Real-Time Decision Making in Multimodal Face-to-Face Communication

Kristinn R. Thórisson

Gesture & Narrative Language Group[*]
The Media Laboratory
Massachusetts Institute of Technology
20 Ames Street, Cambridge, Massachusetts 01239
kris@media.mit.edu http://www.media.mit.edu/~kris

1. ABSTRACT

This paper describes an architecture and mechanism for simulating real-time decision making as observed in full-duplex, multimodal face-to-face interaction between humans. The work bridges between multimodal perception and multimodal action generation and allows flexible implementation of multimodal, full-duplex, conversational characters. It is part of a broad computational model of psychosocial dialogue skills called *Ymir*. The architecture has been tested with a prototype humanoid, *Gandalf* [34][35]. Gandalf can engage in task-oriented dialogue with a person and has been shown capable of fluid turn-taking and multimodal interaction [40]. The primary focus in this paper is the real-time decision-making (action selection) mechanism of Ymir and its relation to the multimodal perception and motor control systems.

1.1 Keywords

Real-time decision making, agent architectures, multimodal, face-to-face communication

2. INTRODUCTION

The work described in this paper is motivated by the idea of communicative, autonomous agents capable of fluid, dynamic face-to-face interaction. The interest is not merely natural language—and surely there are numerous projects limited just to this—but rather a multimodal system, duplicating face-to-face dialogue between two or more communicating humans. Real-time multimodal embodied dialogue is an interesting problem because it requires the integration of areas that have existed in isolation. If we want synthetic characters to comprehend and generate gesture [22][27], body movements [15], facial gestures [11], back channel feedback [15][42], speaking turns [15][30], etc., along with natural language, and do this in real-time interaction with people, we need nothing short of a unified approach that comprehensively incorporates the full spectrum of behaviors exhibited by people in such situations. To do this requires input from artificial intelligence, cognitive science and psychology, as well as robotics and computer graphics.

Three elements of decision making in face-to-face dialogue are discussed in this paper: {1} Organization and coordination of semi-independent decisions that happen at different time scales, such as back-channel feedback (e.g."mhm" and nodding), and higher-level decisions including those related to natural language use (such as complex communicative acts), {2} decision mechanism representation, and {3} real-time scheduling of motor actions resulting from decisions to act. Other aspects of Ymir have been discussed elsewhere: Perception of multimodal events in [35]; real-time motor control in [33] and [35]. A quick overview of the Gandalf prototype is given here, but can also be found in [34] & [35].

3. MULTIMODAL DIALOGUE

The following characteristics of embodied multimodal dialogue are of particular interest to the issue of real-time decision making:[1]

1. *Multi-layered Input Analysis and Output Generation*. In discourse, actions in one mode may overlap those of another in time, they may have different timing requirements, and may constitute different information [22][15]. In order for purposeful conversation to work, reactive and reflective[2] responses have to co-exist to enable adequate behavior of an agent.

2. *Temporal Constraints*. The structure of dialogue requires that participants agree on a common speed of exchange [15]. Certain responses are expected to happen within a given time span, such as looking in a direction being pointed in. If these rules are violated, e.g. the direction of gaze changes 5 seconds after it was *expected* to change,[3] the action's meaning may be drastically altered in the con-

[*]Now at LEGO A/S, Kløvermarken 120, 7190 Billund, Denmark

PUBLISHED IN THE
Second ACM International Conference on Autonomous Agents, Minneapolis, Minnesota, May 11-13, 1998, pp. 16-23.

1. More extensive summaries addressing the full range of multimodal action can be found in [35] & [36].
2. Broadly speaking, we use the terms *reactive* and *reflective* here to refer to *fast* and *slow* responses, respectively. For a more detailed definition in this context see [35] & [36].

text of the dialogue.

3. *Functional & Morphological Substitutability*. Functional substitutability refers to the phenomenon when *identical looking acts can serve different dialogical functions* (one can point at an object by nodding the head, manual gesture, etc.). Morphological substitutability is the reverse: *Different looking acts can serve the same function* (a nod can serve a deictic function in one instance and agreement in another). This complicates decision making.

4. A dialogue participant's decisions to act (or not act) are based on *multiple sources*, both internal and external, including body language, dialogue state, task knowledge, etc.

5. Some *behavior is eventually produced*, no matter how limited sensory or cognitive information is available. If information is perceived to be missing, the behavior produced is likely to have the intent of eliciting the missing information. In this context, inaction can count as a decision to not act.

6. *Interpretation* of sensory input and dialogue state *is fallible*, resulting in erroneous decision making.

7. There can be both *deficiencies* and *redundancies* in the sensory data, making decision making more complex.

8. *Behaviors are under mixed control:* behaviors that are autonomous by default, such as blinking or gaze, can be instantly directed from the "top level" of control, as when you make the conscious decision to stare at someone.

What kind of architecture can successfully address all these issues simultaneously? Below we will first give an overview of the Ymir architecture, which is intended to address these issues, and the Gandalf prototype, then we will look at the decision mechanism in detail, and finally review relevant related work.

4. YMIR: OVERVIEW

Ymir[4] is a computational, generative model of psychosocial dialogue skills which can be used to create autonomous characters capable of full-duplex multimodal perception and action generation [35]. It is intended to attack the main characteristics of face-to-face dialogue, among them those listed above. It borrows several features from prior blackboard and behavior-based artificial intelligence architectures (discussed below), but goes beyond these in the amount of communication modalities and performance criteria it addresses.

Ymir contains three types of processing modules: *perceptual*, *decision* and *behavior*. While the separation of decision-making from other processes in cognitive models is not new [13], Ymir's modularity and process collections is based on recent work presented in [35] & [36]. The architecture's processing modules are found in four process collections, {1} a *Reactive* layer (RL), {2} a *Process Control* layer (PCL), {3} a *Content* layer (CL), and an *Action Scheduler* (AS). Multimodal data streams in to all three layers (see Figure 2). Each of the architecture's layers contains perception and decision modules that communicate results of their processing through blackboards. *Perception modules* with specific computational demands provide the necessary information about the state of the world to support *decisions to act* with specific perceive-act cycle times. The representation and distribution of perceptual modules in the layers are complementary to the decision modules: Perceptual modules in the RL have a relatively low accuracy/speed trade-off; decision modules in this layer share this same characteristic. More sophisticated perceptual processing and decision making happens in the PCL, and still more in the CL. Thus, economical use of computation is gained through bottom-up "value-added" data flow, where perceptual and decision modules in each successive layer add information to the results from the layer below. The layering also gives structure to top-down control of semi-automatic decisions: a decision in a lower layer may be overridden by a decision in a higher layer. For example, even though each of our eye fixations are normally chosen without our conscious intervention, we can still decide to stare at someone, and thus override the normal fixation control. In Ymir this is implemented by creating a decision module in the CL that can override more reactive decision modules in the two lower layers dealing with fixation control, and thus, unlike many other systems, a character created in Ymir can successfully heed the user's command to "Stop staring at me!".

FIGURE 1.
Gandalf references the planet Saturn (large monitor on right) with a manual deictic gesture and the verbal response "That is Saturn" (in this case a response to the author's act "What planet is [deictic gesture] that?"). The *decision* to reference the planet in a non-verbal channel is separate from its eventual *morphology*—whether to use a glance, turn of head or manual gesture—is decided at run time, chosen at the very end of the perception-action loop. This helps achieve high reactivity, e.g. the ability to stop an action at a moment's notice.

4.1 From Decision to Action

A decision module looks for conditions to "fire" (decide to act) in the blackboards. When a decision module fires it sends out an *action request*. The fate of the requests is determined in the next stage, in the *Action Scheduler*—the agent's "cerebellum". The Action Scheduler (AS) prioritizes the action requests, and decides how each requested action should look at the lowest (motor) level, according to the current status of the motor system (the agent's face and body, in the case of Gandalf). By divorcing the *decision to act* from its *form* in this way, decisions can be specified at various levels of abstraction; their exact morphology is determined at a later stage in view of the state of conflicting and currently executing actions. This increases the system's reactivity while allowing long-term planning. The approach taken in the AS's structure is in some ways similar to Rosenbaum et al.'s [29] model of animal locomotion and motion control. Their idea of stored postures is

3. This expectation is part of what has been referred to as the conversants' *common ground* [7].

4. "Ymir" is pronounced *e-mir*, with the accent on the first syllable. Like "Gandalf", the name comes from the Icelandic Sagas.

used in the Action Scheduler, as is the idea of hierarchical storage of increasingly smaller units. The AS is described in detail in [33] and [35].

4.2 Perception & Data Communication

Research has shown that in human perceptual processes, different information becomes available at different times: for example, low-frequency visual information and motion becomes available sooner than higher-frequency information and color (c.f. [4]). A person can select how long to wait before reacting to a particular stimulus, depending on the selected trade-off between cost of delay and cost of errors. This requires a system where information is incrementally processed (anytime algorithm) and can be accessed at any time by decision mechanisms. In Ymir, blackboards are used for this purpose.

There are three main blackboards (Figure 2). There is one for information exchange between the Reactive Layer and the Process Control Layer. This blackboard is called the *Functional Sketchboard* (FS). The name refers to the blackboard's primary role and form of data—initial, rough *"sketches"* of the *functions* of a dialogue participant's behaviors. It stores intermediate and final results of low-level, high-speed perceptual processes such as object motion, whether the agent hears something or not, and other information crucial to the dialogue *process*. These perceptual data serve as conditions for the decision modules in the RL and PCL. The second is the *Content Blackboard* (CB), servicing communication between the PCL and the CL. Here results are posted that are less time-critical than those on the FS, and usually more computationally expensive. The Process Control Layer's decision modules look primarily in this blackboard for conditions, but they can also access data in the FS. The third blackboard in Ymir is the *Motor Feedback Blackboard* (MFB), where the Action Scheduler posts the progress of behaviors currently being processed and executed. The MFB enables the PCL and CL to access the history of motor acts so that a character can have recollection of its own actions and thus make longer-term motor plans and modify them when necessary. The perception-act cycle in the RL is so short that this internal feedback is not useful; the feedback loop for motor acts in the RL is the real-world.

4.3 Gandalf: A Humanoid Prototype

To test the premises of Ymir, a prototype humanoid called *Gandalf* has been designed (Figure 1). Gandalf's modules were constructed with the single purpose of enabling it to carry out embodied, topic-oriented dialogue. For the most part, this was done by data-mining the psychological literature (c.f. [9], [10], [11], [16], [19], [22], [25], [28]). The system has proven to be capable of fluid turn-taking and unscripted, task-oriented dialogue. *Gandalf is capable of producing real-time multimodal behaviors in the following categories:*

- 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, facing user when answering questions.
- Eyes: **Attentional and deictic functions**, both during speaking and listening, e.g. looking where the user is looking or pointing; looking at objects that he is telling the user about.
- Body: **Emblematic body language** — nodding, shaking head.
- Speech: **Back channel feedback** and **meaningful utterances**.
- **Turn-taking signals**: Looking quickly away and back when taking turn, attentional cues such as head and gaze direction (gaze deixis), greeting (in various ways), and more.

These are all coordinated in real-time and correctly inserted into the dialogue. They are based on the *perception and interpretation of the following kinds of user behavior data:*

- 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.

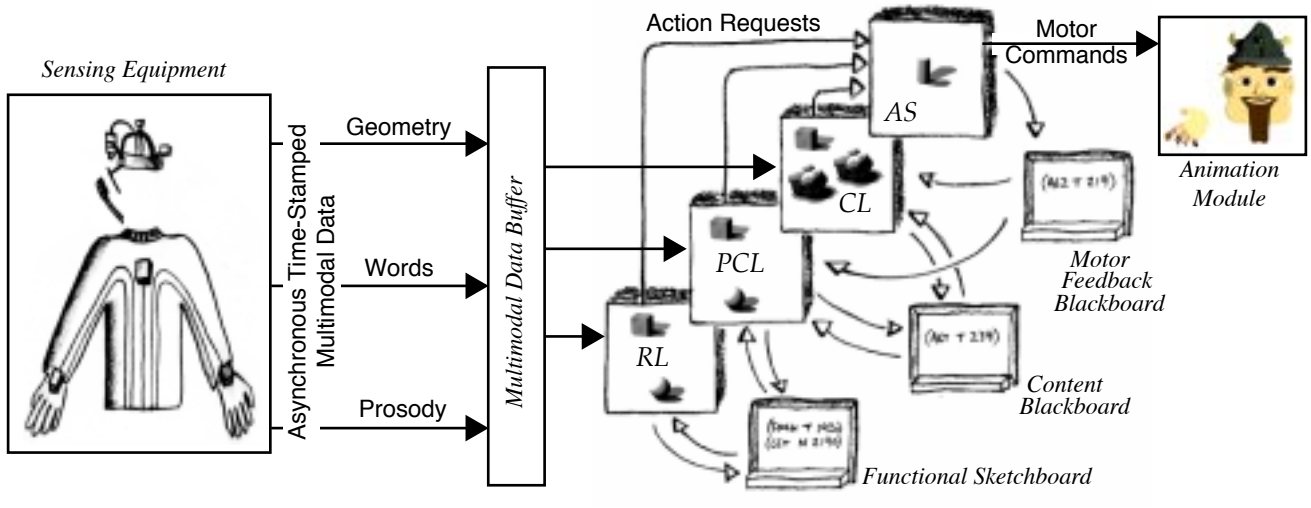

FIGURE 2.
Data flow into and between the layers (RL, PCL & CL), Action Scheduler (AS) and the three specialized blackboards of the Ymir architecture. (Spheres = perceptual modules; cubes = decision modules; blobs = knowledge bases; cylinder = behavior modules; see text.) The sensing equipment shown was used in the Gandalf prototype: body-tracking jacket, data gloves, eye tracker and microphone.

NAME: show-I-know-user-is-addressing-me TYPE: RL-Overt-Dec-Mod EXP-LT: 20 ACT-REQUEST: eyebrow-greet MSG-DEST: FunctionalSketchboard FIRE-CONDS: ((user-saying-my-name T) (user-turned-to-me T) (user-facing-me T)) RESET-CONDS: ((user-turned-to-me NIL))	**1**
NAME: show-listening-1 TYPE: PCL-Overt-Dec-Mod EXP-LT: 2000 ACT-REQUEST: (Turn-To 'user) MSG-DEST: ContentBlackboard FIRE-CONDS: ((user-speaking T) (user-addressing-me T)) RESET-CONDS: ((I-take-turn T))	**2**
NAME: hesitate-1 TYPE: PCL-Overt-Dec-Mod EXP-LT: 100 ACT-REQUEST: hesitate MSG-DEST: ContentBlackboard FIRE-CONDS: ((dial-on T) (I-take-turn T) (spch-data-avail T) (user-speaking nil) (BB-Time-Since 'user-speaking 70) (CL-act-avail NIL)) RESET-CONDS: (I-give-turn T)	**3**
NAME: parse-speech TYPE: PCL-Covert-Dec-Mod EXP-LT: 200 ACT-REQUEST: (parse-speech) MSG-DEST: ContentBlackboard FIRE-CONDS: ((user-giving-turn T) (spch-data-avail T)) RESET-CONDS: ((I-give-turn T))	**4**

FIGURE 3.
Examples of decision module representation and slot values used in Gandalf (see text for explanation; ACTIVE slot not shown). *1*. Overt decision module in the Reactive layer that decides when to greet with an eyebrow-lift; *2*. one of several overt decision modules in the Process Control layer that determines when Gandalf turns to his conversant; *3*. overt module that decides to hesitate when the character has understood what the user said (spch-data-avail), yet it has failed to come up with a reply (CL-act-avail) 70 centiseconds after the user became quiet (BB-Time-Since 'speaking 70); *4*. covert decision module in the Process Control Layer that determines when to parse the incoming speech. T=TRUE, NIL=FALSE.

- Eyes: **Attentional and deictic functions**, both during speaking and listening, e.g. using the user's gaze to infer which object is referenced by words like "that one".
- 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.
- Body: **Direction of head and trunk** — e.g. when user turns away to talk to visitors instead of Gandalf, and **position of hands in body space** (hand position relative to trunk and head), which is important when interpreting gesture.
- **Turn-taking signals**: Various feature-based analysis of combinations of related and/or co-occurring multimodal events, such as intonation, hand position and head direction.

In addition, Gandalf is capable of task-related activities, in this case perceiving and manipulating a graphical model of the solar system. The main missing I/O elements in this prototype are the full range of manual gesture [10][28], intonation control in the output, facial expression in the input and more extensive intonation analysis in the input. However, the current mechanisms in Ymir are believed to be able to allow these additions.

Gandalf has been tested in interaction with computer-naïve users, who have rated him highly on believability, language ability and interaction smoothness. As a baseline, all subjects found Gandalf "much more life-like than interaction with a real fish in a fish bowl" and all have found the smoothness of the interaction to be either "somewhat better" or "much better" than the smoothness of interacting with a real dog.[5] After interacting with Gandalf, 36% of the subjects reported an increased belief that "in the future, computers will become intelligent"; 55% reported no change in their belief; none reported a decreased belief. Comparing Gandalf's performance to interaction with a human on a scale from 0 to 10, subjects rated Gandalf's language *use* to be 79% as good as that of a real person, his language *understanding* to be 73% as good as that of a real person, and the interaction to be 63% as smooth as a real human face-to-face interaction. It is important to note here that believability ratings for Gandalf's command of *language* were significantly lower when his multimodal behavior—facial and manual gesture, eye and head movements—were turned *off*, further supporting the conclusion that his multimodal, human-like dialogue is relatively convincing. For further results of user testing see [35] & [40].

5. DECISION MAKING: THE DETAILS

Decision modules in Ymir look at the internal representation of the outside world as well as the status of internal processes, and make decisions as to what to do from moment to moment. Decisions affect either the outward behavior of the agent or the processing inside the agent's "mind", and fall thus broadly into two categories:

1. *Overt decision modules*—those that initiate external, visible actions, and
2. *covert decision modules*—those that only change the internal state.

The separation of modules into overt and covert decision is also seen in the Hap architecture [1]. In Ymir, each decision module has an associated *action* (or "intention to act") and a *condition list*. If the conditions are fulfilled, the intention "fires". This means that it either results in some internal process running *or* some outward behavior being executed. Each decision module contains knowledge about where to look for data (which blackboard), what to do with it and how to communicate its status to other modules by posting information to blackboards.

A decision module for lifting eyebrows when being looked at by the user ("informal greeting") may look something like module 1 in Figure 3. This decision belongs to the Reactive Layer (RL). The *Process Control Layer* (PCL) contains decision modules that mostly concern the dialogue process, e.g. when to parse the incoming speech, when to report problems with the dialogue ("oh, what's the word...", "Do you mean this one?"), etc. To do this, these decision modules use a protocol to communicate with the knowledge bases in the *Content Layer* (CL) (see Figure 2). A small sample of communication primitives used for this purpose is shown in Figure 4. The CL produces multimodal actions; decision modules in this layer are concerned with planning internal events, requesting multimodal actions to be executed and modifying these on the fly while monitoring the MFB.

5.1 Scheduling of Decisions & Actions

Intentions to act in Ymir are ensured timeliness in two ways: {1} By priority scheduling, where requests initiated by modules in the RL take priority over PCL-initiated requests, which in turn take priority over CL-initiated actions; and {2} by a time-management system that ensures that actions that didn't get executed in time will not be. This is done by giving decisions to act an *expected lifetime*

5. All data based on a convenience sample of 12 subjects [35].

value. When a decision is made, an action request, along with this value and a time stamp, are sent to the AS. If the expected lifetime has been reached before the AS has found a morphology for it, the behavior is cancelled. In the Gandalf prototype, the expected lifetime is a fixed value based on psychological studies of human face-to-face communication. In future versions the expected lifetime will be a mixture of constants and run-time computation, based on the performance of the system.

A "full-loop response cycle" refers to the time from the moment when a particular dialogue event happens until the perceiver of that event starts to execute a response to that event. Decisions to act resulting from processing in the RL generally support full-loop response cycles under 1/2 second, typically in the 150-250 ms range—actions like blinking, determining the next fixation point and giving back-channel feedback (see Figure 5). Decision cycles in the PCL have a frequency around 1 Hz—actions like taking speaking turn or looking at someone who is addressing you. The CL contains the dialogue- and domain-specific knowledge bases. Processing in the CL has response times from seconds up to infinity. Notice that these numbers are not based on current or future computing power of machines—they are based on socially accepted response times in human face-to-face dialogue and on the computational limitations of the human mind/brain.

5.2 Decision Module Representation

In the Lisp prototype of Ymir, decision modules have been implemented as object classes. A decision module has the following slots: (1) FIRE-CONDS—a list of conditions that, when all are met, will make the module fire (turn its own state to `TRUE` and send out an action request), (2) RESET-CONDS—a list of conditions that, when all are met, will reset the module's ACTIVE slot to `TRUE`, (3) EXP-LT: *expected lifetime* of the module's action requests—how long an action request stays valid, from the time it is requested by this decision module, *before* it starts to be executed by the AS, (4) STATE, containing the boolean state of the module, (5) ACT-REQUEST, containing the *action request* that is posted when the module changes state, (6) ACT-DEST, containing a pointer to `act-dest`, the destination for posting a change in the module's state (one of the blackboards), and (7) ACTIVE, a boolean state determining whether the module can send out action requests. When all the conditions in the FIRE-CONDS list are met simultaneously, the module's STATE is set to TRUE, this fact is posted to `act-dest`. In this state, the module waits to be reset before it can fire again. Overt behaviors send action request messages to a buffer in the Action Scheduler. Covert modules contain a *function* name in their ACT-REQUEST slot that is executed when the module fires.

COMMUNICATION FROM CL TO PCL	
Speech-Data-Avail	KB-is-Exec-Act
DKB-Rcv-Speech	TKB-Exec-World-Act
KB-Succ-Parse	
TKB-Act-Avail	TKB-Exec-Speech-Act
CL-Act-Avail	DKB-Exec-Act
KB-is-Exec-Act	Exec-Done

FIGURE 4.
A subset of communication primitives between the Process Control Layer and Content Layer, the latter of which contains a Dialogue Knowledge Base (DKB) and one Topic Knowledge Base (TKB). (Rcv = received, Act = action; Exec =executing/execution; avail = available; succ = succsessful.)

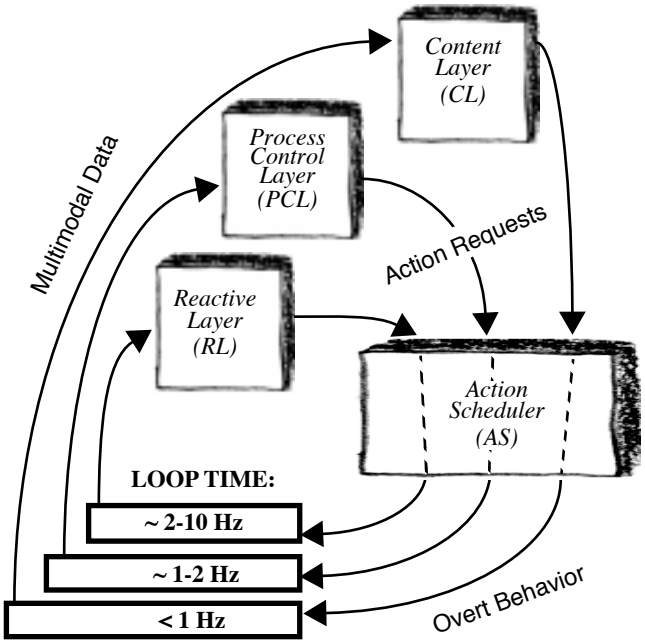

FIGURE 5.
Multimodal input maps into *all three layers* in Ymir. Decision modules operate on these results and decide when to send Action Requests to an Action Scheduler (see Figure 2), which then produces visible behavior. Target loop times for each layer is shown in Hz. It is important to note here that the frequency refers not to the layers' internal update time or sampling rate, nor to the speed of decision making, but to a full action-perception loop.

In dialogue, the idea of being in a particular state can be useful, because certain actions (e.g. back channel feedback) are only produced in a particular state (e.g. the "listener state"). A problem with the simple overt and covert modules described above is that they can't cause actions based on being in a particular *state*. To solve this, *State Transition modules* are made for keeping track of things such as dialogue state, turn state, etc. They can be thought of as the transition rules in a Transition Network with the important difference that they can lead to more than one new state, and they are augmented with a global clock. State modules toggle other processes between ACTIVE and INACTIVE. Among other things, this can provide a mechanism for a simple "narrowing of attention" for the agent by deactivating certain perceptual processes and thus limiting the range of perceptual data that the agent is sensitive to at that moment.

5.3 Methods for Decision Modules

Four methods are defined for decision modules: `UPDATE`, `FIRE`, `ACTIVATE`, and `DEACTIVATE`. `UPDATE` supplies a module with access to all the data it needs to make its decision, and sets it to `TRUE` if all conditions in its FIRE-CONDS lists are met—these are `AND`ed. If a module is ACTIVE and its STATE is `TRUE`, then the `FIRE` method {1} posts the module's state to blackboard `act-dest` and {2} sends its action request to the Action Scheduler (or executes the internal function if a covert decision module), {3} sets the module's STATE to `FALSE` and {4} calls `DEACTIVATE` on the module, which sets the module's ACTIVE slot value to `FALSE`. If the module's ACTIVE slot is `FALSE`, the conditions in its RESET-CONDS list are checked in `UPDATE`, and if all of them are met (these are

also ANDed), the module's ACTIVE slot is set to TRUE by calling ACTIVATE on the module. (By keeping the activation processes as separate methods, these can be called on the modules from other places in the system.) In this state the module is again ready to FIRE.

Each module's FIREing is time stamped when posted to a Blackboard, allowing other modules to activate based on the age or pattern of any message(s) reported on the blackboard. The generic operator BB-Time-Since gives the last posting time for a given module, and can be used in any decision module's condition lists.

The Gandalf prototype has shown that mechanisms for decision making can be made relatively simple, provided sufficient perceptual data. The boolean nature of Gandalf's decision modules supports a level of complexity, within the larger framework of the Ymir architecture, sufficient to keep up a relatively natural, free-form interaction. Another candidate mechanism for decision making is fuzzy logic (c.f. [18]), which could replace the current decision mechanism directly without any modifications to the rest of the system. First-order logic is a good option, however, when processing power for running the whole system may be compromised or cannot be estimated precisely. It may be pointed out that in systems such as this the decisions themselves are always boolean; there are no "halfway" made decisions.

5.4 Cascaded Decision Modules

To understand the idea of cascaded decision modules, let's take an example: Suppose you're engaged in a dialogue and the other party stops talking. You think she's going to continue, but a moment later you realize that she was asking you a question and is expecting you to answer. At this point you want to show that you have realized that she is expecting an answer, but instead of showing this the typical way (which typically may be a subtle raising of the eyebrows) you decide to do something different: you quickly say "ah!", look away in an exaggerated manner while starting to formulate an answer. When you realized you had failed to take turn the "normal" way, you chose (read: *decided* to *execute*) a different behavior. The new decision results in a motion that looks different from the default behavior. However, the alternative behavior you decided on still serves the same purpose of *showing that you know that you're expected to respond*. Thus, the two decisions belong in a group with a *common functional purpose*. In Ymir, the two behaviors are triggered by two separate decision modules: the alternative behavior in this example is generated by a second decision module in a group of *cascaded* decision modules, all with the same function, namely, that of showing the participating conversational parties that you are about to respond in some way. In the second module's FIRE conditions is the failure of the first module to FIRE, a condition that is avaiable to in one of the Blackboards. By cascading a number of decision modules, each representing a variation on behavior morphology, and each triggered in the case of another's cancellation, inappropriateness, or failure, whole classes of behaviors can be built up in a flexible way. This scheme has worked well for constructing behaviors in the Gandalf prototype.

5.5 Remaining Issues

The main issues of the decision mechanism that need to be further addressed are {1} more extensive interaction between the Motor Feedback Blackboard and the Content Layer, to allow for real-time modifications of motor plans; {2} a stronger support for perceptuomotor loops, necessary for grasping objects and manipulating them, {3} a learning mechanism to allow decision making to improve over time, and {4} a more extensive testing of the malleability of the decision mechanism itself, in the context of the rest of the Ymir system.

6. RELATED WORK

Approaches taken to date to the creation of autonomous characters can be classified roughly into two categories, "classical A.I." and "behavior-based A.I." (c.f. [20]). As Brooks [3] pointed out at the beginning of the decade the approaches are certainly complementary. Both certainly have features to offer for the design of communicative agents. An example of a behavior-based A.I. system is Maes' *competence module* architecture—software modules that contain enough information to execute a particular behavior from beginning to end (c.f. [2], [21]). The modules are connected together by neural-like activation links that determine their sequence of execution. The input to the modules can come both from internal goals and the environment. Decision making is made by executing the program contained in the module with the highest activation level at any moment. This architecture, and other similar approaches are very good for effective, fast decision making, and some allow learning. However, they lack methods to deal with external and internal time-constraints and are limited in the planning they can handle.

Blackboard architectures [31] were invented as a method to handle unpredictable information like that encountered in speech recognition and planning [17][23]. The blackboard architecture attacks the problem of unpredictability in the input data by the use of a common data storage area (blackboard) where results of intermediate processing are posted and can be inspected by other processes working on the same or related problem. Modifications to the original blackboard idea include mechanisms to allow interleaved execution of subsystems, as well as communication between them [12], resource management, speed/effectiveness trade-off and reactive systems behavior [8]. The principles of these architectures are very useful for real-time multimodal communication systems.

Working on a piece of the multimodal puzzle, Cassell et al. [5][6] describe a hybrid system for automatic speech and gesture generation. The system employs two graphical humanoid characters that interact with each other using speech, gaze, intonation, head, face and manual gesture. The system employs what the authors call PaT-

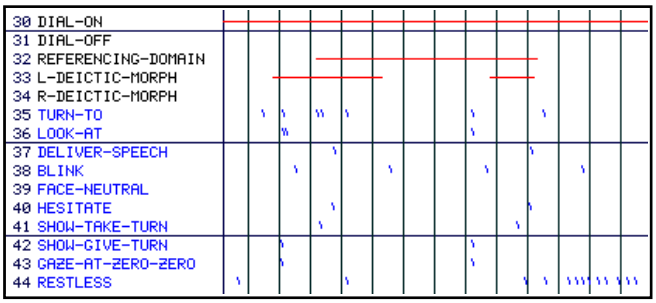

FIGURE 6.
An example of event timing for part of Gandalf's perceptual modules (lines 30-34) and decision modules (lines 35-44) during interaction with a user, for a period of 14 seconds (each interval = 1 sec.). Gray horizontal lines show when a perceptual state is true, small tics mark moment of decision. (DIAL-ON = dialogue ongoing; REFERENCING-DOMAIN="is user looking/pointing at a relevant object?"; DEICTIC-MORPH="does the morphology of a user's hand-posture and arm-motion indicate a pointing gesture?"; TURN-TO=agent decides to turn head in a (variable) direction; LOOK-AT = point eyes in a (variable) direction; GAZE-AT-ZERO-ZERO=look at user (by pointing head and eyes straight out of monitor).)

Nets (Parallel Transition Networks) in which synchronization between gestures and speech is accomplished as simultaneously executing finite state machines. The system highlights the complexities of synchronizing various levels of multimodal action generation, from the phoneme level up to the phrase and full utterance, but leaves open the complicating issue of real-time adaptation and interaction between real humans and synthetic ones.

Hap is an architecture for creating broad agents with goals, emotions, planning and perceptuo-motor capabilities [1]. It addresses flexibility of plan execution and goal-directed planning, as well as real-time natural language generation. Whereas Hap's decision mechanisms were directed toward plan execution and language generation, Ymir's decision system addresses the full timing range of human action — from milliseconds to hours to days — in addition to natural language (spoken or otherwise). The specificity of the Hap architecture is also more fine-grained; Ymir being more of a meta-structure that could in fact accommodate the Hap method of planning. Another big difference lies in the complexity of the kind of sensory and motor input it addresses: Like Cassell's et al. work, Hap is directed at synthetic characters that mostly interact with each other inside simulated worlds. Ymir deals primarily with dialogue between synthetic characters and *real humans*, addressing the full range of multiple modes relevant in face-to-face conversation.

All of the systems reviewed lack one (or more) of the *three crucial ingredients* in face-to-face dialogue, *multimodal action generation and integration*, use of *natural language* and *real-time response* (dialogue-relevant perception and action timing). This makes it very difficult to apply any one of them directly to humanoids that participate in face-to-face interaction. A strong dichotomy exists in many of the prior systems between language capability and action generation/coordination. A few, like Cassell et al.'s system [5][6], integrate both in a consistent way. However, PaT-Nets are generally not a good solution to resource-allocation and real-time control. Hap is a relatively broad architecture that integrates planning, emotion and natural language in a concise way, but the main weakness of Hap is the simplicity of its perception and motor systems, which make it difficult to take advantage of a richer set of input data and output mechanisms, or move it outside of a simulated world. In behavior-based systems, such as Brooks' [3] and Maes' [20], interfaces between action control modules are defined at a relatively low level— creating large systems in them can be problematic at best, impossible at worst. But their greatest problem by far is adding natural language capabilities.

7. SUMMARY & FUTURE WORK

The decision-making mechanism described in this paper results in a system with several novel features. Because the action selection and scheduling mechanism is based on the real-time requirements of human face-to-face interaction, concurrent behaviors, such as glancing over to an object the speaker points at and nodding at the same time, happen naturally, where and when expected. The decision process includes feedback loops at multiple levels of granularity; a character's behavior is therefore interruptible at natural points in its dialogue with people, without being rigid or step-lock. The architecture models bottom-up processing and incremental (anytime) computation of perceptual data while allowing top-down process control through covert decision modules, supporting both top-down and bottom-up processing. By providing layers that address the issue of resource management in this logical way, along with a modular approach to modelling perception, decision and action, new features and modules can be added incrementally without resulting in exponentially increasing complexity. Decisions in semi-independent layers that are directed at different time-scales (reactive, reflective) produce relatively coherent, reliable, and believable behavior. This is done by separating decision from the morphology of action using a dedicated action coordination/composition processor, along with an inherent action prioritization scheme.

Future work on the decision mechanism focuses on building larger action repertoires and more sophisticated decisions, testing the architectures flexibility further. Part of this work will involve extending the agent's understanding of the dialogue, its participants, and decisions about the dialogue process. The decision and motor mechanisms in Ymir are very relevant to semi-autonomous avatar control (c.f. [41]). Gandalf's topic knowledge and action repertoire are also being extended [26], and Ymir is being extended to control mobile agents.

8. ACKNOWLEDGMENTS

The author would like to thank the sponsors of this work: Thomson-CSF, the M.I.T. Media Laboratory, RANNÍS and The HUMANOID Group. Thanks to Justine Cassell, Pattie Maes, Steve Whittaker, Tom Malone & Richard A. Bolt for guidance; Joshua Bers, David Berger, Christopher Wren, Steven Levis, Calvin Yuen, Nimrod Warshawsky, Roland Paul & Hannes Vilhjálmsson for technical assistance. Thanks also to my anonymous reviewers.

9. REFERENCES

[1] 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.

[2] Blumberg, B. (1996). Old Tricks, New Dogs: Ethology and Interactive Creatures. Ph.D. Thesis, Massachusetts Institute of Technology.

[3] Brooks, R. (1990). Elephants Don't Play Chess. In P. Maes (ed.), *Designing Autonomous Agents*, 3-15. Cambridge, MA: MIT Press.

[4] Card, S. K., Moran, T. P. & Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Hillsdale, New Jersey: Lawrence Earlbaum Associates.

[5] Cassell, J., Pelachaud, C., Badler, N., Steedman, M., Achorn, B., Becket, T., Deouville, B., Prevost, S. & Stone, M. (1994a). Animated Conversation: Rule-based Generation of Faceial Expression, Gesture & Spoken Intonation for Multiple Conversational Agents. *Proceedings of SIGGRAPH '94*.

[6] 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.

[7] 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.

[8] Dodhiawala, R. T. (1989). Blackboard Systems in Real-Time Problem Solving. In Jagannathan, V., Dodhiawala, R. & Baum, L. S. (eds.), *Blackboard Architectures and Applications*, 181-191. Boston: Academic Press, Inc.

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

[10]Effron, D. (1941/1972). *Gesture, Race and Culture*. The Hague: Mouton.

[11]Ekman, P. & Friesen, W. (1978). Facial Action Coding System. Palo Alto, CA: Consulting Psychologists Press.

[12]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.

[13]Glass, A. L. & Holyoak, K. J. (1986). *Cognition*. New York, NY: Random House.

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

[15]Goodwin, C. (1981). *Conversational Organization: Interaction Between Speakers and Hearers*. New York, NY: Academic Press.

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

[17]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.

[18]Kacprzyk, J. (1992). Fuzzy Sets and Fuzzy Logic. In S.C. Shapiro (ed.), *The Encyclopedia of Artificial Intelligence*, 2nd ed., 537-542. N.Y.: Wiley Interscience.

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

[20]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

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

[22]McNeill, D. (1992). *Hand and Mind: What Gestures Reveal about Thought*. Chicago, IL: University of Chicago Press.

[23]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.

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

[25]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.

[26]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.

[27]Poyatos, F. (1980). Interactive Functions and Limitations of Verbal and Nonverbal Behaviors in Natural Conversation. *Semiotica*, 30-3/4, 211-244.

[28]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.

[29]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.

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

[31]Selfridge, O. (1959). Pandemonium: A Paradigm for Learning. *Proceedings of Symposium on the Mechanization of Thought Processes*, 511-29.

[32]Steels, L. (1990). Cooperation Between Distributed Agents Through Self-Organization. In Y. Demazeau & J. P. Müller (eds.), *Decentralized A. I.* Amsterdam: Elsevier Science Publishers B. V. (North-Holland).

[33]Thórisson, K. R. (1997a). Layered Action Control in Communicative Humanoids. *Proceedings of Computer Graphics Europe '97*, June 5-7, Genieva.

[34]Thórisson, K. R. (1997b). Gandalf: A Communicative Humanoid Capable of Real-Time Mutlimodal Dialogue with People. *ACM First Conference on Autonomous Agents*, Marina del Rey, California, February 5-8.

[35]Thórisson, K. R. (1996). Communicative Humanoids: A Computational Model of Psychosocial Dialogue Skills. Ph.D. Thesis, Massachusetts Institute of Technology.

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

[37]Thórisson, K. R. (1995b). Multimodal Interaction with Humanoid Computer Characters. *Conference on Lifelike Computer Characters,* Snowbird, Utah, September 26-29, p. 45 (Abstract).

[38]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.

[39]Thórisson, K. R. (1993). Dialogue Control in Social Interface Agents. *InterCHI Adjunct Proceedings '93*, Amsterdam, April, 139-140.

[40]Thórisson, K. R. & Cassell, J. (1997). Communicative Feedback in Human-Humanoid Dialogue. *IJCAI '97 Workshop on Animated Interface Agents*, Nagoya, Japan, Aug. 25-26.

[41]Vilhjalmsson, H. H. & Cassell, J. (1998). BodyChat: Autonomous Communicative Behaviors in Avatars. *This volume*.

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