-level actions taking 1-2 seconds to execute. The CL contains knowledge bases which produce plans, multimodal actions and content-related responses in the dialogue. By allowing Deciders in each layer to turn on and off modules in the lower layers, the system allows characters to follow a verbal command like "stop staring", controlling low-level, semi-automatic processes from a higher level. It also allows the architecture to address a wide range of time scales in real-time action, a necessity if we want to create communicative humanoids (a 10 minute face-to-face dialogue contains actions whose different execution times span four orders of magnitude (FIGURE 3)).

Behavior Requests issued by Deciders in the RL are generally specified at a lower level than those in the PCL, since these are under tighter time constraints, with the result that the AS doesn't have to spend valuable time composing or selecting a set of motor commands for the action involved, but simply looks up the default motor specification and sends it to the animation module. Behavior Requests issued by the RL take the highest priority, followed by the PCL and then the CL. The process collections are semi-independent in that they can all execute in parallel; each process collection has it's own target updating frequency.

The Action Scheduler separates the execution of motor actions from their decision, and thus in combination with Deciders in the RL allows a character to stop any action it is executing, based on a perceptual cue, at a moment's notice. The process collections provide a structure to deal explicitly with real-time constraints.

Blackboards

What is the best way for the processes in each process collection to talk to each other? Gandalf contains roughly 150 perception, decision and behavior modules, and we can only expect this number to grow for more complete agents. If we were to "string wires" between all the modules that need to talk to each other, this would not only be tedious but could be difficult to change later. Changing connections when adding new modules, or modifying the connections on the fly during run-time would also be difficult.

The obvious choice here is to use blackboards (FIGURE 5) (Selfridge, 1959). There are three main blackboards in Ymir. The first one is for information exchange primarily between the Reactive Layer and the Process Control Layer. This blackboard is called the *Functional Sketchboard*. It stores intermediate and final results of low-level (high-speed) perceptual processes such as motion, whether the agent hears something or not, and first-pass multimodal descriptions. It also keeps a record of all decisions that have been initiated from the RL. In the prototype, most messages take the form **[<Message>, <State>, <Timestamp>]**, where **State** is either **true** or **false**.

The second blackboard is the *Content Blackboard*, servicing communication between the Process Control Layer and the Content Layer. This blackboard is the key to the separation of process control and content analysis in Ymir. Here results are posted that are less time-critical than those on the Functional Sketchboard. Examples of messages between the CL and the PCL are also shown in FIGURE 9.

An important feature of the AS is its ability to execute long chains of actions incrementally. To simulate human dialogue skills, behaviors that span long stretches of time need to be interruptible. They also need to be composed incrementally to allow relevant, unpredicted events to influence actions in a natural t ↓

USER's SPEECH	FUNCTIONAL SKETCHBOARD	CONTENT BLACKBOARD	ACTION SCHEDULER
	(RHAND-IN-GEST-SPACE T 5140)		(TURN-TO 2 68790 1000 'USER 5100)
	(HAND-IN-GEST-SPACE T 5150)		
	(R-DEICTIC-MORPH T 5180)		
	(SPEAKING T 5180)		
	(USER-TAKING-TURN T 5190)		
	(I-TAKE-TURN NIL 5330)		
	(I-GIVE-TURN T 5340)		
		(SPCH-DATA-AVAIL NIL 5540)	
	(SPEAKING NIL 5650)		(GAZE-AT-ZERO-ZERO 1 5430 200 NIL 5570)
	(USER-TAKING-TURN NIL 5660)		(SHOW-I-GIVE-TURN 1 5430 200 NIL 5570)
	(REFERENCING-DOMAIN T 5670)		(LOOK-AT 2 5520 1000 'USER 5580)
	(IS-ACTIVE NIL 5670)		
	(COMPLETE-UTTER NIL 5670)		(LOOK-AT 2 5700 2000 'DOMAIN 5820)
	(LOOKING-AT-ME NIL 6080)		(TURN-TO 2 5500 1000 'USER 6030)
		(RCV-SPCH T 6220)	
	(I-TAKE-TURN T 6440)	(RCV-SPCH NIL 6230)	
	(I-GIVE-TURN NIL 6450)		
		(RCV-SPCH T 6630)	
(IS 5130 6640)			
(THAT 5260 6650)			
(JUPITER 5670 6660)			
		(RCV-SPCH NIL 6670)	
		(SPCH-DATA-AVAIL T 6670)	(SHOW-I-TAKE-TURN 1 6510 500 NIL 6690)
	(R-DEICTIC-MORPH NIL 6760)		
	(RHAND-IN-GEST-SPACE NIL 6810)	(KB-SUCC-PARSE T 6980)	
	(HAND-IN-GEST-SPACE NIL 6820)	(TKB-ACT-AVAIL T 6990)	
	(COMPLETE-PRAGM T 7080)		(BLINK 1 6900 1000 NIL 7010)
	(COMPLETE-GRAM T 7080)		(HESITATE 2 6900 200 NIL 7020)
	(COMPLETE-SYNT T 7090)		
		(CL-ACT-AVAIL T 7210)	(TURN-TO 2 6802 2000 'PLANET 7230)
		(TKB-ACT-AVAIL NIL 7420)	
		(KB-EXE-ACT T 7440)	
		(TKB-EXE-SPEECH-ACT T 7520)	(DELIVER-SPEECH 3 6990 9000 "No, that is SATURN" 7520)
		(CL-ACT-AVAIL NIL 7730)	
			(TURN-TO 2 7660 2000 'USER 7750)
		(TKB-EXE-SPEECH-ACT NIL 10070)	
		(KB-EXE-ACT NIL 10070)	(BLINK 1 10000 1000 NIL 10080)
	(FACING-DOMAIN T 10140)		
			(TURN-TO 2 10240 2000 'DOMAIN 10300)
	(FACING-ME NIL 10360)		
			(RESTLESS 2 10460 2000 NIL 10510)
	(TURNED-TO-ME NIL 10570)		
	(SPEAKING T 10780)		

FIGURE 9. (page 20) An actual trace from the Functional Sketchboard, Content Blackboard and behavior requests sent to the Action Scheduler (AS) over a period of approximately 4 seconds. On the far left the user's words are also shown, with timestamps (when uttered and when received, respectively). Time is in milliseconds. Message format, FS & CB: (Decision/Perception, State, Timestamp); Behavior Requests received by the AS: (Behavior, Initiating-Layer, Time-when-behavior-was-requested, Expected-Lifetime-in-ms, Data, Time-when-AS-started-to-execution-of-behavior); Initiating-Layer: 1 = RL, 2 =PCL, 3 = CL; T = true, NIL = false.

The sensory module **Is-Active** is used to help determine when the user is giving turn. The sensory module **Referencing-Domain** posts true whenever it looks like the user is looking or pointing at the work area—in this case the large screen showing the planets.

way. This can only be accomplished if we keep track of which part, in a stream of actions, is currently being planned and/or executed by the AS. The AS does this through the third blackboard in Ymir, the Motor Feedback Blackboard (MFB). The PCL and CL can read the MFB for status of formerly initiated behaviors and replace those that are cancelled or have failed for some reason. The MFB also allows the KBs to refine actions already started, and make incremental additions to these as they get executed by the AS. The perception-act cycle in the RL is so short that this internal feedback is not useful; for motor acts initiated by modules in the RL the real-world is the only feedback loop. The blackboards need update rates that allow multiple, asynchronous modules read-write access without blocking.

Gandalf: Humanoid One

A character can be built in Ymir by {1} specifying perceptual, decision and behavior modules, {2} the specific procedures needed for data and internal computations, and {3} grouping these into the correct process collections. To give an example of character implementation in Ymir, this section describes the first prototype character Gandalf (FIGURE 10) and relates his abilities to the constructs described in the preceding sections.

Gandalf's Dialogue Abilities

Gandalf is an expert in the solar system and can tell users facts about the planets. He can also travel to the planets, zoom in and out, and start and stop the planets' moons in their orbits. Gandalf has been given a minimal set of modules necessary for face-to-face interaction. His modules were designed by data-mining the psychological literature (c.f. McNeill, 1992, Rimé & Schiaratura, 1991, Clark & Brennan 1990, Pierrehumbert & Hirschberg, 1990, Whittaker & Stenton, 1988, Goodwin, 1986, Grosz & Sidner, 1986, Kleinke, 1986, Nespolous & Lecours, 1986, Goodwin, 1981, Kahneman 1973, Duncan, 1972, Yngve 1970, Ekman & Friesen 1969, Searle, 1969). This work, which included reformulating human-subject experimental results in terms of Ymir's elements and process collections, produced a total of 16 Unimodal Perceptors, 10 Multimodal Integrators and 35 Deciders. The Behavior Lexicon contains 83 behavior nodes (63 leaf nodes).

To determine which layer a new module should belong to when designing an agent in Ymir, one can use time-specificity (i.e. decisions that have expected life-span under 500 ms are most likely to belong in the RL), and/or the time the action takes to execute (e.g. RL if less than 500 ms). Another criteria is the kind of perceptual data needed for decisions to be made reliably; those requiring complex data are less likely to belong to the RL. More specific guidelines for determining which layer a particular behavior belongs to (i.e. what are its real-time requirements), and how to design these, remains to be developed (cf. Bryson & McGonigle, 1998).

Gandalf has proven to be capable of fluid turn-taking and dynamic dialogue sequencing (Thórisson, 1996), and integrates topic and dialogue knowledge seamlessly. He is capable of producing real-time multimodal behaviors of several important kinds:

Multimodal motor output:

- Hands: **Deictic gesture** (pointing to objects of discussion), **emblematic gesture** — e.g. holding the hand up, palm forward, signalling "hold it" to interrupt when the user is speaking, and **beat gestures** — hand motion synchronized with speech production.
- Face: **Emotional emblems** — smiling, looking puzzled, **communicative signals** — e.g. raising eyebrows when greeting.
- Eyes: **Attentional** and **deictic functions**, both during speaking and listening.
- Body: **Emblematic body language** — nodding, shaking head.
- Speech: **Back channel feedback** and **meaningful utterances**.
- **Turn-taking signals**: E.g. looking quickly away and back when taking turn, attentional cues such as head and gaze direction (gaze deictics), greeting (in various ways), etc.

These are all coordinated in real-time and correctly inserted into the dialogue in the right context. Their execution is based on the *perception and interpretation of the following kinds of* **user behavior** *data,* and their relation to the *inferred dialogue state*.

Multimodal Sensory input:

- Hands: **Deictic gesture** — pointing at objects, and **iconic gesture** — when a user tells Gandalf to tilt an object she can show the intended direction of tilt with the hand.
- Eyes: **Attentional** and **deictic functions**, both during speaking and listening.
- Speech: **Prosody** — the timing and sequence of speech-related sounds and intonation, and **speech content** — in the form of word tokens from a speech recognizer.

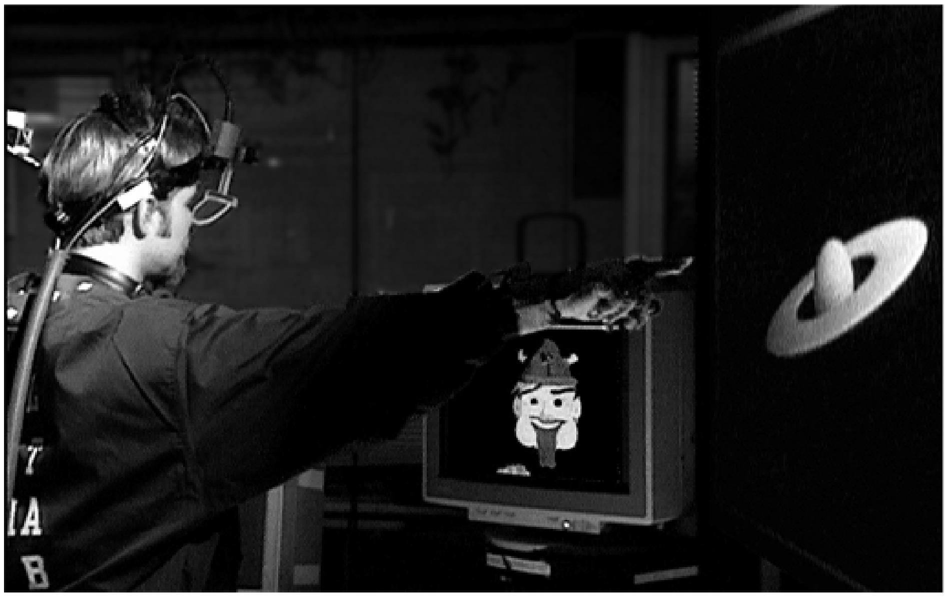


FIGURE 10. The author asks Gandalf "What planet is that?" Gandalf appears to users on its own monitor. A model of the solar system appears on a large screen in front of the user.

- Body: **Direction of head** and **trunk** — e.g. when user turns away from Gandalf to talk to someone else, and **position of hands in body space** (hand position relative to trunk and head).
- **Turn-taking signals**: Various feature-based analysis of combinations of co-occurring multimodal events, such as intonation, gaze, hand position and head direction.

Natural Language Processing & Knowledge Bases

Language processing in Gandalf takes two paths. A grammar-based speech recognizer gives Gandalf speaker-independent, continuous speech recognition, with a vocabulary of around 100 words, and a prosody analyzer classifies and timestamps speech pauses and intonation. The latter has primarily been used (along with other multimodal data, processed mostly in the RL) to help Gandalf predict turn transitions. In the future it could be used for finding *given* and *new* information in the speech stream (Prevost, 1996).

An (admittedly skimpy) dialogue knowledge base and a topic knowledge base allow Gandalf to orchestrate the interaction, travel to planets in the solar system (manipulating a graphic database) and tell facts about the planets at the user's request. Templates are used to parse multimodal actions and to compose multimodal responses. Speech acts with deictic references can be realized in many ways. For instance, if Gandalf fails to look in the direction that the user points when asking a question like "What planet is that?", thus failing to indicate to the user which object he thinks the gesture refers to, he may subsequently use a deictic manual or gaze gesture in his response, to indicate which planet he thinks the user pointed at ("That [deictic gesture] is Saturn"). Whether such a reference is made via manual gesture or gaze is not predetermined—the Action Scheduler, slaved to a real-time clock, will make run-time trade-offs between behavior morphologies immediately before execution, taking into consideration processing load and network delays of the whole system. This has the additional benefit of making Gandalf's behavior "naturally" non-repetitive, because internal processing may affect visible behaviors in subtle ways.

Verbal utterances are composed with a template-based mechanism using facts about the solar system. To resolve ellipsis, Gandalf keeps a state of which planet was last the focus. If a reference is made to an object that hasn't been mentioned before, he will assume that it is the planet that is most clearly visible on the screen at the moment. For requests like "Take me to Jupiter", positions and names of planets are accessed in a spatial knowledge base and animated paths are generated through the graphics database to "fly" to the right place.

Hardware

FIGURE 11 gives an overview of the hardware used for Gandalf. To capture the user's multimodal actions (speech, prosody, gaze & body movements) the prototype uses a microphone (Gandalf's "ear"), an eye tracker and a body suit (Gandalf's "eyes"; Bers, 1996). A face and hand, a total of 23 degrees-of-freedom, allow Gandalf to display his behaviors (Thórisson, 1997, 1996). His voice is provided by a commercial speech synthesizer. Processing is done on 8 networked computers, distributed in a way that reflects significant breaks in bandwidth requirements. Computational requirements could be reduced somewhat through optimization. The speed of the face/hand graphics presented one of the bottlenecks (Thórisson, 1997), limiting the frame-rate to 150 ms, about 50 ms slower than desired for precise human motor simulation. Another bottleneck was the speech recognition, leaving much to be desired with a full 2 seconds delay from the end of the utterance until words are available. However, this long delay is made less noticeable by Gandalf's time- and context-sensitive eyebrow motion and gaze shifts. To eradicate the natural language bottleneck we could start by adding incremental speech processing, cue-word spotting and more sophisticated real-time prosody analysis.

The eight computers used to run Gandalf's software were:

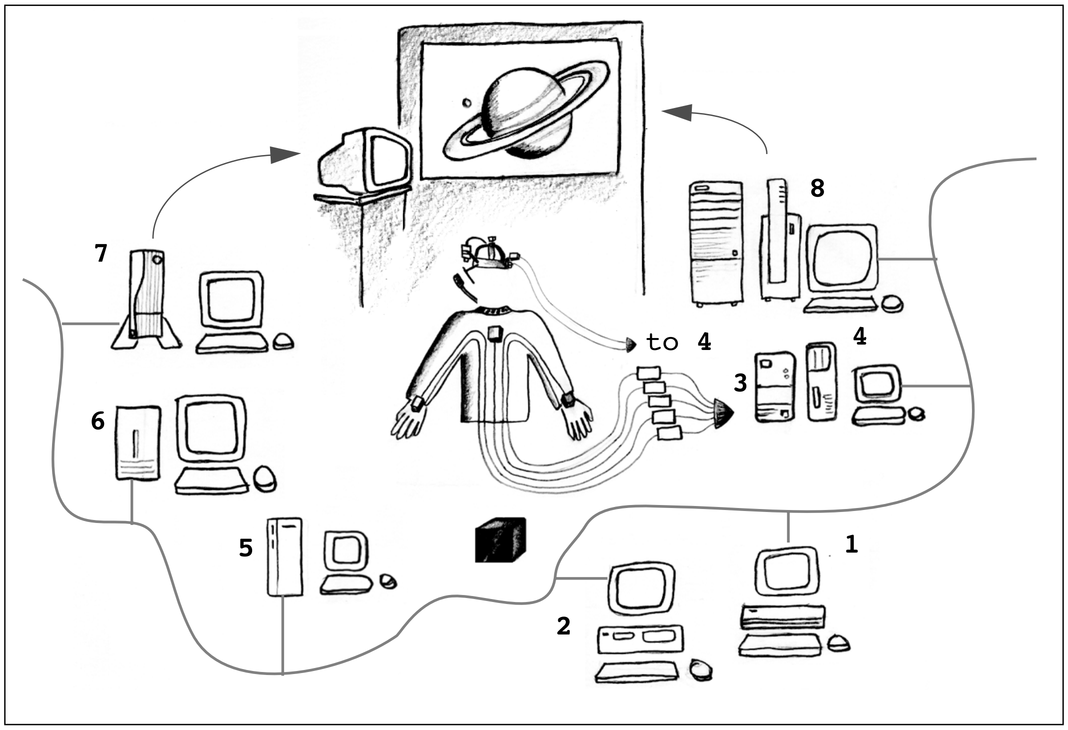


FIGURE 11. Gandalf system layout. Grey arrows show display connections; grey line is Ethernet. Eye tracker (video and magnetic sensor outputs) is connected to computer 4 via a serial port and the data is fed to computer 3 via another serial connection; output from the three magnetic sensors on the jakcet is connected to computer 3 via serial connections (dark arrowheads). Microphone signal is split to two computers (5 & 6) via an audio mixer (connections not shown). A DECTalk [1985] speech synthesizer is connected to computer 7 through a serial port (not shown). Black cube (connected to computer 3) generates the magnetic field needed for sensing the posture of user's upper body.

1. Most of Ymir, perceptual processes and Deciders in the Reactive Layer, Process Control Layer, Content Layer, as well as the Knowledge Bases, runs on a Digital Equipment Corporation Alpha 3000/300 150-MHz workstation.
2. The Action Scheduler runs on a Digital Equipment Corporation 5000/240 40 MHz workstation.
3. A 486 Intel processor PC is used to collect data from a body tracking suit.
4. A 386 Intel processor PC is used to collect data from an eye tracker.
5. A 33-MHz Macintosh Quadra 950 runs prosody analysis software using a Roland CP-40 pitch-to-MIDI (MIDI, 1983) converter to track the formant of the user's intonation.
6. A 100-MHz Silicon Graphics Iris is used for speech recognition.
7. A 100-MHz Silicon Graphics Indigo2 is used to animate the character and output speech, through a text-to-speech system (DECtalk, 1985).
8. A Hewlett-Packard Apollo 9000/750 66-MHz animates a virtual solar system.

FIGURE 12. (page 26) Graph showing internal states of Gandalf during interaction over a 16 second interval (each vertical line marks a second). Perceptual modules characterizing the *user's* behavior are listed in lines 1-17, 21-23, 28, 29 & 32-34; covert State Deciders are listed in lines 18, 19, 30 & 31; lines 24-27 & 35-45 are messages posted on the Content Blackboard; lines 46-58 are all the words that the user has spoken so far in the interaction (beginning of horizontal lines marks time when words were actually uttered, end shows when the words were delivered by the speech recognizer). Tick marks (lines 46-59) mark the time at which a decision to perform an over behavior was made (or, alternatively, the time at which a Behavior Request was sent to the Action Scheduler). See text for details.

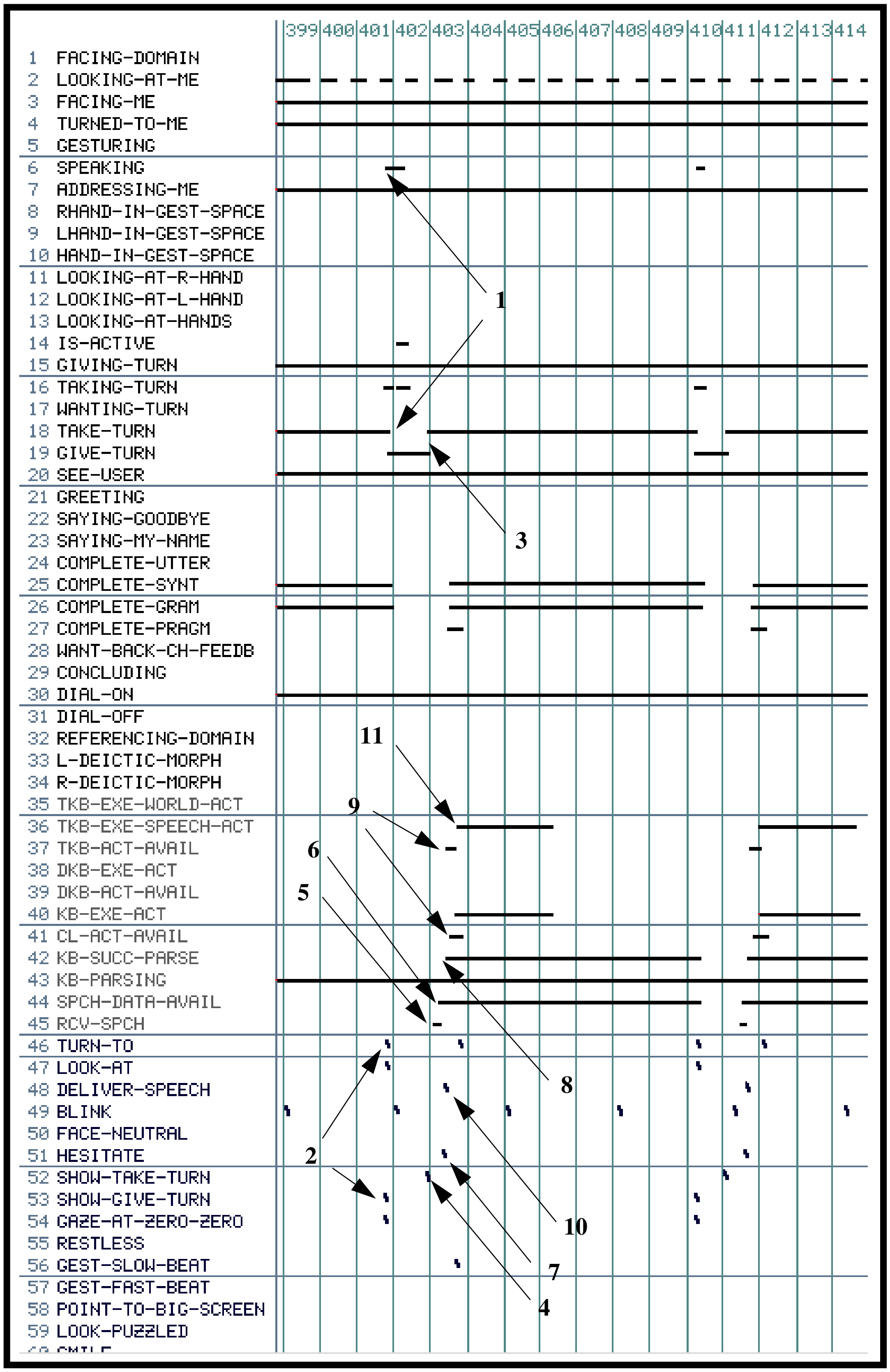
Performance Examples

FIGURE 12 gives a glimpse of some of the internal events in Gandalf during his interaction with a user. Two verbal requests made in this example by the user: "Tell me about Mars" (at second 401) and "Tell me more" (at second 410). When the user starts speaking a prosody perceptor called **Speaking** posts **true** (**Speaking**, line 6) and Gandalf gives turn [1], turns to the user and shows that he is giving turn [2]. (Internal events leading up to this include a Multimodal Integrator called **Taking-Turn** posting **true** because the user's gaze and head directions falls on Gandalf's face at the same time she starts speaking; a Decider called **Give-Turn** posting **true** because of this, and another Decider called **Show-Im-Giving-Turn** posting **true** because the **Give-Turn** module fired.) When the person falls silent (at second 402) and looks back at Gandalf (**Looking-at-Me**) Gandalf takes the turn [3], and shows that he is doing so [4]. At about the same time [5] something is received from the speech recognizer (a Unimodal Perceptor called **Rcv-Spch** posts **true** when the speech recognizer outputs something) and shortly thereafter [6] it is reported as available words uttered by the user (a Unimodal Perceptor called **Spch-Data-Avail** posts **true** when speech recognizer delivers actual words). These are then parsed (when **Spch-Data-Avail** is **true**, and a few other conditions are met, a covert Decider called **Parse-Spch** fires, and thus starts the parsing process).[1] The parse is reported as a successful parse [8] by the TKB; meanwhile Gandalf hesitates because he has taken the turn but has nothing to say yet [7] (in the Action Scheduler the behavior **Hesitate** can be realized in two or three ways, e.g. saying "ahhh" or looking to the side). When a multimodal response is available [9] (the message **[TKB-act-avail true]** is posted on the Content Blackboard by the solar-system Knowledge Base), Gandalf starts delivering this response [10] and the event is posted internally on the Content Blackboard [11].

To consider an alternative sequence of events we can look at what might have happened if the speech recognizer had not successfully recognized the spoken words (perhaps because of popping sounds in the microphone or noise in the background). Gandalf would still have known that the user had spoken because sensory modules analyzing prosody would have posted speech activity, and the modules (user-) **Taking-Turn** and (user-) **Giving-Turn** had posted **true** in sequence. So when subsequently **Rcv-Spch** posts **true** (because the speech recognizer sends out an empty message or error message) and **Spch-Data-Avail** fails to post **true** a certain period after that, the Decider **Deliver-Faulty-Reception-Report** would have fired a covert behavior which includes a **Look-At-User** action, a **Look-Puzzled** facial expression and the verbal report "Sorry, I did not get that."

FIGURE 13 shows another example of internal events during interaction with a user. This example spans 23 seconds, during which time the user makes three different requests, "Take me to the Sun", "Take me to Jupiter" and "What planet is that?" Notice that although everything seems to have worked correctly internally in the first request, Gandalf does not execute any action (there is no line drawn for **TKB-exe-world-act** after the request). This is because the Sun was already on the screen at the time of the request, and instead of executing the action, the TKB produces the utterance "This is Sun, dude". The

1 The reason for KB-Parsing posting continuously true here is related to the particular implementation of the parse process. It was later changed to make it's actual state available to the rest of the system.


399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
1 FACING-DOMAIN
2 LOOKING-AT-ME
3 FACING-ME
4 TURNED-TO-ME
5 GESTURING
6 SPEAKING
7 ADDRESSING-ME
8 RHAND-IN-GEST-SPACE
9 LHAND-IN-GEST-SPACE
10 HAND-IN-GEST-SPACE
11 LOOKING-AT-R-HAND
12 LOOKING-AT-L-HAND
13 LOOKING-AT-HANDS
14 IS-ACTIVE
15 GIVING-TURN
16 TAKING-TURN
17 WANTING-TURN
18 TAKE-TURN
19 GIVE-TURN
20 SEE-USER
21 GREETING
22 SAYING-GOODBYE
23 SAYING-MY-NAME
24 COMPLETE-UTTER
25 COMPLETE-SYNT
26 COMPLETE-GRAM
27 COMPLETE-PRAGM
28 WANT-BACK-CH-FEEDB
29 CONCLUDING
30 DIAL-ON
31 DIAL-OFF
32 REFERENCING-DOMAIN
33 L-DEICTIC-MORPH
34 R-DEICTIC-MORPH
35 TKB-EXE-WORLD-ACT
36 TKB-EXE-SPEECH-ACT
37 TKB-ACT-AVAIL
38 DKB-EXE-ACT
39 DKB-ACT-AVAIL
40 KB-EXE-ACT
41 CL-ACT-AVAIL
42 KB-SUCC-PARSE
43 KB-PARSING
44 SPCH-DATA-AVAIL
45 RCV-SPCH
46 TURN-TO
47 LOOK-AT
48 DELIVER-SPEECH
49 BLINK
50 FACE-NEUTRAL
51 HESITATE
52 SHOW-TAKE-TURN
53 SHOW-GIVE-TURN
54 GAZE-AT-ZERO-ZERO
55 RESTLESS
56 GEST-SLOW-BEAT
57 GEST-FAST-BEAT
58 POINT-TO-BIG-SCREEN
59 LOOK-PUZZLED
1
2
3
4
5
6
7
8
9
10
11

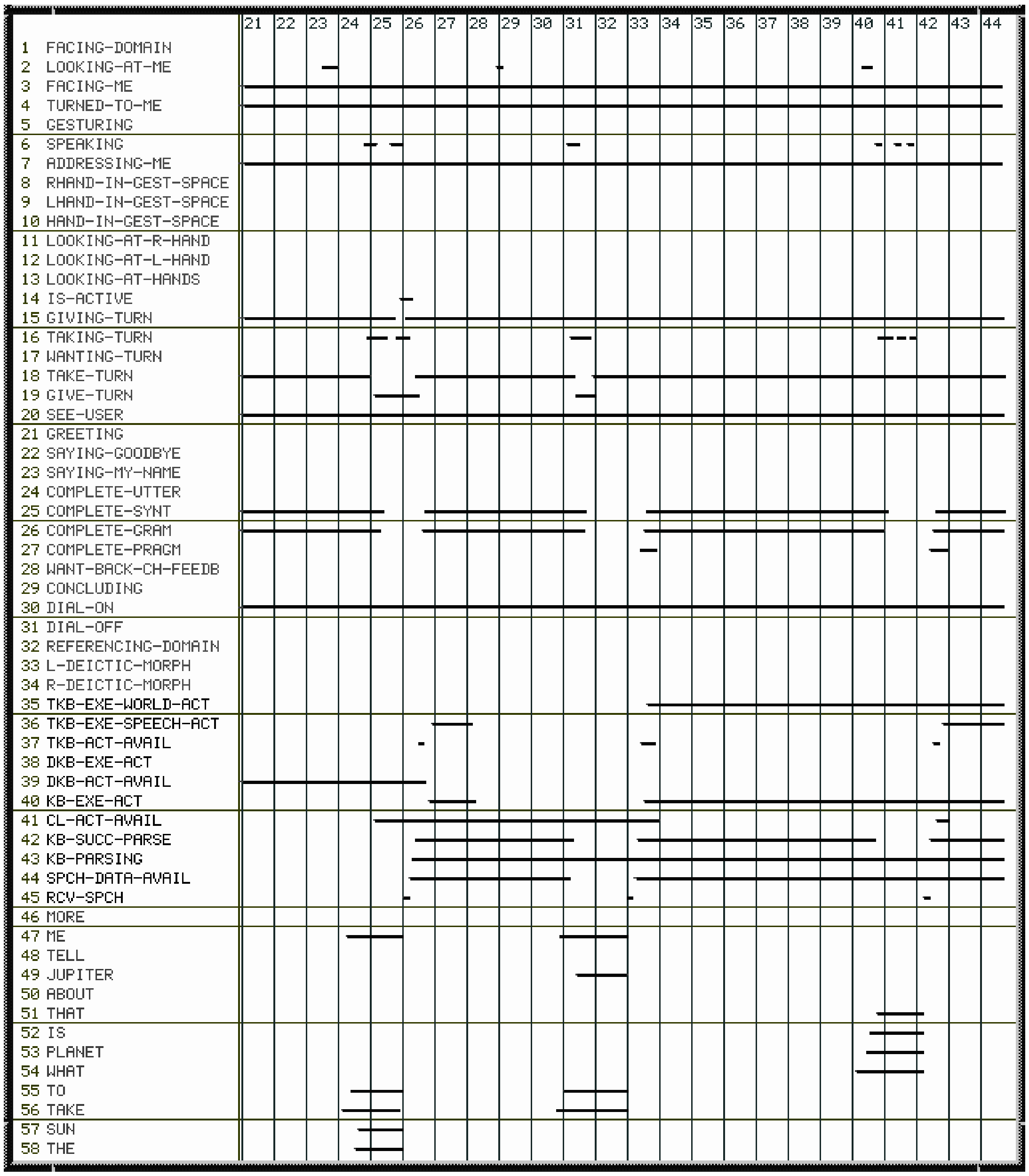


FIGURE 13. Example of internal events during an interaction betwen a user and Gandalf. Covert state Deciders are listed in lines 18, 19, 30 & 31; overt Deciders are listed from 46-59. Words spoken by the user are shown in lines 46 through 58 (beginning of line marks the time the speech recognizer estimates that a word was uttered; end of line maks the time when Gandalf received the word.) Around second 31 the user asks Gandalf to “take him to Jupiter”. This trip then starts at second 33 (**TKB-exe-world-act**) and continues through second 44. At second 40, before Gandalf has finished travelling to Jupiter, the user asks him a question, which Gandalf successfully parses (**KB-Succ-Parse**), generates a speech act response to (**TKB-act-avail**, **CL-act-avail**), and answers (**TKB-exe-speech-act**, second 42). The two sensory modules **L-Deictic-Morph** and **R-Deictic-Morph** post true when the hand of the user makes a fist with the index finger extended—the hand morphology associated with pointing in our society. This information, along with where the hand is positioned (e.g. in gesture space) and where the user is looking, allows Gandalf to predict when the user is performing a deictic gesture—which in turn can help the speech recognition to select the right grammar when parsing the speech.

duration of most of the messages in the Content Blackboard (lines 35-45) are determined by the duration between user turns. Module 24 was not active during this interaction.

Sequences such as the ones depicted include many more conditions and events than those mentioned here. However, these are illustrative (and real) examples of how the system works in action.

The Missing Pieces

Several features of multimodal dialogue remain unimplemented in the Ymir/Gandalf prototype. The main missing elements in the *input* are {1} recognizing the remaining categories of manual gesture, symbolic, pantomimic and metaphoric (McNeill, 1992, Rimé & Schiaratura, 1991), {2} facial expression recognition and {3} more extensive prosody analysis. The knowledge base separation assumes that {4} each knowledge base could use a separate speech recognizer, with individual vocabulary and grammar, or an equivalent method for getting n-best parses without compromising recognition speed and reliability. There are strong reasons to believe that Ymir can handle these four additions, since they do not introduce any kind of information flow or data types or mechanisms that have not already been addressed in the current prototype. Obviously, {5} natural language parsing can be taken much further in this prototype without serious research efforts. However, {6} speaker-independent, free-form speech in noisy environments may take a few years to get fully integrated into a situated, real-world agent (c.f. Brown & Cooke, 1994). An extension on the equipment side that would be nice is replacing the body tracking with cameras (cf. Maes et al., 1995). The main limiting factors of using state of the art computer vision with cameras is low reliability and speed, which is still too slow to support fine-grain, multimodal dialogue.

An important missing feature of the *output* is {1} a more extensive intonation generation (Prevost 1996, Prevost & Steedman, 1994). The current prototype uses the default intonation of the speech synthesizer—something that obviously could be improved by making use of the dialogue structure, which is already known to some extent. Gandalf's {2} knowledge, however, can be expanded in obvious ways; knowledge representation schemes abound. Another remaining task is to test further the idea of {3} multiple knowledge bases. Allowing Gandalf to perform {4} sensory-motor feedback-loops, such as hand-eye coordination and smooth pursuit eye movements, remains to be explored within the current framework. We may need to add new elements to address these, but they should be possible within the basic model.

Some of the most talked-about features of artificial creatures are motivation, emotion, learning and metabolism. To fully mimic human (and animal) behavior, these are of course needed. How this is best done in a modular, systems-level architecture like Ymir, while keeping the other features intact, is an interesting question. Since the focus here is on communication skills, there is less need for incorporating these than improving natural language, social rules and human-specific perception. However, we are relatively optimistic about Ymir's usefulness to explore these issues in a holistic way.

Summary & Conclusion

The Ymir architecture described in this paper allows us to create characters capable of the real-time orchestration of seamless, multimodal input and output in situated, natural-language based dialogue. The model underspecifies characteristics of topic knowledge but makes instead some specifications for the interaction protocol between the administrative tasks of multimodal dialogue on the one hand and topic knowledge on the other. Action and perception in the system offer various degrees of "reactiveness", from very reflex-like to highly "intelligent". The number of features that task-oriented dialogue shares with other skills required of autonomous agents, including real-time performance, task knowledge, and action selection at multiple levels of granularity, makes it likely that the architecture can be used in a broad range of applications.

Current work on the architecture focuses on adding the missing features (described above) of the Gandalf prototype, as well as trying different computational methods for achieving the functions of the various modules in the system. It will be interesting to see how much further the system can be pushed by for example using artificial neural network methods in the perceptual modules, fuzzy logic in the Deciders and more extensive scheduling mechanisms in the Action Scheduler. Larger knowledge bases will allow us to determine how the interface between the CL and the PCL needs to be extended for smarter agents; increased dialogue knowledge would enable humanoids to interact with more than one person at a time. Because of Ymir's modular structure, these extensions are not expected to impact the simplicity or performance of the system. The model as described here is not fixed; it has enough flexibility to meet various demands on the implementation and substructures of its parts. The possibilities of using Ymir as a general framework for many types of communicative, task-knowledgeable agents seem quite promising.

Acknowledgments. The author would like to thank the sponsors of this work: the M.I.T. Media Laboratory, Thomson-CSF, TSG Magic and HUMANOID Inc. Thanks to Justine Cassell, Pattie Maes, Steve Whittaker, Tom Malone & Richard A. Bolt for guidance and inspiration; Joshua Bers for developing the body model; Christopher Wren, Steven Levis, Calvin Yuen, Nimrod Warshawsky, Roland Paul & Hannes Vilhjálmsson for writing various parts of the animation systems; David Berger for network and programming advice. Thanks to NeCSys for keeping signals between the machines flowing. Thanks to Joanna Bryson, Mikkel Arentoft, Paul McKevitt and my anonymous reviewers for useful comments on this paper.

References

Adler, R. 1989. Blackboard Systems. In The Encyclopedia of Artificial Intelligence, 2nd edition, ed. S. C. Shapiro, pp. 110-116. New York, NY: Wiley Interscience.

Agre, P. & Chapman, D. 1987. Pengi: An Implementation of a Theory of Activity. Proceedings, 6th AAAI: 268-72.

Albus, J, McCain, H & Lumia, R. 1987. NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM), Technical Report Technical Note 1235, National Bureau of Standards, Gaithersburg, Maryland.

Arbib, M. A. 1992. Schema Theory. In The Encyclopedia of Artificial Intelligence, 2nd edition, ed. S. C. Shapiro, 1427-1443. New York, N.Y.: Wiley Interscience.

Badler, N. I., Phillips, C. B. & Webber, B. L. 1993. Simulating Humans: Computer Grahphics Animation and Control. New York, New York: Oxford University Press.

Bates, J., Loyall, A. & Reilly, W. S. 1994. An Architecture for Action, Emotion, and Social Behavior. Artificial Social Systems: Fourth European Workshop on Modelling Autonomous Agents in a Multi-Agent World. Springer-Verlag.

Bers, J. 1996. A Body Model Server for Human Motion Capture and Representation. Presence: Teleoperators and Virtual Environments, 5(3), 381-392.

Bers, J. 1995. Directing Animated Creatures through Gesture and Speech. Unpublished Master's Thesis, Media Arts and Sciences, Massachusetts Institute of Technology, U.S.A.

Blumberg, B. M. 1996. Old Tricks, New Dogs: Ethology and Interactive Creatures. Unpublished Ph.D. thesis, the Media Laboratory, Massachusetts Institute of Technology, U.S.A.

Blumberg, B. M. & Galyean, T. A. 1995. Multi-Level Direction of Autonomous Creatures for Real-Time Virtual Environments. Proceedings of SIGGRAPH '95, August, 47-54.

Bolt, R. A. 1987. The Integrated Multi-Modal Interface. The Transactions of the Institute of Electronics, Information and Communication Engineers (Japan), November, J79-D(11), 2017-2025.

Brooks, R. & Stein, L. A. 1993. Building Brains for Bodies. M.I.T. Artifical Intelligence Laboratory memo No. 1439, August.

Brown, G. J. & Cooke, M. 1994. Computational Auditory Scene Analysis. Computer Speech and Language, 8, 297-336.

Bryson, J. & McGonigle, B. 1998. Agent Architecture as Object Oriented Design. The Fourth International Workshop on Agent Theories, Architectures and Languages (ATAL '97), M. P. Singh, A. S. Rao & M. J. Wooldridge (eds.). New York, NY: Spring-Verlag.

Cassell, J. & Thórisson, K. R. 1998. The Power of a Nod and a Glance: Envelope vs. Emotional Feedback in Animated Conversational Agents. This volume.

Cassell, J., Stone, M., Douville, B., Prevost, S., Achorn, B., Steedman, M., Badler, N. & Pelachaud, C. 1994b. Modeling the Interaction between Speech and Gesture. Sixteenth Annual Conference of the Cognitive Science Society, Atlanta, Georgia, August 13-16, 153-158.

Chin, D. N. 1991. Intelligent Interfaces as Agents. In J. W. Sullivan & S. W. Tyler (eds.), Intelligent User Interfaces, 177-206. New York, NY: Addison-Wesley Publishing Company.

Clark, H. H. 1992. Arenas of Language Use. Chicago, Illinois: University of Chicago Press.

Clark, H. H. & Brennan, S. E. (1990). Grounding in Communication. In L. B. Resnick, J. Levine & S. D. Bahrend (eds.), Perspectives on Socially Shared Cognition, 127-149. American Psychological Association.

Dean, T. L. (1987). Intractability and Time-Dependent Planning. Proceedings of the 1986 Workshop on Reasoning About Actions and Plans, M. P. Georgeff & A. L. Lansky (eds.), Los Altos, California: Morgan Kaufman.

DECtalk (1985). DECtalk and DTC03 text-to-speech system owner's manual. Bedford, MA: Digital Equipment Corporation.

Duncan, S. Jr. 1972. Some Signals and Rules for Taking Speaking Turns in Conversations. Journal of Personality and Social Psychology, 23(2), 283-292.

Ekman, P. & Friesen, W. 1978. Facial Action Coding System. Palo Alto, CA: Consulting Psychologists Press.

Ekman, P. & Friesen, W. (1969). The Repertoire of Non-Verbal Behavior: Categories, Origins, Usage, and Coding. *Semiotica*, 1, 49-98.

Engelmore, R. & Morgan, T. (eds.) 1988. Blackboard Systems. Wokingham, England: Addisson-Wesley.

Essa, I. A., Darrell, T. & Pentland, A. 1994. Modeling and Interactive Animation of Facial Expression using Vision. M.I.T. Media Laboratory Perceptual Computing Section Technical Report No. 256.

Fehling, M. R., Altman, A. M. & Wilber, B. M. 1989. The Heuristic Control Virtual Machine: An Implementation of the Schemer Computational Model of Reflective, Real-Time Problem-Solving. In Jagannathan, R. Dodhiawala & L. S. Buam, Blackboard Architectures and Applications, 191-218. Boston: Academic Press, Inc.

Goodwin, C. 1986. Gestures as a Resource for the Organization of Mutual Orientation. Semiotica, 62(1/2), 29-49.

Goodwin, C. 1981. Conversational Organization: Interaction Between Speakers and Hearers. New York, NY: Academic Press.

Grice, H. P. 1989. Studies in the Way of Words. Cambridge, Massachusetts: Harvard University Press.

Grosz, B. J. & Sidner, C. L. 1986. Attention, Intentions, and the Strucutre of Discourse. Computational Linguistics, 12(3), 175-204.

Hayes-Roth, B., Hayes-Roth, F., Rosenschein, S. & Cammarata, S. 1988. Modeling Planning as an Incremental, Opportunistic Process. In R. Engelmore & T. Morgan, Blackboard Systems, 231-245. Reading, MA: Addison-Wesley Publishing Co.

Kahneman, D. (1973). Attention and Effort. New Jersey: Prentice-Hall, Inc.

Kleinke, C. 1986. Gaze and Eye Contact: A Research Review. Psychological Bulletin, 100(1), 78-100.

Kosslyn, S. M. & Koenig, O. 1992. Wet Mind: The New Cognitive Neuroscience. New York: The Free Press.

Laird, J. E. & Rosenbloom, P. S. (1995). The Evolution of the Soar Cognitive Architecture. In D. Steier & T. Mitchell (eds.) Mind Matters: Contributions to Cognitive and Computer Science in Honor of Allen Newell. Hillsdale, New Jersey: Lawrence Earlbaum Associates.

Laurel, B. 1990. Interface agents: Metaphors with character. In B. Laurel (ed.) The Art of Human-Computer Interface Design, 355-365. Reading, MA: Addison-Wesley Publishing Co.

Litman, D. J. (1996). Cue Phrase Classification Using Machine Learning. Journal of Artificial Intelligence Research 5, 53-94.

Maes, P., Darrell, T., Blumberg, B. & Pentland, A. 1995. The ALIVE System: Full-body Interaction with Autonomous Agents. IEEE Computer, Special Issue on Virtual Environments, 11-18.

Maes, P. 1990. Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back. In P. Maes (ed.), Designing Autonomous Agents, 1-2. Cambridge, MA: MIT Press.

Maes, P. 1989. How to Do the Right Thing. A.I. Memo No. 1180, December, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

McNeill, D. 1992. Hand and Mind: What Gestures Reveal about Thought. Chicago, IL: University of Chicago Press.

MIDI (1983). MIDI: Musical Instrument Digital Interface Specification 1.0. North Hollywood, California: International MIDI Association.

Nespolous, J-L & Lecours, A. R. (1986). Gestures: Nature and Function. In J-L Nespolous, P. Perron & A. R. Lecours (eds.), The Biological Foundations of Gestures: Motor and Semiotic Aspects, 49-62. Hillsdale, NJ: Lawrence Earlbaum Associates.

Newell, A. 1990. Unified Theories of Cognition. Cambridge, MA: Harvard University Press.

Nii, P. 1989. Blackboard Systems. In A. Barr, P. R. Cohen & E. A. Feigenbaum (eds.), The Handbook of Artificial Intelligence, Vol. IV, 1-74. Reading, MA: Addison-Wesley Publishing Co.

Pelachaud, C., Badler, N. I. & Steedman, M. 1996. Generating Facial Expressions for Speech. Cognitive Science, 20 (1), 1-46.

Pierrehumbert, J. & Hirschberg, J. 1990. The Meaning of Intonational Contours in the Interpretation of Discourse. In P. R. Cohen, J. Morgan & M. E. Pollack (eds.), Intentions in Communication. Cambridge: MIT Press.

Prevost, S. 1996. A Semantics of Contrast and Information Structure for Specifying Intonation in Spoken Language Generation. Ph.D. Thesis, Faculty of Coputer and Information Science, University of Pennsylvania.

Prevost, S. & Steedman, M. 1994. Specifying Intonation from Context for Speech Synthesis. Speech Communication, 15, 139-153.

Poesio, M. & Traum, D. R. 1995. Miscommunication in Multi-modal Collaboration. Proceedings of the IJCAI Workshop on Context in Natural Language Processing, 103-111, August.

Rimé, B. & Schiaratura, L. 1991. Gesture and Speech. In R. S. Feldman & B. Rimé, Fundamentals of Nonverbal Behavior, 239-281. New York: Press Syndicate of the University of Cambridge.

Rosenbaum, D. A. & Kirst, H. 1992. Antecedents of Action. In H. Heuer & S. W. Keele (eds.), Handbook of Motor Skills. New York, NY: Academic Press.

Searle, J. R. 1969. Speech Acts: An Essay in the Philosophy of Language. Cambridge: Cambridge University Press.

Sacks, H., Schegloff, E. A.. & Jefferson, G. A. 1974. A Simplest Systematics for the Organization of Turn-Taking in Conversation. Language, 50, 696-735.

Selfridge, O. 1959. Pandemonium: A Paradigm for Learning. Proceedings of Symposium on the Mechanization of Thought Processes, 511-529.

Sparrell, C. J. 1993. Coverbal Iconic Gesture in Human-Computer Interaction. Master's Thesis, Massachusetts Institute of Technology. Cambridge, Massachusetts.

Steele, G. L. 1990. Common Lisp the Language, second ed. Cambridge, Massachusetts: Digital Press.

Thorpe, C. E. 1992. Robots, Mobile. In S.C. Shapiro (ed.), The Encyclopedia of Artificial Intelligence, 2nd ed., 1409-1416. New York, New York: Wiley Interscience.

Thórisson, K. R. 1998. Decision Making in Real-Time Face-to-Face Multimodal Communication. Second ACM International Conference on Autonomous Agents '98, Minneapolis, Minnesota, May 12-15.

Thórisson, K. R. 1997. Layered Action Control in Communicative Humanoids. Proceedings of Computer Graphics Europe '97, June 5-7, Genieva, 134-143.

Thórisson, K. R. 1996. Communicative Humanoids: A Computational Model of Psychosocial Dialogue Skills. Ph.D. Thesis, Massachusetts Institute of Technology, U.S.A.

Thórisson, K. R. 1995. Computational Characteristics of Multimodal Dialouge. AAAI Fall Symposium Series on Embodied Lanugage and Action, November 10-12, Massachusetts Institute of Technology, Cambridge, 102-108.

Thórisson, K. R. 1994. Face-to-Face Communication with Computer Agents. AAAI Spring Symposium on Believable Agents Working Notes, Stanford University, California, March 19-20, 86-90.

Todd, N. P. M. & Brown, G. J. (1994). A Computational Model of Prosody Perception. Proceedings of the International Conference on Spoke Dialogue Processing, Yokohama, Japan, September 18-22, 127-130.

Traum, D. & Dillenbourg, P., 1996. Miscommunication in Multi-modal Collaboration. In working notes of the AAAI Workshop on Detecting, Repairing, And Preventing Human--Machine Miscommunication, August, 37-46.

Wahlster, W. 1991. User and Discourse Models for Multimodal Communication. In J. W. Sullivan & S. W. Tyler (eds.), Intelligent User Interfaces, 45-67. New York, New York: ACM Press, Addison-Wesley Publishing Company.

Walker, M. & Whittaker, S. 1990. Mixed Initiative in Dialogue: An Investigation into Discourse Segmentation. Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics.

Waters, K. 1990. Modeling 3D Facial Expressions. ACM SIGGRAPH '90 Course Notes, State in the Art in Facial Animation, 108-129.

Wexelblatt, A. 1994. A Feature-Based Approach to Continuous-Gestures Analysis. Unpublished Master's Thesis, Media Arts and Sciences, Massachusetts Institute of Technology, U.S.A.

Whittaker, S. & Stenton, P. (1988). Cues and Control in Expert-Client Dialogues. Proc. 26th Annual Meeting of the Association of Computational Linguistics, 123-130.

Wilson, S. W. 1991. The Animat Path to AI. In From Animals to Animats, eds.J-A. Meyer & S. W. Wilson,. Cambridge, MA: MIT Press.

Yngve, V. H. (1970). On Getting a Word in Edgewise. Papers from the Sixth Regional Meeting, Chicago Linguistics Society, 567-78.