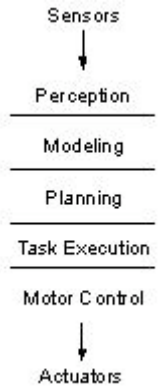
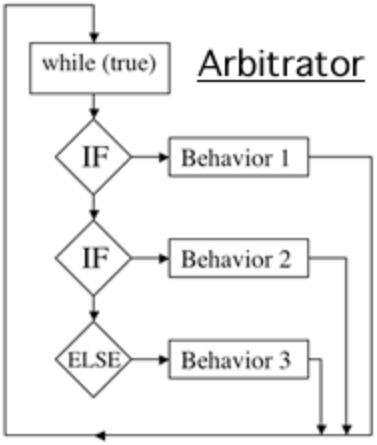
Robot Control Architecture



Sense-Plan-Act

This consists of three linear steps.

1. Sense the environment.
2. Plan what to do next by building a world model through sensory input. Consider all goals both short term and long term.
3. Execute the plan through the actuators.



Subsumption architecture is a form of robot control in which the control is divided into layers corresponding to levels of behavior. The idea of subsumption is that not only do more complex layers depend on lower, more reactive levels, but that they could also influence their behavior.

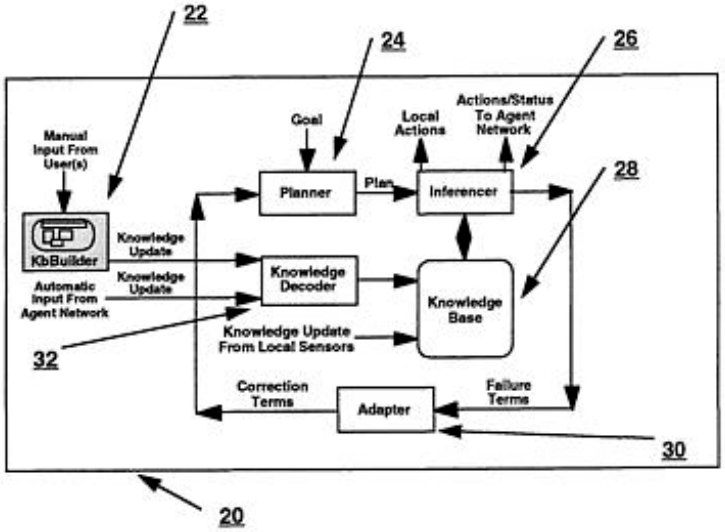
Within subsumption architecture, the controlling structure is an arbitrator. The arbitrator looks through a list of behaviors, and depending on the current conditions, will fire off a certain behavior.

Behavior generally refers to the actions or reactions of an object or organism, usually in relation to the environment or surrounding world of stimuli. In our case, we are going to want to have behaviors that fire depending on information that we receive in from our sensors. Therefore, we can brake down our behaviors into two parts, the conditions that determine whether or not to fire a certain behavior, and the actions to take if those conditions are satisfied.

From www.mcs.alma.edu/LMICSE.

Example of a ***Multiple-Agent Hybrid Control Architecture***
designed by SAGENT CORPORATION, published June 6, 1997

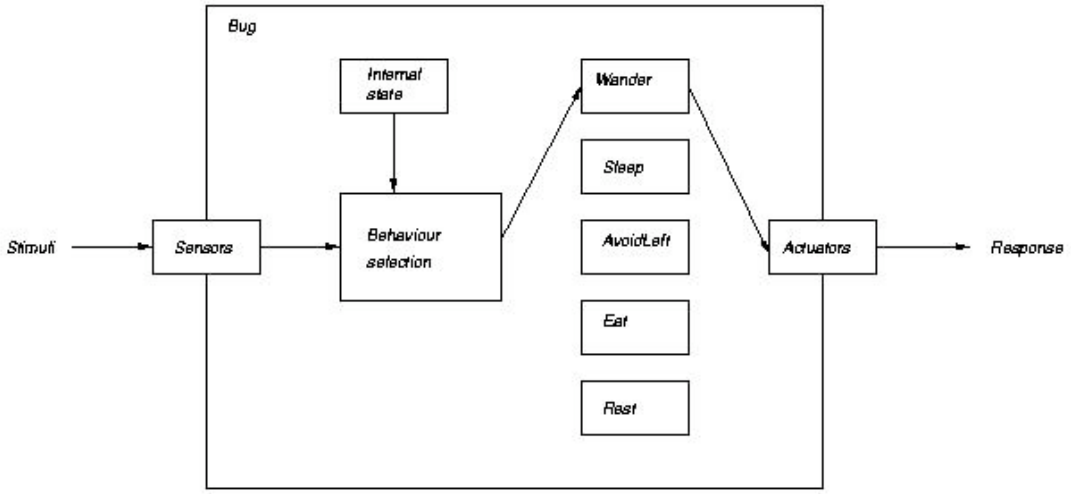
"A Multiple-Agent Hybrid Control Architecture (MAHCA) uses agents to analyze, design, and implement intelligent control of distributed processes. A single agent can be configured to control a local process and a network of agents can be configured to control more complex distributed processes. Each agent includes a Knowledge Base Builder (22) which supports manual input from users, a Planner (24) which generates a statement representing the desired



behavior of the system as an existentially quantified logic expression, an Inferencer (26) which determines whether this statement is a theorem currently active in the Knowledge Base (28) and if it does the Inferencer generates the current control action schedule, an Adapter (30) which replaces or modifies statements that do not follow from the current status of the Knowledge Base, and a Knowledge Decoder (32) which supports automatic input from other agents. Multiple agents interact through messages and can be either permanent or temporary."

Example of a ***Behavior-Based Control Architecture***
From http://legolab.daimi.au.dk/Projects/JungleCube.dir/Chapter.html

"An investigation of robots as a medium for artistic expression started in January 2000. As a result, insect-like LEGO robots, Bugs, have been created, that through movements and sounds, are able to express what to an observer seem like emotions, intentions, and social behaviour."



"These initial experiments led us to use a behaviour-based architecture for the control program of the Bug, Brooks (1986, 1991), Mataric (1992, 1997). In Figure 10.7 the Behaviour-based architecture is shown. Each Bug has a repertoire of different behaviours: Wander, Sleep, Avoid-Left, etc. Each behaviour consists of two independent program modules, a locomotion module and a sound module. When executed, each module generates a sequence of simple instantaneous actuator actions. Locomotion modules start or stop the motors, sound modules start or stop one or both oscillators that control the speaker. At any given instant only one behaviour is active. The two modules of the active behaviour are executed concurrently: the locomotion module moves the Bug around, while the sound module generates the accompanying sounds. Triggered by external stimuli like a bumper being pushed, or triggered by changes in the internal state, e.g., an avoiding behaviour that finishes, the active behaviour is terminated and a new behaviour is selected to be the active one. Hence, the behaviour selection depends on external stimuli and internal state as shown in Figure 10.7. The selection mechanism itself is based on the concept of motivations, Krink (1999). At any given instance, each behaviour is associated with a motivation value. The behaviour with the highest motivation value is selected to be the active behaviour. Motivation values are calculated every 50 msec by means of motivation functions, one for each behaviour. The motivation functions use sensor values resulting from external stimuli, and the internal state to calculate a motivation value: during the day the motivation value for the behaviour Wander is high, during the night it is low. The opposite applies to the motivation value for the behaviour Sleep; the motivation value for Avoid-Left is higher than the value for Wander or Sleep when the left bumper is activated and the value stays high until the Avoid-Left behaviour ends. One of the consequences of this behaviour selection mechanism is that the reaction to obstacles is delayed at most 50 msec and is perceived as immediate reaction."