Unifying Undergraduate Artificial Intelligence Robotics: Layers Of Abstraction Over Two Channels*

Frederick L. Crabbe
Computer Science Department
United States Naval Academy
572C Holloway Rd Stop 9F
Annapolis, Maryland 21402

Abstract

From a Computer Science and Artificial Intelligence perspective, Robotics often appears as a collection of often disjoint, sometimes antagonistic sub-fields. The lack of a coherent and unified presentation of the field negatively impacts teaching, especially to undergraduates. The paper presents an alternative synthesis of the various sub-fields of Artificial Intelligence robotics, and shows how these traditional sub-fields fit in to the whole. Finally, it presents a curriculum based on these ideas.

Introduction

Modern Artificial Intelligence Robotics education treats the field as a collection of overlapping subfields. An examination of the current robotics textbooks (McKerrow 1991; Arkin 1998; Dudek & Jenkin 2000; Murphy 2000; Niku 2001, e.g.) indicates that these subfields are: traditional Planning based robotics, Behavior Based robotics, Probabilistic robotics, Mobile robotics, and Engineering robotics. Reading these texts gives the impression that each of these fields is overlapping, yet distinct, except for Engineering robotics, which most Computer Science/Artificial Intelligence instructors consider to be an entirely separate field.

This fragmentation of fields likely derives from the sometimes malignant relations between the fields as they competed for primacy. "The whole idea of plan execution and the runtime maintenance of something called a 'plan' is misguided." (Brooks 1986) "This development follows a much broader trend in mobile robotics, where probabilistic techniques are commonly the method of choice over more ad hoc approaches, such as behavior-based techniques." (Thrun 2002) While such competition is natural in a research setting, it makes it difficult to present these multiple fields coherently. Exacerbating the situation, Robotics instructors tend to borrow the tendency from Artificial Intelligence instructors of presenting topics chronologically according to historical development. This approach, while interesting to practitioners, fails to put the areas in their *technical* context.

This paper argues for a perspective that unifies all the competing Robotics sub-fields into a single framework for instruction. It results in a more natural order of topics and emphasizes their relations rather than differences. The organizing principle is that robotic systems are best understood as layers of abstractions over input and output channels.

Layers of Abstraction

The application of layers of abstraction in Computer Science is a well known technique used either prescriptively to coordinate standards development, or descriptively to make sense of complicated processes and ease comparison of apparently conflicting ideas. The classic example of the former is the ISO network layer system (International Organization for Standardization 1994) which specifies an organization for computer networking. On the other hand, layers of abstraction are used as a general guideline for the understanding of complicated systems throughout Computer Science and Engineering, such as the organization of computer hardware, operating systems or large pieces of software. This technique is also handy for understanding robotics systems.

Layers of abstraction are a natural way to characterize Intelligent Robotics, in which low level perceptions are converted to high level actions and back down to low-level motor movements. Every system does this by: processing multiple sensor inputs; combining the input into increasingly higher levels of abstraction until an action decision can be made; and breaking down the decision into increasingly more specific information until it can be executed as motor commands. Intelligent Robotics as a field is best seen as two information channels (input and output) crossing multiple layers of abstraction from physical signals to sophisticated symbols. All of the major paradigms fit into and can be interrelated by this paradigm. Figure 1 shows the framework presented here with two channels and the layers of abstraction through which information is processed. The top row is the input channel, starting with physical signals on the left, and passing through multiple abstractions as it moves to the right. The bottom row is the output channel, moving from high level symbols on the right to motor signals on the left. Each input channel can be made up of multiple pathways along which individual pieces of information can travel. For instance, input from multiple sensors would all be in the input channel, but until they are fused, they would be along multiple pathways. All of the common robot architectures can be see in this view as variations on how many layers of abstraction are passed through, and at which layer (or lay-

*This work was supported in part by a grant from the Office of Naval Research, N0001403WR20241

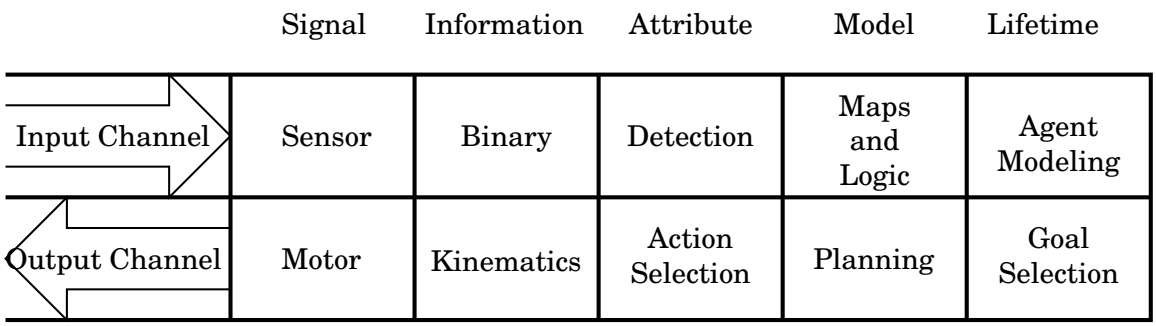

Figure 1: The Layers of Abstraction framework.

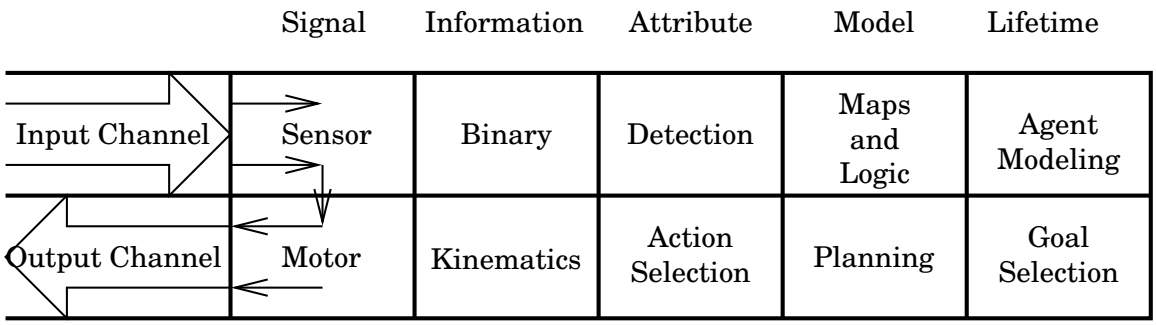

Figure 2: Information flow in Braitenberg Vehicles.

ers) information *crosses over* from the input channel to the output channel.

The Layers

Figure 1 describes five distinct layers. Because this model is descriptive there is some room for adjustment in both the numbers of layers and where they are divided. These particular layers were identified because they correspond to major robotic architectures in the literature and have proved useful in teaching.

The lowest layer is the Signal Layer. Information at this layer takes the form of electrical impulses from sensors and to motors. While all electronic robots have activity at this layer (otherwise they would never move), few perform any crossover from input to output at this layer. The classic exception to this is the set of Braitenberg Vehicles (Braitenberg 1984) which directly connect sensors to motors. Figure 2 shows the information flow in the framework of a Braitenberg Vehicle.

The next layer is the Information Layer, where input and output is converted between electric current and information on the robot and its environment. At this layer, in the input channel, the electrical signals are converted to bits in RAM. The input is also often processed to the extent that some relative position information is identified. In the output channel, the information layer performs kinematic analyses to convert desired physical positions into motor positions and thus electrical outputs. It is at this layer where most of what would be called Engineering robotics takes place (Figure 3). Sensor inputs are used the generate desired robot positions which are converted to motor movements, such as a camera directing an articulated arm to grasp an object.

In the attribute layer the input channel generalizes the information input by recognizing simple environmental states such as *obstacle-detected* or *goal-detected*. These often integrate sensor information over short periods of time from multiple sensors. In the output channel collections of possible actions are weighed, an action is selected, resulting in a desired physical position of the robot. The attribute layer is where most of the action takes place in Behavior-Based architectures (Figure 4). Collections of independent modules fuse input to recognize simple world states and make action recommendations, which are then selected from to result in a new desired robot position.

In the Model Layer, the input channel builds explicit models of the external world. The models built can have varying degrees of abstraction. For example, probabilistic occupancy grid maps are low level abstractions, as they make no attempt to cluster the occupied cells into objects. Topographical maps are higher level because they model the relations between objects and locations. Higher level abstract models are built describing the environment in terms of languages such as predicate logic. In the output channel is where planning takes place. The system reasons about the models to find a plan to satisfy a goal. Traditional planning based systems perform most of their computation at the model layer (figure 5). From the map models, low level planning (i.e. path planning) can occur. Often instead

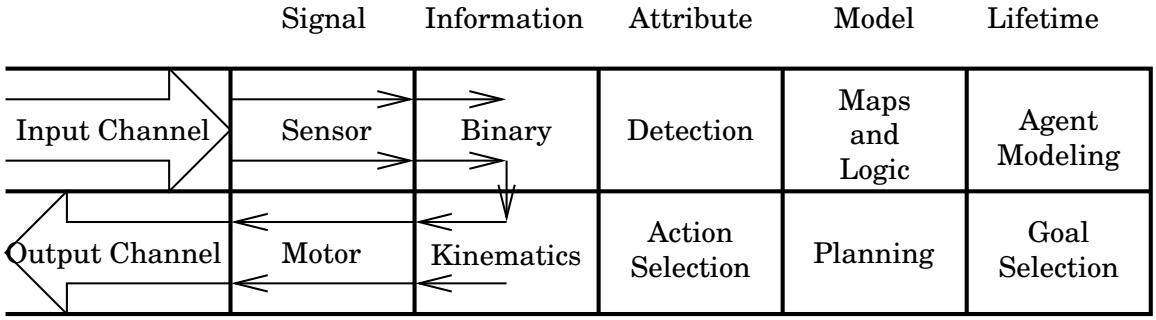

Figure 3: Information flow in Engineering robots.

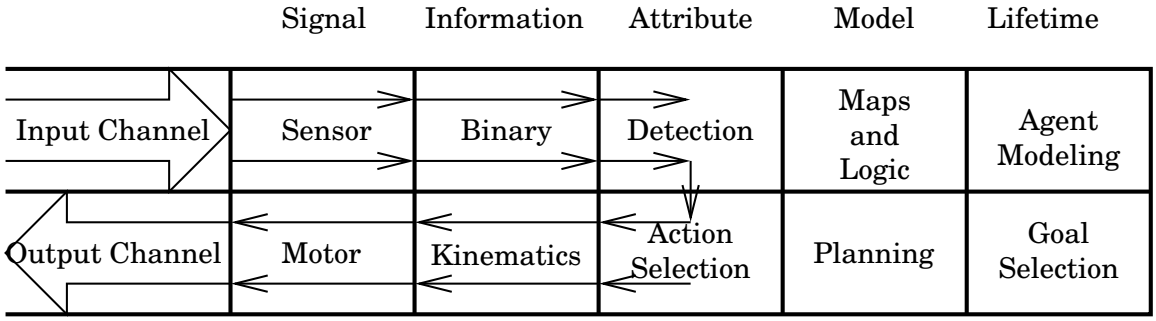

Figure 4: Information flow in Behavior-Based robots.

of path planning, the robot behaves reactively toward that model, skipping the planning part altogether (Figure 6).

Up to this point in all of the paradigms presented, input information is abstracted up to some layer where it crosses over to the output channel and converted to motor commands. There is no reason that information cannot cross from input channel to output channel at multiple layers. This is exactly what happens in the Planning/Behavior Hybrid systems. In these systems, longer term information is modeled and planned over, which is used to direct the reactive decisions being made in the attribute layer (figure 7).

In the top layer, the Lifetime Layer, decisions are made about the longer term behavior of the robot. It is here that other agents are modeled in order to coordinate collective behavior, and the robots are able to consider what tasks they will pursue over their lifetime.

In general, as we move from left to right in the input channel, the amount of state required in the layer increases, and the time window over which that state integrates also increases. For instance, when comparing the input processing at Attribute and the Model Layers, processing at the Attribute Layer is both less intense than at the Model Layer and it requires fewer input samples in order to generate output. The Model Layer needs to integrate many input samples before it can build a model suitable for use in decision making. At the same time, moving from left to right, the number of pathways used in the input channel decreases as information is consolidated into the abstractions. This process reverses as information travels from right to left down the output channel.

The Curriculum

The two channel layered model described above suggests a particular curriculum based of alternately presenting the workings of each layer followed by one or two example architectures that focus on that layer. This enables the students to implement an example of that architecture on a robot suited to that layer. We have developed this curriculum in a four hour per week robotics course taken primarily by Senior Computer Science majors who have not necessarily taken an Artificial Intelligence course. In this course we emphasize how each robot architecture relates to the others in the context of the layered model.

For the Signal Layer, we present the basic operation of the sensors and electric motors found on most robots. As an example robot architecture, we present Braitenberg's architecture as described in Vehicles, chapters one through five (Braitenberg 1984). The students can experiment directly with these ideas using simple commercial robots such as the BYO-bot (Miller 2003). The BYO-bot does provide some of the standard functions of the Braitenberg Vehicles, but are not programmed by the student directly connecting sensors and motors with wires, and are limited in the number of ways it can be programmed. To remedy this, we designed a similar low-cost robot that the students can program by connecting wires. The sensors and motors have pluggable connectors into which the students can insert wires to attach the sensors to the motors so that they resemble the diagrams

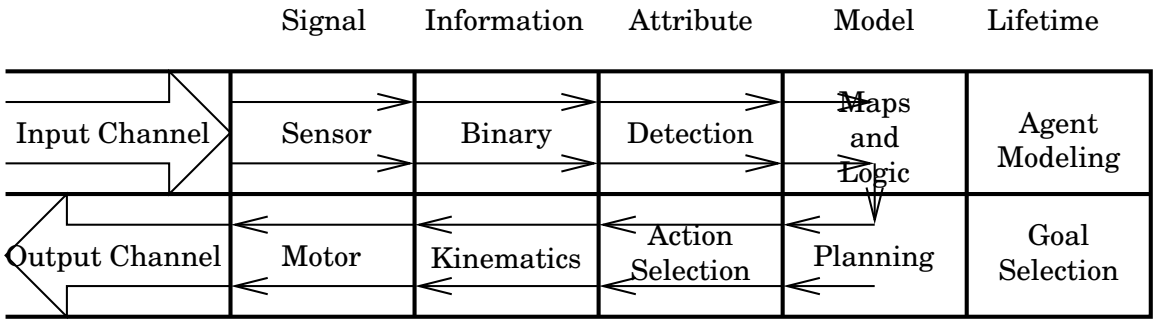

Figure 5: Information flow in Planning robots.

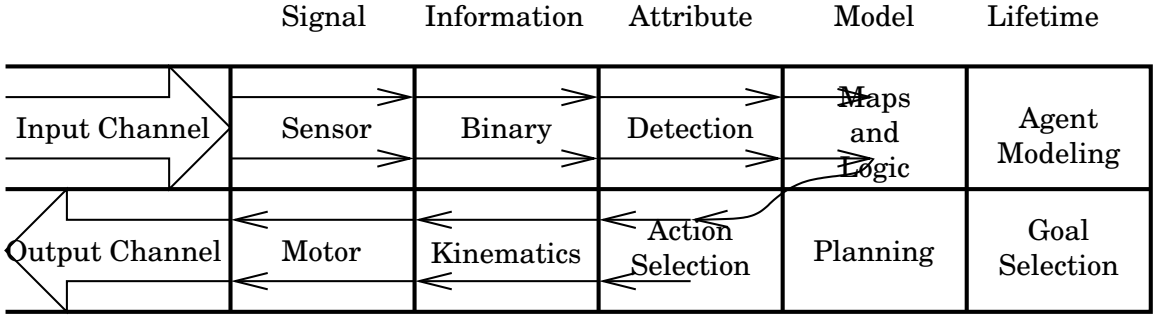

Figure 6: Information flow in Probabilistic robots using occupancy grid maps.

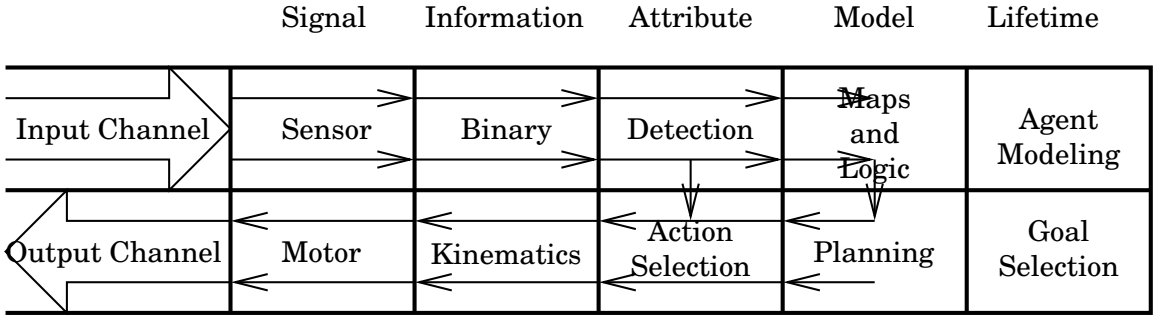

Figure 7: Information flow in hybrid systems.

in the Vehicles book. Our initial design included only direct connections, but designs in development now also include inhibitory connections and potentiometer knobs to adjust connection strength.

For the Information Layer, we present the mathematical models of Engineering Robotics, with emphasis on kinematics in the Denavit-Hartenberg system (Denavit & Hartenberg 1955) for rigid arms and extend it to the Sheth-Uicker system (Sheth & Uicker 1971) for mobile robots. The students perform both forward and inverse kinematics on both a rigid arm and a mobile robot. For the rigid arm we use the Robix Rascal (Advanced Design Inc. 2003) a rugged kit suitable when high accuracy is not required. Typical assignments are to pick up an object in a known location in the coordinate system. The students then carry this over to mobile robotics by implementing ded. reckoning on a Rug Warrior (A K Peters, Ltd. 2003). They implement a light follower, and then return to the starting position and report the location of the light.

In presenting the Attribute layer, we discuss basic techniques for identifying properties in input streams, sensor fusion, and action selection mechanisms. From this we present behavior-based architectures, emphasizing how they handle chaotic environments. Students implement a behavior-based system on the Rug Warriors for a competitive or cooperative game, such as capture-the-flag.

At the Model layer, we present the basics of a occupancy grid map and how they are built. This is followed by topographical maps, and how they could be built out of occupancy grids by identifying objects, and building Voronoi diagrams. Then we discuss logical representations and how they can be gathered both from properties in the identification layer and from the maps built in the lower portions of the model layer. For the output channel, we show how reactive modules in the attribute layer can use maps like the occupancy grids. We discuss how to build Voronoi diagrams, and use them to search for paths. Finally, we discuss planning and how that can use logical descriptions to solve hard problems in the world. Because many of the students have not yet taken Artificial Intelligence, we present the AI problems solely as finding paths in graphs and leave the algorithmic details to the AI course. The student's final project assignment is to use a Khepera robot (K-Team S.A. 2003) equipped with a gripper to move blocks. We give the students a map, a set of built in actions and the world state, and they solve small reasoning problems such as unblocking a doorway.

The final few classes are left to discussion of the lifetime layer, where the highest level operations take place. It is at this layer that much of the cutting edge research is taking place and thus it is difficult to give a clear picture of where these issues are headed. We typically concentrate multi-agent systems, the coordination of robots to perform a task, and how fully autonomous robots might select which goals they want to pursue and which they do not.

At the conclusion of the class we can re-emphasize that all of the styles of robotics share the same general architecture, and only vary on haw many levels of abstraction are applied and at what levels information crosses from the input to the output channel. We also point out that just like good software engineering practices, high levels of abstraction can reuse implementations of lower levels.

Discussion

The idea of abstraction in computing is not new, nor is the bottom up approach to teaching. However, by examining the texts currently available in robotics, it appears that undergraduate robotics is rarely taught in that manner. Textbooks that focus on robotics sub-field usually ignore all of the other sub-fields. Even books that have broad coverage, often fail to relate the sub-fields to each other. For each major sub-field in robotics, examining it from the perspective of this framework provides additional insight.

At the signal layer, Braitenberg architectures are often presented as an interesting but unrelated thought experiment in robotics. Instead, they fit naturally into the framework and make an excellent place to introduce robots.

Behavior-based robots have been traditionally presented as an antidote to and a radical departure from the traditional planning systems. According to this layered framework they are neither unrelated nor antithetical to planning systems; the are a difference of opinion on how much abstraction is necessary to perform various tasks. Furthermore, the framework highlights that hybrids of behavior-based and planning architectures are a natural combination of crossing information from the input to the output channel at multiple layers, thus taking advantage of the time differences at the various layers.

Mobile robotics is commonly presented as its own subfield, with its own kinematics and own high-level issues such as navigation. By examining mobile robotics in terms of the layer framework one can see that the system organizational issues for Mobile robotics are identical to those of other robotics systems. They can been seen as requiring slight modifications of details, such as using a variant on the kinematic formulations, but otherwise are the same as stationary articulated arms.

Advocates of Probabilistic robotics maintain that it can bestow the fast reaction time and robustness to noise benefits of Behavior-Based robotics in a more principled system. These systems are more robust than STRIPS planning systems because of the robustness that probability provides compared to predicates, but much of their robustness and fast reaction time is due to skipping the planning step altogether. Instead they use modules at the attribute layer that react to the probabilistic map rather than to the environment. Thus they *owe* some of their robustness to Behavior-Based architectures.

It is common to find discussion of AI planning robotics without any explanation of where the logic representation comes from, or how actions output by the planner end up as motor movements. It is similarly common to find discussions of the use of topographical maps without discussion of where they come from. In the layered model, these connections are emphasized at each stage, providing the student with the connections and context to understand these topics.

Many authors highlight differences in autonomy in various systems as a spectrum, from fully autonomous to tele-

operated. While strictly speaking the level of autonomy is orthogonal to the architecture, it is interesting to note that as autonomy increases, it is always introduced at the lowest layers of the system first. As systems become progressively more autonomous, they add autonomy layer by layer from the lower layers on the left to the higher layers on the right. By describing the level of autonomy in this manner, by showing *where* autonomy is added, it provides a more specific concept than just that of a spectrum.

Finally, it is interesting to note that because this frame of reference encompasses all of the standard robotics sub-fields it can be interesting to a researcher as an evaluation tool of the claims of novel architectures. By examining new proposals from the perspective of this framework, a researcher can determine relations to existing systems and evaluate uniqueness claims.

Conclusion

This paper presented and argued in favor of a perspective for teaching robotics that looks at the problem as a two-channel set of layers of abstraction. This perspective is useful for teaching an undergraduate course that focuses on the broad spectrum of subfields in robotics. It covers all of the major issues with the advantage that it unifies them, and puts them in context of each others, and removes possible skew from historical developments.

References

A K Peters, Ltd. 2003. Rug warrior pro. *http://www.akpeters.com/book.asp?bID=39*. Web page accessed Oct 2003.

Advanced Design Inc. 2003. Robix rascal. *http://www.robix.com/*. Web page accessed Oct 2003.

Arkin, R. 1998. *Behavior-Based Robotics*. Cambridge, MA: MIT Press.

Braitenberg, V. 1984. *Vehicles: Experiments in Synthetic Psychology*. Cambridge, MA: MIT Press.

Brooks, R. 1986. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation* RA-2:14–23.

Denavit, J., and Hartenberg, R. S. 1955. A kinematic notation for lower-pair mechanisms based on matrices. *Journal of Applied Mechanics* 22:215–221.

Dudek, G., and Jenkin, M. 2000. *Computational Principles of Mobile Robotics*. Cambridge, UK: Cambridge University Press.

International Organization for Standardization. 1994. ISO Standard 7498-1.

K-Team S.A. 2003. Khepera II robot. *http://www.k-team.com/robots/khepera*. Web page accessed Oct 2003.

McKerrow, P. J. 1991. *Introduction to Robotics*. Sydney: Addison Wesley.

Miller, D. P. 2003. Byo-bot. *http://www.kipr.org/robots/byobot.html*. Web page accessed Oct 2003.

Murphy, R. R. 2000. *Introduction to AI Robotics*. Boston: MIT Press.

Niku, S. 2001. *Introduction to Robotics Analysis, Systems, Applications*. Upper Saddle River, NJ: Prentice Hall.

Sheth, P. N., and Uicker, J. J. 1971. A generalized symbolic notation for mechanisms. *Journal of Engineering for Industry* 93(70-Mech-19).

Thrun, S. 2002. Robotic mapping: A survey. In Lakemeyer, G., and Nebel, B., eds., *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann.