# A Reactive Behavioral System for Intelligent Spaces

**Ajay Kulkarni** KULKARNI@AI.MIT.EDU
MIT Artificial Intelligence Laboratory, 200 Technology Square, Cambridge MA, 02139 USA

## 1. Introduction

Inherent in the design of an Intelligent Space for Project Oxygen is the ability for the room to react to user behavior: anthropocentric computing requires human-computer interaction. This paper extends the work of the Metaglue System (Coen et al., 2000) in the Intelligent Room. The existing research in behaviors in the Intelligent Room has involved the establishment of a software and physical architecture to allow the Room to respond to specific user actions and commands on a 1-1 correspondence. If the Room were an intelligent robot, the past research can be likened to the creation of a "body" that can react to external commands. New research needs to internalize these commands; the Intelligent Room needs a "brain." This paper proposes a system to provide this reactive behavior.

## 2. Design Criteria

The following criteria are important for the design of a reactive behavior system for the Intelligent Room:

1. The Room should be able to exhibit a straightforward behavior: e.g., reaction to user entry.
2. Depending on the presence of specific conditions, the Room should be able to exhibit another behavior for the same input: *e.g., reaction to user entry while a movie is playing.* There can be several layers of conditional behaviors, each overriding the one below it. Adding these layers should make the Room more robust (as it can handle more sitations); adding the layers should not significantly slow down the Room's ability to react.
3. The presence of these conditions should be detected by the Room through audio, visual, other sensory, or user cues: *e.g., a camera detecting a movie is playing and/or a user telling the system that a movie is playing.* Other agents may also provide knowledge of the current conditions: *the DVD agent can tell the lights agent that it is playing a movie.*
4. To handle the situation of multiple cues presenting conflicting conditions, each information cue should be accompanied with a probability of its validity.
5. These overall conditional behaviors should be modular: can be implemented in some rooms, but need not be in others. *For example, the movie conditional behaviors are not important if you can not view a movie in the Room.*
6. Adding a behavior to the overall reactive system should be dynamic and not require to stop and restart the Room.

These design criteria are discussed in further detail in the next section.

## 3. Straightforward and Conditional Behavior

### 3.1 Limitations of Straightforward Behavior

Currently, the Intelligent Room can exhibit straightforward behaviors to specific user inputs. Imagine the following simple behavior: User enters Room; Room detects entry through visual or other sensory cues; Room turns on the lights.

Such a behavior can already be performed – and is performed – with the current level of development through an explicit set of rules.

The faults of such explicit behaviors are apparent: under certain circumstances the lights in the Room would be turned off purposefully, and user entry should not turn them back on. For example, when the current users in the Room are listening to a slideshow presentation or are watching a movie, turning on the lights on user entry would disturb the presentation or movie. But under special circumstances – e.g., someone important enters the room – the lights should turn on despite the presentation or movie.

Again, these explicit behaviors can be performed with the current level of development through an explicit set of overriding rules. This design would be similar to the **Subsumption** approach (Brooks, 1991), where higher levels of behavior are subsuming lower levels. This approach of explicitly hardcoding each situation brings forth scalability concerns. The nature of the Intelligent Room subjects it to

a variety of situations when a user might enter. Checking the conditions for each specific scenario each time a user enters the Room introduces a great amount of logical reasoning. As the robustness of the Room is proportional to the number of situations is can handle, the more robust the Room became, the longer it would take to turn on the lights when a user entered the Room. Clearly, explicitly describing rules is not an optimal approach for this problem.

### 3.2 Better Idea: An Agent-Driven Behavioral System

A better solution can be drawn from previous work in Self-Configuring Space Systems. A spacecraft system requires a reactive behavior that can withstand the hostile space environment. **Livingstone** (described in (Williams & Nayak, 1996)) describes a mode-transitions model with mode identification and reconfiguration that can be used for guidance.

The current implementation of the Reactive Behavioral System satsifies most of the design criteria mentioned earlier. In this system, a mode is represented by a **Behavior**, which is an organized collection of rules. A rule, in turn, is represented by an **Action** in a Behavior. When information about the state of the Room is passed to the Behavior, its Actions decides what do with that information. Each Behavior is an independent Agent. This autonomy allows for modular Behaviors that can be installed in some Rooms but not others.

A Behavior is connected to other behaviors through **child-parent relationships** and **action-dependencies**. A child-parent relationship exists between two Behaviors if the Room can transition from one Behavior (parent) to the other (child). An action-dependency exists between two Behaviors if an Action of one Behavior can override the Action of another behavior. The **Dependency Diagram** in figure 1 displays a example set of Behaviors and their connections. In this diagram, dashed-arrows denote child-parent relationships, and solid-arrows denote action-dependencies.

Note that a connection stems from a child to the parent; the child has knowledge of the parent's existence, but not vice-versa. This quality of inter-Behavioral connections, combined with a readily modifiable Dependency Diagram, facilitates the addition of new Behaviors to an already running Room.

While the diagram suggests a very complicated structure of Behaviors – and some seemingly contradictory connections – at any given time, not all of these connections are active. These relations suggest *potential* connections. Depending on the series of events in the Room, different Behaviors and connections will become active, allowing for a hierarchical structure of layered Behaviors. (Please see Figure 2 for an example.)

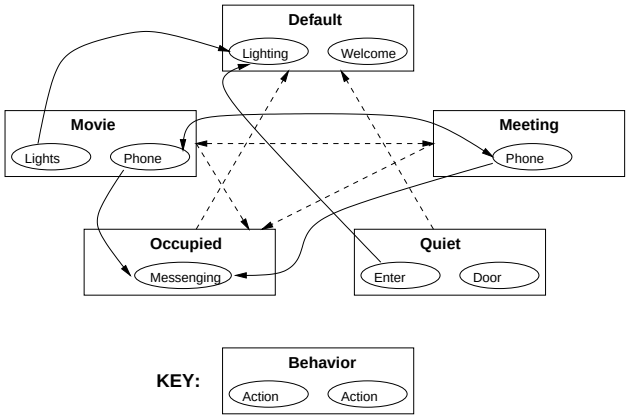


*Figure 1.* Dependency Diagram

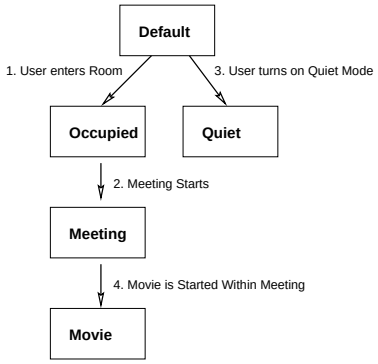


*Figure 2.* Active Behavior Hierarchy

The crux of the Reactive Behavioral System is embodied in the **Behavior Coordinator**. While a Behavior knows about its own state and about the state of its parents, the Behavior Coordinator maintains the state of all Behaviors. Specifically, the Behavior Coordinator serves as the connection between the Behaviors and the external **Perception Engine**. An event sent from the Perception Engine is received by the Behavior Coordinator, which then passes the event to the active Behaviors.

## References

Brooks, R. (1991). Intelligence without reason. Computers and Thought, IJCAI-91.

Coen, M., Phillips, B., Warshawsky, N., Weisman, L., Peters, S., & Finin, P. (2000). Meeting the computational needs of intelligent environments: The metaglue system. In submission.

Williams, B. C., & Nayak, P. P. (1996). A model-based approach to reactive self-configuring systems. Recom Technologies, NASA Ames Research Center, MS 269-2.