

Architectural issues for integration of sensing and acting modalities

Henry Hexmoor
Department of Computer Science
University of North Dakota, ND 58202
Cesar Bandera
Amherst Systems, Inc.
Buffalo, NY 14221

Abstract

The goals of this paper are twofold. First, we present our experiences with (1) fusion of vision, sonar, and contact sensory modalities in perceiving obstacles and targets, (2) arbitration of sensing and acting at reactive and deliberative levels, and (3) integration of asynchronous instruction taking and communication. Following a discussion of lessons learned we discuss architectural modules and issues that help standardize integration of sensor and acting modalities and solution across platforms.

We have developed a robot assistant, which communicates in natural language and moves in a room using vision as its primary sensory mechanism. This system takes instruction from a supervisor to go to various agents in the room and follow them along to offer assistance. Our assistant uses a three tiered architecture which models: (a) knowledge representation and reasoning including natural language interactions, (b) routine interactions not explicitly controlled by the agent, and (c) reflexes. Our robotic assistant integrates (a) vision and sonar sensing in obstacle avoidance, (b) memory-based and reactive navigation (i.e., deliberative versus skill-based), (c) instruction taking and goal-driven behaviors, and (d) concurrent visual focusing behaviors.

1. Introduction

We have designed and implemented a robotic assistant (henceforth we will call it Freddy) which is instructed in natural language with commands such as "Find that robot" or "Follow this green robot." The purpose of this work is to evaluate techniques for the arbitration of concurrent behaviors driven simultaneously by bottom-up sensory data (typically visual data) and by top-down instructions (typically natural language commands).

The physical robot is an augmented Nomad 200 (Figure 1). Freddy merges both anthropomorphic and neuromorphic designs. Anthropomorphism is represented by the Grounded Layered Architecture with Integrated Reasoning (GLAIR), which interfaces semantic perception with a perceptuo-motor level (PML) and a sensori-actuator level (SAL) [Hexmoor, et al 93] [Hexmoor 96]. Neuromorphism is represented by Hierarchical Foveal Machine Vision (HFMV), which exploits the multiacuity sensing and processing prevalent among vertebrates to achieve an effective visual information acquisition power that is higher than that of traditional uniform acuity machine vision [Bandera, et al 89]. Crucial to the successful operation of Freddy in its non-deterministic environment is the proper execution of foveal gaze control [Bandera, et al 96]. This emphasizes the arbitration of simultaneous demands for visual attention from competing behaviors.

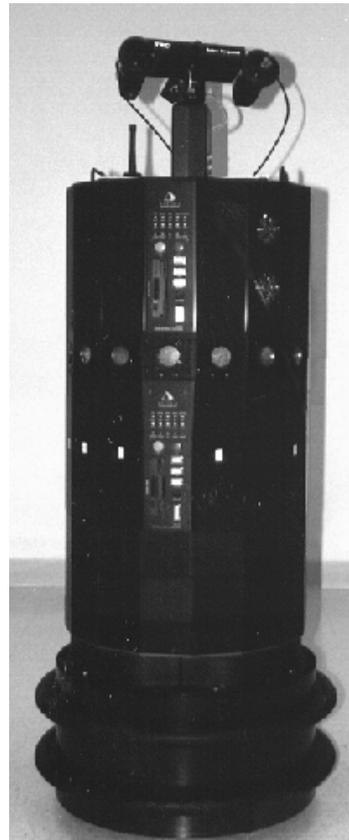


Figure 1: Freddy autonomous mobile robot.

Freddy can reason and communicate about other agents. In carrying out a command, Freddy decomposes the command into primitive actions and hands each off to the corresponding parts of its body, which may further decompose actions into skills and carry out the skills automatically. The robot can carry out these primitive actions and reason about or speak about them. However, Freddy cannot reason or speak about skills.

When executing a syntactic command, such as “go to John and then find the red robot,” Freddy decomposes the sentence into primitive behaviors, which are members of the set {“find,” “go,” “wander,” and “stop”}. Each primitive behavior is further broken down into a number of low-level operations, such as motor control, visual servoing (synchronization of vision and movement), obstacle avoidance, and geometry computation. With this decomposition, one can think of Freddy as possessing “conscious” and “unconscious” behaviors. The primitive commands “find,” “go,” “wander,” and “stop” constitute the Freddy’s consciousness, or behaviors that Freddy can reason about, arbitrate, schedule, or sequence. The unconscious functions implementing the higher level primitives are not under Freddy’s direct control. That is, Freddy can approach a problem and create a high-level command sequence such as [“find” -> “go” -> “stop”], but cannot link together a string of motor commands, distance measurements, or vision requests.

Freddy performs one user command at a time. The user can interrupt Freddy with a new command before completing the old one, i.e., consciously, Freddy does one thing at a time. However, once the primitive command is decomposed into the unconscious parts of the robot, concurrent behaviors are activated to maintain basic functions such as obstacle avoidance and maintaining proper distances. It is in the unconscious regions that Freddy has to coordinate concurrent behaviors. This seems consistent with psychological theories for performing dual tasks.

2. Instruction Taking in Natural Language

Freddy’s command language is embedded in a larger language that includes statements and questions. Relevant statements that Freddy can understand include “Who did you go to?” and “Who have you followed?” The command language is:

```
<command> ::= Stop
           ::= Come here | Go there
           ::= <action> <np> [and [then]
                               <command>]
```

```
<action> ::= Go to | Follow | Help | Find | Look at
           | Talk to
```

```
<np> ::= <npr>
       ::= <pron>
       ::= (this | that | a | an | the) [<adj>] <n>
       ::= <np> and [then] <np>
```

The relevant lexicon is:

```
<adj> ::= green | red
<n> ::= astronaut | robot | supervisor
<npr> ::= John | Stu
<pron> ::= me | him | her | it
```

Whoever Freddy is currently talking to (Stu or John) may use the first person pronoun (“me”). A sequence of commands may be given using ellipsis on the verb, for example, “Follow John and then the green robot.” and/or the preposition, for example “Go to John and then the green robot.”.

In addition, “then” is optional. An interesting command that Freddy understands and can carry out is “Go to the green robot and then come here and help me.” The use of “this” or “that” must be combined with a deictic gesture pointing to an object. If the user uses an ambiguous definite “np” such as “the red robot,” Freddy responds “Which one do you mean?” and the user may then enter a deictic “np” and point to the intended object.

3. Actuators, Sensors, and Reflexes

Head and body are the actuation mechanisms in Freddy. These operate and are commanded independently. The head provides information about objects of interest in a visual range (Figure 2). The head consists of two color video cameras mounted on a fast 3-DOF binocular mechanism (pan, tilt, vergence), which in turn is mounted on a slower turret that rotates 360°. The head includes a C-40 based multiprocessor that performs foveal scene understanding and outputs visual servoing control signals. This vision system provides information about relevant objects in the visual range, whereby relevance is itself time varying and defined by the visual behaviors that are active.

Vision is the predominant sensing mechanism in Freddy. It is used for three purposes: (1) visual obstacle avoidance, (2) visual search for a target of given shape and color, and (3) visual tracking of a detected target. Except when instructed to visually interrogate a region or track a moving target, as a reflexive behavior Freddy returns what it sees in its

field-of-view. Freddy's head can be commanded to search a region in space looking for an object of a given color and shape.

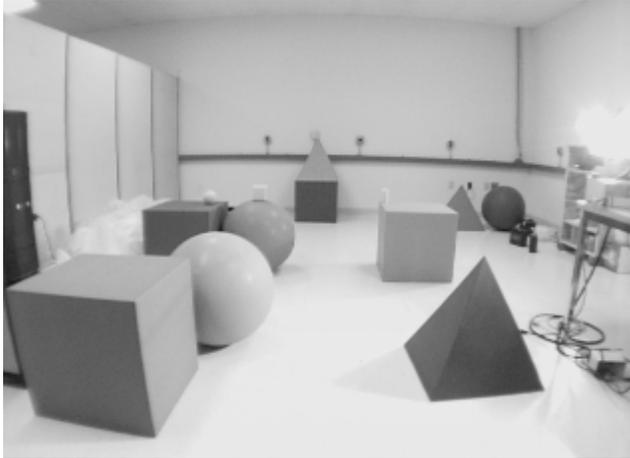


Figure 2: A typical view from Freddy's vision system

Freddy's body can move in the current heading and it can be steered simultaneously. Commands to the body are a suite of turn and translation.

Our complete set of sensors is a bumper ring, a sonar ring, and the vision system. The bumper is used as a last resort for obstacle avoidance, when Freddy's vision and sonar has failed to alert it about obstacles. We have a reflex associated with the bumper that backs up Freddy when it collides with the walls or obstacles. Sonar is the intermediate range sensing used primarily to avoid obstacles when the vision system fails to adequately alert Freddy about obstacles. We have developed a reflex associated with the sonar that steers the robot away from the walls or other obstacles until the front of the robot is clear.

Freddy concurrently uses vision for obstacle detection and aversion as well as target detection and tracking. Our robot also concurrently instructs its head and body for motion by different behaviors. In the following section we present the algorithms for behaviors that show fusion of vision and sonar data in navigation and obstacle avoidance.

4. Architecture of Behaviors

Freddy performs only one user command (or conscious behavior) at a time; the robot cannot "find a red robot" and simultaneously "go to a green sphere." However, at the unconscious level, some of the behavior primitives are implemented as a concurrent combination of basic functions. For example, "go" is a

concurrent combination of finding a target, obstacle avoidance, remembering the robot position relative to the target, and moving forward. Therefore, by unconsciously coordinating low-level actions, Freddy pieces together higher, conscious behaviors. This idea is consistent with psychological theories for performing dual tasks.

In psychology, it is argued that task execution can be divided into three successive stages of (a) perceiving the stimulus, (b) choosing the response, and (c) producing the response [Pashler 93]. The strongest theory is that the second stage, choosing the response, is the bottleneck for concurrent tasks. This means the agent's sensory "attention" operates in parallel so the agent can perceive multiple stimuli. Similarly, producing responses can be done in parallel.

In order to synchronize the conscious and unconscious behaviors and remember the target properties, short-term visual memory is implemented as global data structures. The iconic short-term memory (ISTM) resides in the unconscious regions of Freddy and contains the properties of the last target seen. The conscious counterpart of ISTM is denoted as KL-STM, and is also the short-term memory of the intended target. Once the robot decides to look at a target, KL-STM maintains the target description, thus simulating visual attention. Finding an object involves passing the description of the object to the PML, which uses that description to identify an object, and stores that object in ISTM.

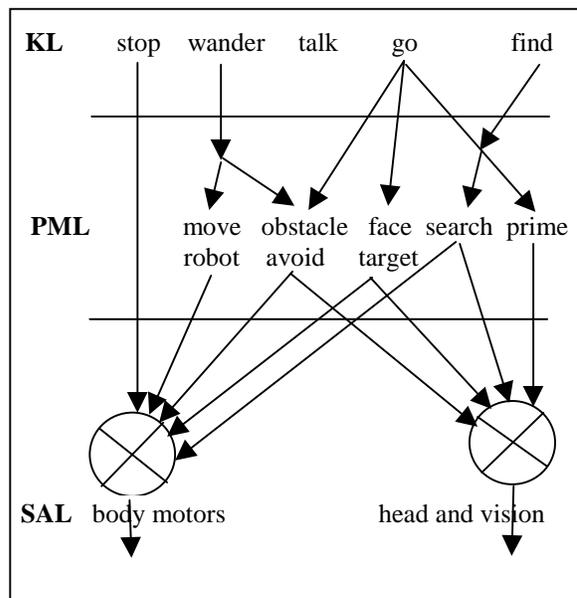


Figure 3: Decomposition of behaviors. What is shown at the KL are primitive behaviors after commands are parsed.

Figure 3 shows the decomposition of the behavior primitives “stop,” “wander,” “go,” and “find” into unconscious actions and motor commands. The conscious level is denoted “KL,” the unconscious functions are in the “PML” layer, while the motor commands and vision request reside in the “SAL” layer. The layered behavioral structure implemented in Freddy allows for future expansion of the robot behaviors. Additional high-level behaviors or unconscious functions can be added without having to redesign all the robot software.

Consciously, an entity is represented by a concept at the KL representing the entity as Freddy thinks of it. At the PML, however, Freddy can only have a sensory impression of the entity. Therefore, KL target concepts are aligned with descriptions, where a description is implemented as a list of the color of the entity and its shape. For example, description of John is BLUE SQUARE.

As shown in Figure 3, for each primitive action at the KL, there is a corresponding PML behavior. Each behavior starts up several concurrent behaviors at the unconscious level that might be consciously inaccessible. For example, a behavior that is concurrent with most other behaviors is visual obstacle detection and aversion.

Behaviors are implemented in a production style system we call PMA [Hexmoor 96]. Command-Transitions (CT) are the production rules which are transformation functions that map situation/behavior pairs into commands for execution.

Issuing a “Stop” command halts all ongoing robot processes, essentially rendering Freddy in an idle state. First, the motors driving the robot are shut down. Next, the vision processes are halted. If Freddy is currently tracking a target, then head tracking and the transmission of target updates to the robot body are terminated. If Freddy is currently in obstacle avoidance mode, then this behavior is terminated instead.

Asking Freddy to “Find” a target of a particular color and shape causes a cascade of primitive behaviors that carefully interact with one another. First, the target’s color and shape are gathered from Freddy’s iconic, short-term memory. This information is transmitted to Vision in order to “prime” it to a given color and shape. Vision then searches its current field-of-view and attempts to find an object matching the request. If the target is found, then its description (including color, shape, distance, pan angle, tilt angle, and size of the bounding box) is transmitted back to the robot. If Freddy receives an affirmative target detection

message, then a tracking request is immediately transmitted to Vision, which in turn begins issuing pan, tilt, and verge commands to the robot head motors to keep the target centered in the field-of-view. In addition to the tracking commands, Vision sends target update messages to the robot at a rate of approximately 5Hz to inform Freddy as to the target’s current position. Therefore, if Vision loses sight of the target, Freddy will be informed of this fact and can invoke measures to reacquire it.

If Freddy fails to receive target detection or target update message from Vision, then a scanning behavior is initiated. First, all robot motors are commanded to stop as a safety precaution (i.e., we want to make absolutely sure all translation and turret motors are not being driven by old velocity commands). After the robot has come to a stop, the pan and tilt parameters describing the scan are transmitted to Vision, which begins sending motor commands to the head. The scanning behavior is a pan maneuver from -45 degrees to +45 degrees at a -20 degree tilt angle, broken into discrete intervals. Each time the robot head stops, Vision searches the scene and adds the target detections to a growing list. At the completion of the scan, the list is sent back to Freddy where it is parsed for a target matching the desired color and shape. If the target is found in the list, then the vision process described in the preceding paragraph (i.e., prime, detect, track) is continued. Otherwise, the head will scan with a -40 degree tilt angle to detect targets near the base of the robot. If the target is still not found, the robot’s turret is turned 90 degrees to the left in order to scan a different area. This scanning behavior is continued until either the target is found, the user enters a new target description, or the user types a “Stop” command.

Asking Freddy to “Wander” will put the robot in obstacle avoidance mode and cause it to wander the room without crashing into obstacles. Since no goal is specified, Freddy will continue to wander until another command is issued or it runs out of battery power. Upon receiving the “Wander” command, Freddy first notifies Vision to enter its obstacle avoidance mode. Vision stops all tracking behaviors and places the robot head in a calibrated position (0° pan, -40° tilt, 0° verge) so that it can identify obstacles and determine their distance without extensive computation. When the head is in position, Vision begins finding obstacles and transmitting reports to Freddy at a rate of 8Hz. These obstacle update reports are gathered by the robot and used to steer Freddy around obstacles and through safe, narrow passages.

When moving, Freddy checks the five forward-most facing sonars on the turret. Since the visual system only sees colored objects, the sonars are required to detect the white walls in our robot lab. If any of the central three sonars report a detection less than 20 inches from the robot, then forward robot motion is halted immediately and Freddy rotates until both the sonars and Vision detect a clear path.

If no obstacles are in front, then Freddy will move forward with a velocity of 5 inches/sec. Detections by the two peripheral sonars effect a rotation away from the obstacle. The speed of the rotation is a function of the obstacle's proximity. Since the motors are driven by velocity control, the forward motion and turning behaviors are smooth; Freddy stops moving only when an obstacle appears directly in front of it or the sonars detect a wall.

This rule-based obstacle avoidance algorithm is tuned so that Freddy can navigate through narrow passages created by obstacles. Due to the discrete nature of the visual obstacle detection algorithm and the position of the cameras on the robot, objects close to the base of Freddy and to the sides are difficult to detect. Therefore, Freddy will occasionally brush low obstacles such as the colored shapes we use as targets. Increasing the margin of safety in the avoidance algorithm will prevent this brushing, but also eliminate Freddy's ability to navigate narrow gaps. We favor the existing design since hardware modifications may be the only necessary step to prevent occasional bumping of objects. Extending the cameras slightly ahead of the robot chassis will improve Freddy's low, side vision; moving the sonar ring lower on the chassis will also detect objects that Vision may have missed.

The "Go" command causes Freddy to find a specified target and then drive toward it while avoiding obstacles. It is implemented as a concurrent combination of "Find," "Wander," and an unconscious behavior called "Assess situation". The latter behavior is responsible for synchronizing "Find" and "Wander," remembering the target's position in the world relative to Freddy, and biasing the obstacle-free paths chosen by "Wander" so that the robot's movements drive it toward the target.

After "Go" has been issued, "Assess situation" waits until "Find" is successfully tracking the target. Then, the position of the target (which is reported by the vision system in terms of pan, tilt, and turret angles as well as distances) is converted to an (x,y) position relative to the front of the robot and stored in the short term memory. After this, "Assess situation" rotates the robot body to face the target, places Vision in obstacle

avoidance mode, and enables Freddy's "Wander" behavior. Freddy then begins moving toward target while avoiding obstacles. As Freddy moves around, "Assess situation" constantly updates the target's (x,y) position relative to the robot. This serves two purposes: a) Freddy always knows where the target is located even if the Vision loses sight of it, and (2) the relative (x,y) position is used to bias the paths chosen by the "Wander" behavior. After each position computation, "Assess situation" checks to see if the robot is within 30 inches of the target. If the robot is closer than this threshold, "Assess situation" will halt all motion, notify the user, rotate Freddy so that it faces the target, and set the vision system to visually track the target. Otherwise, the robot is allowed to continue its "biased" wandering.

"Assess situation" periodically reacquires the target. This is necessary when error in the robot's position feedback integrates into a value so large that the target is no longer in the field-of-view. An anxiety measure is used to trigger the target reacquisition process. If the robot's anxiety gets too high "Assess situation" will halt the robot motion; turn off the "Wander" behavior; convert the relative position to pan, tilt, and turret angles; and turn the robot body and head toward the supposed target location. Finally, "Find" behavior is activated and Freddy continues as described earlier. The sequence of actions keeps Freddy fixed on the target, preventing it from wandering too far away as it attempts to navigate around obstacles.

5. Summary of Lessons Learned from Freddy

An internal model of time and a long term memory is needed for reasoning and answering queries about actions. Freddy understands the succession of events in "Come here and go help John." After execution, when asked "What did you do?" using proper tenses it responds with "I came to you. Then I went to help John."

Short term memory is needed in coordination of activities that time-share sensory apparatus. Freddy needed to remember direction of motion to target when it used vision to avert obstacles.

Instruction giving is more powerful and natural when pointing is allowed. This is called deixis. By visually disambiguating the referent, Freddy is commanded to "Come here", "Go there" or "Find that green robot".

By providing relatively independent and concurrent layers of interaction with the world, robot interaction can be more timely. Freddy interacts at the reflex level, at the reactive level with quick responses, and at the

deliberative level with reasoning about communicating about actions.

6. Principles that Help Standardize Integration

An architectural principle is grouping of loci for associating sensing to acting. This has resulted in developing three layer architectures [Hexmoor, et al 97]. There are variations in this principle but these architectures provide loops of interaction with the world with different time delays.

A key to integration is a high degree of coherence and a low degree of coupling in software modules. Maintaining pedigree of data and control best performs interface among modules. In humans and animals, pedigree means managing memories of time, space, states of behavior, and intentions. We believe distinguished pedigree mechanisms and centers are required.

An issue that impedes the integration/interface among behaviors and systems is the lack of metrics. If we cannot measure a behavior or a system or quantify its parameters, we cannot expect to easily reuse it. Metrics is an open problem and deserves attention.

References

[Hexmoor, et al 97] **Software Architectures for Harward Agents**, Special issue of JETAI, H.

Hexmoor, D. Kortenkamp, and I. Horswill (editors), Vol 9, NO 2,3, 1997.

[Hexmoor, et al 93] Henry Hexmoor and Johan Lammens and Guido Caicedo and Stuart C. Shapiro (1993). Behavior Based AI, Cognitive Processes, and Emergent Behaviors in Autonomous Agents, *Applications of AI in Engineering VIII*, Vol.2, G. Rzevski and J. Pastor and R. Adey, CMI/Elsevier, pp. 447—461.

[Hexmoor 96] Henry Hexmoor (1996). Learning Routines, *Intelligent Agents II: Agent Theories, Architectures and Languages* M. Wooldrige and J. Mueller and M. Tambe (eds), Springer Verlag, pp. 97—110.

[Bandera, et al 89] Cesar Bandera and Peter D. Scott, Foveal Machine Vision Systems, *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, Cambridge, MA, November 1989.

[Bandera, et al 96] Cesar Bandera, Francisco Vico, Jose Bravo, Mance Harmon, and Leemon Baird, Residual Q-Learning Applied to Visual Attention, *Proceedings of the 13th International Conference on Machine Learning*, Bari, Italy, July 3, 1996.

[Pashler 93] Harold Pashler (1993). Doing Two Things at the Same Time, *American Scientist* Vol. 81, No. 1, pp 48-55, Sigma Xi, The Scientific Research Society.