

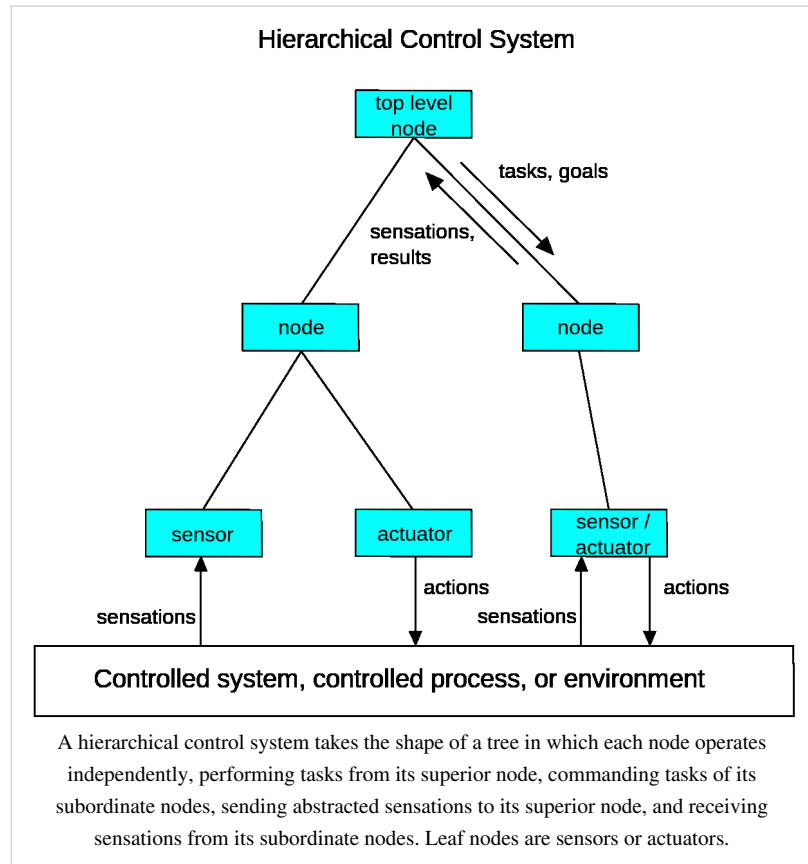
Hierarchical control system

A **Hierarchical control systems** is a form of control system in which a set of devices and governing software is arranged in a hierarchical tree. When the links in the tree are implemented by a computer network, then that hierarchical control system is also a form of networked control system.

Overview

A human-built system with complex behavior is often organized as a hierarchy. For example a command hierarchy has among its notable features the organizational chart of superiors, subordinates, and lines of organizational communication. Hierarchical control systems are organized similarly to divide the decision making responsibility.

Each element of the hierarchy is a linked node in the tree. Commands, tasks and goals to be achieved flow down the tree from superior nodes to subordinate nodes, whereas sensations and command results flow up the tree from subordinate to superior nodes. Nodes may also exchange messages with their siblings. The two distinguishing features of a hierarchical control system are related to its layers.^[1]



- Each higher layer of the tree operates with a longer interval of planning and execution time than its immediately lower layer.
- The lower layers have local tasks, goals, and sensations, and their activities are planned and coordinated by higher layers which do not generally override their decisions. The layers form a hybrid intelligent system in which the lowest, reactive layers are sub-symbolic. The higher layers, having relaxed time constraints, are capable of reasoning from an abstract world model and performing planning. A hierarchical task network is a good fit for planning in a hierarchical control system.

Besides artificial systems, an animal's control systems are proposed to be organized as a hierarchy. In perceptual control theory, which postulates that an organism's behavior is a means of controlling its perceptions, the organism's control systems are suggested to be organized in a hierarchical pattern as their perceptions are constructed so.

Applications

Manufacturing, robotics and vehicles

Among the robotic paradigms is the hierarchical paradigm in which a robot operates in a top-down fashion, heavy on planning, especially motion planning. Computer-aided production engineering has been a research focus at NIST since the 1980s. Its Automated Manufacturing Research Facility was used to develop a five layer production control model. In the early 1990s DARPA sponsored research to develop distributed (i.e. networked) intelligent control systems for applications such as military command and control systems. NIST built on earlier research to develop its Real-Time Control System (RCS) and Real-time Control System Software which is a generic hierarchical control system that has been used to operate a manufacturing cell, a robot crane, and an automated vehicle.

In November 2007, DARPA held the Urban Challenge. The winning entry, Tartan Racing ^[2] employed a hierarchical control system, with layered mission planning, motion planning, behavior generation, perception, world modelling, and mechatronics.^[3]

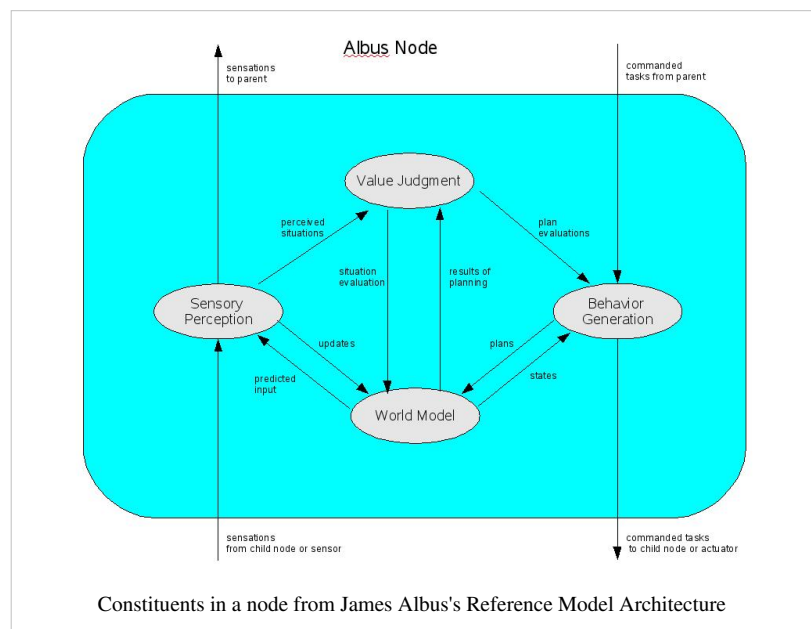
Artificial intelligence

Subsumption architecture is a methodology for developing artificial intelligence that is heavily associated with behavior based robotics. This architecture is a way of decomposing complicated intelligent behavior into many "simple" behavior modules, which are in turn organized into layers. Each layer implements a particular goal of the software agent (i.e. system as a whole), and higher layers are increasingly more abstract. Each layer's goal subsumes that of the underlying layers, e.g. the decision to move forward by the eat-food layer takes into account the decision of the lowest obstacle-avoidance layer. Behavior need not be planned by a superior layer, rather behaviors may be triggered by sensory inputs and so are only active under circumstances where they might be appropriate.^[4]

Reinforcement learning has been used to acquire behavior in a hierarchical control system in which each node can learn to improve its behavior with experience.^[5]

James Albus, while at NIST, developed a theory for intelligent system design named the Reference Model Architecture (RMA),^[6] which is a hierarchical control system inspired by RCS. Albus defines each node to contain these components.

- *Behavior generation* is responsible for executing tasks received from the superior, parent node. It also plans for, and issues tasks to, the subordinate nodes.
- *Sensory perception* is responsible for receiving sensations from the subordinate nodes, then grouping, filtering, and otherwise processing them into higher level abstractions that update the local state and which form sensations that are sent to the superior node.
- *Value judgment* is responsible for evaluating the updated situation and evaluating alternative plans.
- *World Model* is the local state that provides a model for the controlled system, controlled process, or environment at the abstraction level of the subordinate nodes.



At its lowest levels, the RMA can be implemented as a subsumption architecture, in which the world model is mapped directly to the controlled process or real world, avoiding the need for a mathematical abstraction, and in which time-constrained reactive planning can be implemented as a finite state machine. Higher levels of the RMA however, may have sophisticated mathematical world models and behavior implemented by automated planning and scheduling. Planning is required when certain behaviors cannot be triggered by current sensations, but rather by predicted or anticipated sensations, especially those that come about as result of the node's actions.^[7]

References

- [1] Findeisen, page 9
- [2] (<http://www.darpa.mil/GRANDCHALLENGE/Teams/TartanRacing.asp>) Tartan Racing team description
- [3] Urmson, C. et al., Tartan Racing: A Multi-Modal Approach to the DARPA Urban Challenge (http://www.darpa.mil/GRANDCHALLENGE/TechPapers/Tartan_Racing.pdf) 2007, page 4
- [4] Brooks, R. A. "Planning is just a way of avoiding figuring out what to do next" (http://www.ece.osu.edu/~fasiha/Brooks_Planning.html), Technical report, MIT Artificial Intelligence Laboratory, 1987
- [5] Takahashi, Y. , and Asada, M., Behavior Acquisition by Multi-Layered Reinforcement Learning. (<http://citeseer.comp.nus.edu.sg/rd/0.465821.1.0.25.Download/http://citeseer.comp.nus.edu.sg/cache/papers/cs/24389/http:zSzzSzwww.er.ams.eng.osaka-u.ac.jpzSzpaperszSz1999zSzTakahashi99d.pdf/behavior-acquisition-by-multi.pdf>) In Proceeding of the 1999 IEEE International Conference on Systems, Man, and Cybernetics, pages 716-721
- [6] Albus, J. S. A Reference Model Architecture for Intelligent Systems Design. (http://www.isd.mel.nist.gov/documents/albus/Ref_Model_Arch345.pdf) In Antsaklis, P.J., Passino, K.M. (Eds.) (1993) An Introduction to Intelligent and Autonomous Control. Kluwer Academic Publishers, 1993, Chapter 2, pp27-56. ISBN 0-7923-9267-1
- [7] Meystel, A. M., Albus, J.S., Intelligent Systems, John Wiley and Sons, New York, 2002, pp 30-31

Further reading

- Albus, J.S. (1996). "The Engineering of Mind" (<http://books.google.com/books?hl=en&lr=&ie=UTF-8&id=V3pksEEKxUkC&oi=fnd&pg=PA23&dq=The+engineering+of+mind&ots=sFjNGvuR8m&sig=Hrvf8uN732vQVBhgJjrohsUtfqI>). *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*. MIT Press.
- Albus, J.S. (2000). "4-D/RCS reference model architecture for unmanned ground vehicles" (http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=845165). *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on* **4**.
- Findeisen, W.; Others, (1980). *Control and coordination in hierarchical systems*. Chichester [Eng.]; New York: J. Wiley.
- Hayes-roth, F.; Erman, L.; Terry, A. (1992). "Distributed intelligent control and management(DICAM) applications and support for semi-automated development" (<http://www.csa.com/partners/viewrecord.php?requester=gs&collection=TRD&recid=N9317513AH>). NASA. Ames Research Center, Working Notes from the 1992 AAAI Workshop on Automating Software Design. Theme: Domain Specific Software Design p 66-70 (SEE N 93-17499 05-61). Retrieved 2008-05-11.
- Jones, A.T.; McLean, C.R. (1986). "A Proposed Hierarchical Control Model for Automated Manufacturing Systems" (<http://www.nist.gov/msidlibrary/summary/8606.html>). *Journal of Manufacturing Systems* **5** (1): 15–25. doi: 10.1016/0278-6125(86)90064-6 ([http://dx.doi.org/10.1016/0278-6125\(86\)90064-6](http://dx.doi.org/10.1016/0278-6125(86)90064-6)). Retrieved 2008-05-11.
- Stengel, R.F. (1993). "Toward Intelligent Flight Control" (<http://www.princeton.edu/~stengel/TIFC.pdf>). *IEEE Trans. Systems, Man, and Cybernetics* **23** (6): 1699–1717. doi: 10.1109/21.257764 (<http://dx.doi.org/10.1109/21.257764>).

External links

- The RCS (Realtime Control System) Library (<http://www.isd.mel.nist.gov/projects/rcslib/>)
- Texai (<http://texai.org>) An open source project to create artificial intelligence using an Albus hierarchical control system

Article Sources and Contributors

Hierarchical control system *Source:* <http://en.wikipedia.org/w/index.php?oldid=541926240> *Contributors:* Bwfrank, Chaosdruid, Destynova, Jiuguang Wang, Jncraton, Mdd, Pbroks13, Robbbb, StephenReed, 3 anonymous edits

Image Sources, Licenses and Contributors

Image:Hierarchical-control-system.svg *Source:* <http://en.wikipedia.org/w/index.php?title=File:Hierarchical-control-system.svg> *License:* Creative Commons Attribution 3.0 *Contributors:* user:Pbroks13

Image:Albus-node.jpg *Source:* <http://en.wikipedia.org/w/index.php?title=File:Albus-node.jpg> *License:* Creative Commons Attribution-Sharealike 3.0,2.5,2.0,1.0 *Contributors:* StephenReed, 1 anonymous edits

License

Creative Commons Attribution-Share Alike 3.0 Unported
[//creativecommons.org/licenses/by-sa/3.0/](http://creativecommons.org/licenses/by-sa/3.0/)