
An Attempt to Generalize AI Part 2: Planning and Actions

By Paul Almond

6 March 2010

Website:

<http://www.paul-almond.com>

E-mail:

info@paul-almond.com

This is the second in a series of articles that will attempt to give an overview of how minds may work and how similar systems could be implemented in computers. The first article described a probabilistic, hierarchical modeling system intended to provide a general ontology. This article describes the use of this for planning actions. The approach to planning of actions makes planning a process occurring mainly in the hierarchical model itself. An evaluation function score is continually computed and encoded as input data for the hierarchy, just as conventional, external inputs are, the scores corresponding to bottom-level pattern instances. When an output is imminent, the different values for the output are tried, the hierarchy being updated appropriately, and predictions of a future input of the evaluation function score are obtained. It is shown that an approach like this allows learning. There is no explicit attempt to represent “consciousness” or a “self”. Instead, in humans, these are objects within the modeling system, constructed to explain previous behavior, and are only different from other objects in that they relate outputs with later outputs, and therefore relate to intentionality.

Table of Contents

1 Introduction	5
2 The Hierarchy	6
3 The Action Selection Process	9
3.1 The General Idea	9
3.2 More Detailed Description.....	9
3.2.1 Computation of Evaluation Function Score	9
3.2.2 Input of Evaluation Function Score.....	10
3.2.3 Selecting Actions.....	12
4 How it Works.....	17
4.1 The <i>real</i> planning process	17
4.2 Why the Action Selection Process is Needed	18
4.2.1 Problem 1: Initially, there is no history of suitable behavior.	18
4.2.2 Problem 2: Random drift would occur.	18
4.2.3 The Same Problem	19
4.2.4 How the Action Selection Process Gets the System Started	19
4.2.5 Learning as Modeling.....	20
4.2.6 How the Action Selection Process Prevents Random Drift	21
4.3 Why the Evaluation Function Score is Used as an Input	21
5 The Self.....	23
6 Chess, Computers and Humans	25
7 Conclusion.....	26
8 Bibliography	28

Table of Figures

Figure 1: Patterns and the Hierarchy.....	7
Figure 2: Input of Evaluation Function Score (EFS).....	11
Figure 3: Trying 0 as an Output Value.....	13
Figure 4: Trying 1 as an Output Value.....	15
Figure 5: The Self.....	24

List of Abbreviations

AI	artificial intelligence
EF	evaluation function
EFS	evaluation function score

1 Introduction

This article is the second in a series about artificial intelligence (AI) and how our own minds might work. The previous article, *An Attempt to Generalize AI - Part 1: The Modeling System*, should be read first and is available at <http://www.paul-almond.com/AI01.pdf>.¹

The previous article described a probabilistic, hierarchical modeling system, and how it could generate probabilistic predictions of future inputs/outputs from previous inputs/outputs. The hierarchy is based on *patterns*. Each pattern is a formally described set of *pattern instances*. This allows statistical predictions to be made about a pattern instance's pattern output with only partial knowledge about its inputs. An important goal of this hierarchical model is achieving a *general ontology*.

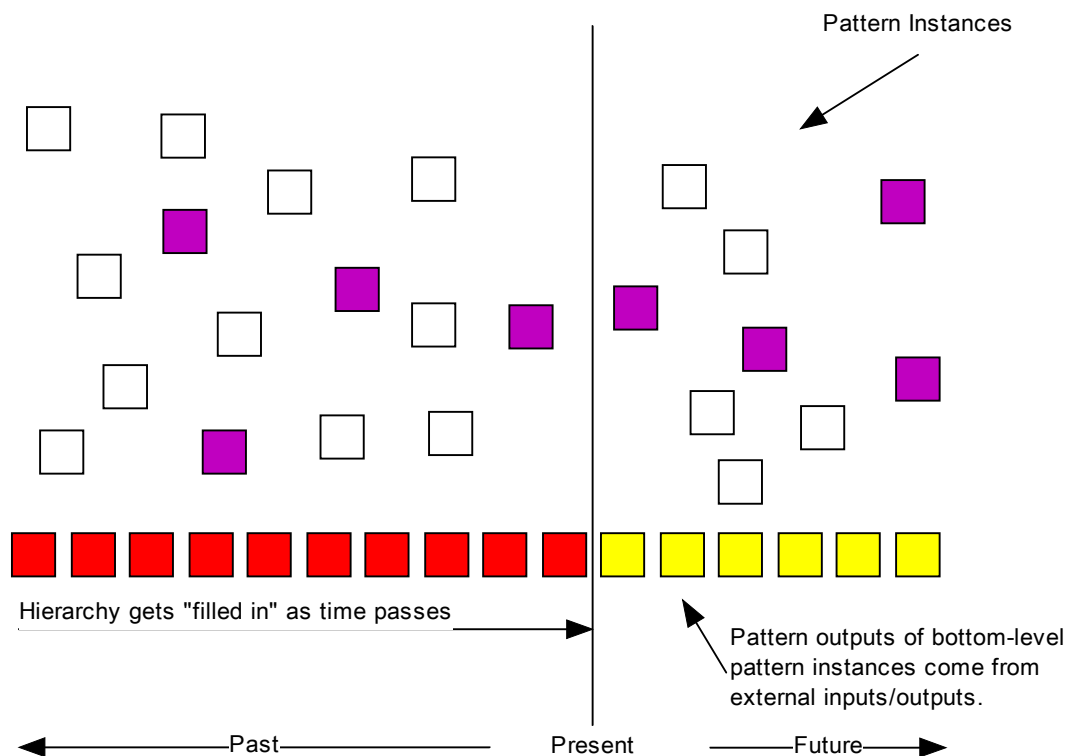
In that article, I only described how the system would perform modeling, and not how actions would be planned – how the system would decide what to do. An intelligent system needs to act intelligently, however, and modeling is useful only in as much as it informs action. In this article I will describe how the hierarchical modeling system from the previous article could be made to *act* intelligently. I am also intending both articles together to give an idea of how humans may model the world and plan actions.

The method given here is a basic method – the minimum needed to get the hierarchy planning actions – and might be improved on later.

¹ Almond, P. (2010). An Attempt to Generalize AI - Part 1: The Modeling System. *paul-almond.com*. <http://www.paul-almond.com/AI01.pdf>. (Also available at <http://www.paul-almond.com/AI01.doc>.)

2 The Hierarchy

The system described in the previous article is a hierarchy based on patterns. Each pattern is a set of pattern instances, distributed throughout the hierarchy and described by a pattern specification. The pattern specification consists of a logic specification, describing how each pattern instance of a pattern determines its pattern output from its labeled inputs, and a construction specification, describing how each pattern instance connects its labeled pattern inputs to the pattern outputs of other pattern instances. The bottom level of the hierarchy consists of special pattern instances, each corresponding to some external input/output event occurring at some instant. These pattern instances record the inputs/outputs that have occurred over a period of time. The rest of the hierarchy is built on top of this according to the pattern specifications of patterns. (See Figure 1 on page 7.)



Key

- Pattern instances of the same pattern. All these pattern instances are related in some way.
- Pattern instances of other patterns.
- Bottom-level pattern instances corresponding to external inputs/outputs that have already occurred.
- Bottom-level pattern instances corresponding to external inputs/outputs that have yet to occur.

Figure 1: Patterns and the Hierarchy

Because the pattern instances of a pattern are all related by being described by the same pattern specification, statistical observations of pattern instances with known pattern inputs can be used to assign probabilities to pattern instances with partially and probabilistically known pattern outputs. This can be used to propagate probabilities up the hierarchy to determine probabilities for high-level pattern instances, based on previous inputs/outputs, and down the hierarchy, to fill in the parts of the hierarchy corresponding to future inputs/outputs.

An Attempt to Generalize AI – Part 2: Planning and Actions

All this enables the hierarchy to predict future inputs/outputs probabilistically, based on previous inputs/outputs. This is going to be the main principle behind selection of actions.

3 The Action Selection Process

3.1 The General Idea

If we can predict future inputs/outputs, based on previous ones, planning becomes simple. Suppose we have some output to make, and we need to decide whether to make an output of 0 or 1. We can try each output value in turn and tell the hierarchy the output has actually happened with that value – even though it has not yet happened. We can then look at the probabilistic predictions for future inputs/outputs to see the kind of situation in which the system is in the future, and how desirable it is. We can then select the output value which causes the most desirable situation to be predicted, and actually make the output with that value.

To find out the desirability of a situation, based on predictions of inputs/outputs to the system, we could look at some inputs that correspond to desirable things – such as inputs showing that the system is receiving energy for example – and some inputs that correspond to undesirable things – inputs corresponding to threats of damage, such as excessive temperature, or pressure for example. A good way of doing this is to use an evaluation function (EF). The evaluation function examines the inputs/outputs occurring at approximately some instant of time and inputs them into an algorithm to generate an evaluation function score (EFS). By looking at the predictions for inputs/outputs, we could predict what the evaluation function score would be at some time in the future. However, we can do this in a more sophisticated way. If we continually compute the EFS, and provide it to the hierarchy as one or more inputs, by looking at a prediction for the inputs in the future corresponding to the evaluation function score, we would be obtaining the predicted EFS.

3.2 More Detailed Description

3.2.1 Computation of Evaluation Function Score

The evaluation function (EF) is continually used to compute the evaluation function score (EFS). Each time the EFS is computed, this is based on inputs which have occurred around the time that the score is computed.

e.g.

At $t=0.01s$, the EFS is computed based on inputs occurring just before $t=0.01s$.

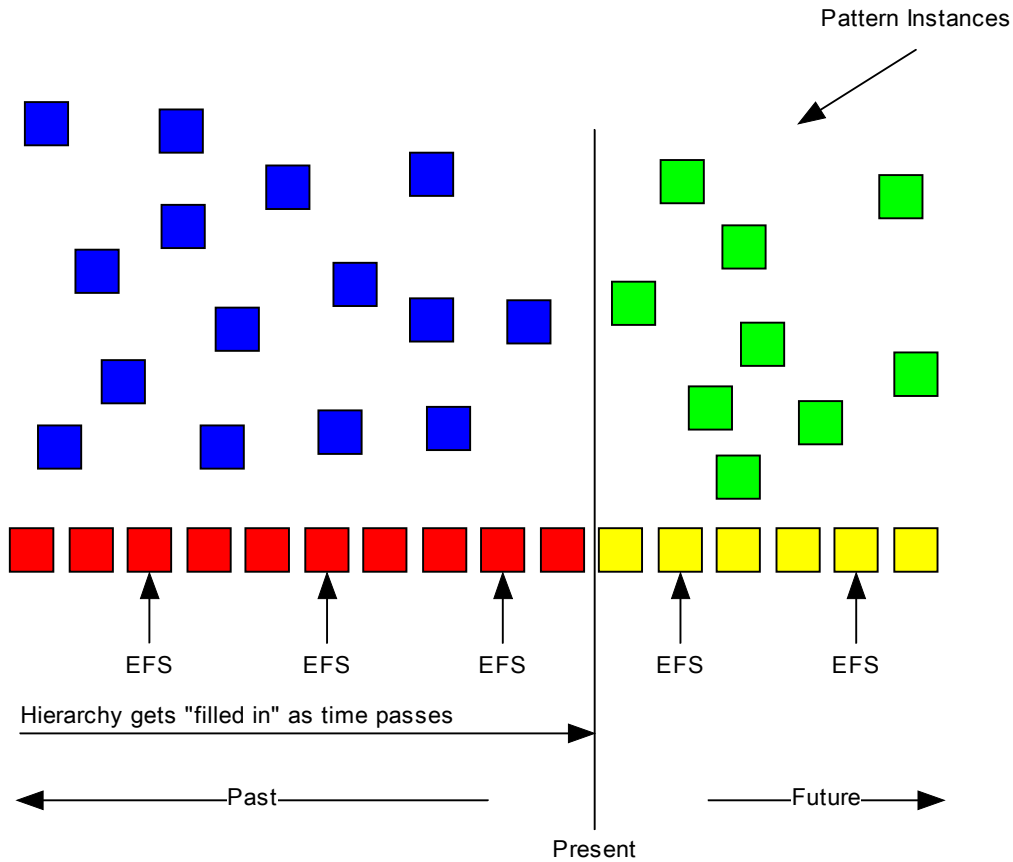
At $t=0.02s$, the EFS is computed based on inputs occurring just before $t=0.02s$.

At $t=0.03s$, the EFS is computed based on inputs occurring just before $t=0.03s$ and so on.

3.2.2 Input of Evaluation Function Score

Each time an EFS is computed, it is encoded as input data to the hierarchy and one or more pattern instances corresponding to external inputs at the bottom of the hierarchy are “fixed”: They are permanently assigned pattern output values encoding the most recently computed EFS.² (See Figure 2 on page 11.)

² A way of imagining the input of the desirability score comes from the movie, *The Terminator*, and its sequels. In this, a robot had computer generated information overlaid on its visual field, so that it would see information, presumably generated by its own computer systems, superimposed on objects that it was seeing. This meant that information generated by the software in the system was going back into the system as an input.



Key

EFS Input of evaluation function score into the hierarchy.

■ Bottom-level pattern instances corresponding to external inputs/outputs that have already occurred. Each of these is "fixed": Its pattern output value was assigned, permanently, when the relevant input/output event occurred.

■ Pattern instances which depend, directly or indirectly on bottom-level pattern instances for inputs/outputs which have already occurred. Each of these is also "fixed": Its pattern output value was assigned, permanently, when all the pattern instances being used for its inputs were assigned pattern output values.

■ Bottom-level pattern instances corresponding to external inputs/outputs that have yet to occur. Each of these has not yet been "fixed": The pattern output value will be assigned when the relevant input/output event occurs.

■ Pattern instances which depend, directly or indirectly on bottom level pattern instances for inputs/outputs that have yet to occur. Each of these has not yet been "fixed": Its pattern output value will be assigned, permanently, when all the pattern instances being used for its inputs have been assigned pattern output values.

Figure 2: Input of Evaluation Function Score (EFS)

3.2.3 Selecting Actions

Suppose some output, $Output_n$, has to be made at time, $t=t_1$. The output has two possible values: 0 or 1.

The expected effects of $Output_n=0$ are determined. To do this, the hierarchy is experimentally updated. The pattern instance corresponding to $Output_n$ is fixed such that $Output_n=0$. The effects of this are propagated through the hierarchy by logic application and statistics application. Probabilities for pattern instances dependent on future inputs/outputs will be affected and, ultimately, probabilities for pattern instances corresponding to future, external inputs will be affected. As the EFS values have been continually encoded as inputs, some of the pattern instances corresponding to future, external inputs will correspond to future input of the EFS. The probabilities for one or more inputs corresponding to future input of an EFS value are examined.

For some output, $Output_n$, occurring at time, $t=t_1$:

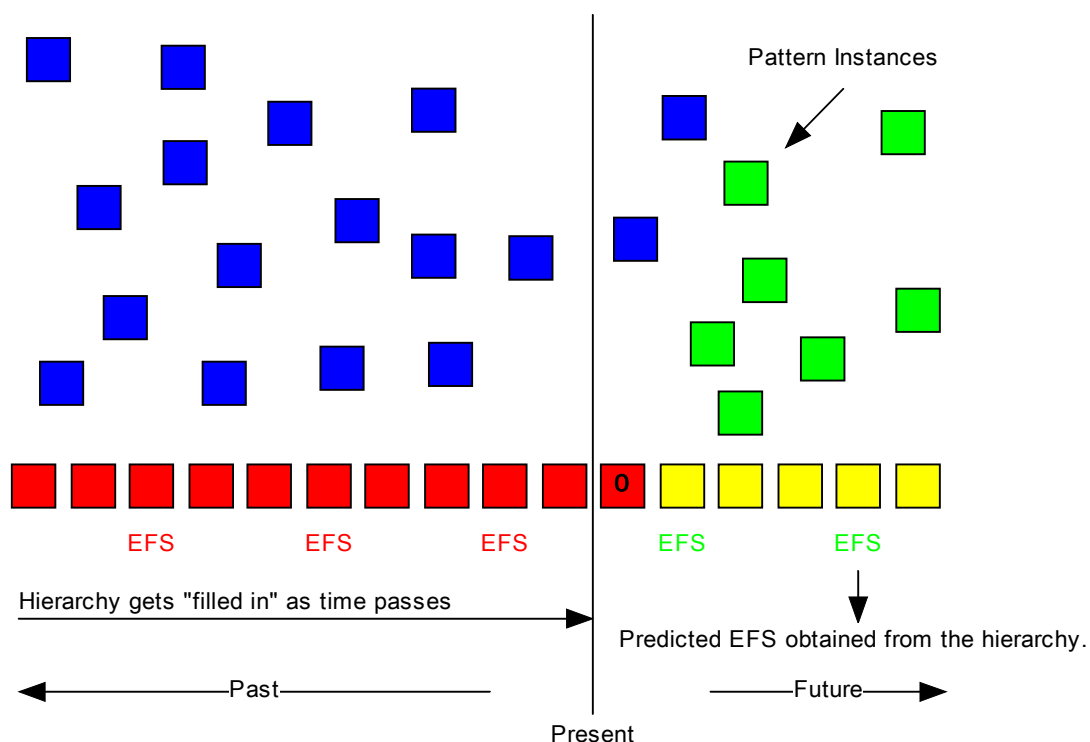
First, try $Output_n=0$:

Pretend that $Output_n=0$ has actually occurred. Set the pattern instance for $Output_n$ as if $Output_n$ has already occurred and $Output_n=0$.

Propagate the effects of this through the hierarchy using logical application and statistics application.

For one or more pattern instances corresponding to input of the EFS at some future time, $t=t_2$, obtain one or more probabilities from the bottom level of the hierarchy.

Use these probability values to obtain a predicted EFS at $t=t_2$. This expected EFS is an indication of the desirability of making $Output_n$ such that $Output_n=0$. (See Figure 3 on page 13.)



Key

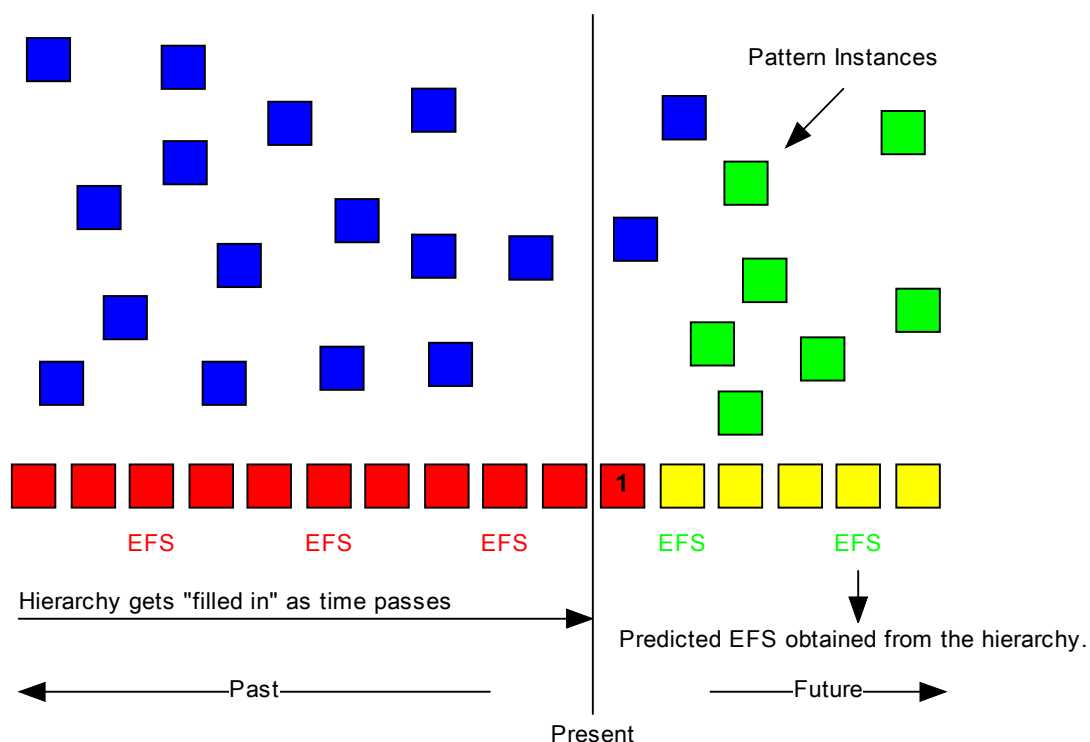
- EFS Evaluation function score previously provided as an input.
- EFS Predicted evaluation function score input.
- Bottom-level pattern instances corresponding to external inputs/outputs that have already occurred. Each of these is "fixed": Its pattern output value was assigned, permanently, when the relevant input/output event occurred.
- Pattern instances which depend, directly or indirectly on bottom-level pattern instances for inputs/outputs which have already occurred. Each of these is also "fixed": Some of the pattern instances that depend on future inputs/outputs are shown blue here, because the value of an imminent output has been set as if it had already happened.
- Bottom-level pattern instances corresponding to external inputs/outputs that have yet to occur. Each of these has not yet been "fixed": The pattern output value will be assigned when the relevant input/output event occurs.
- 0 Value being tried for an imminent output. The red color is because, as far as the hierarchy is concerned, this output has now occurred.
- Pattern instances which depend, directly or indirectly on bottom-level pattern instances for inputs/outputs that have yet to occur. Each of these has not yet been "fixed": Its pattern output value will be assigned, permanently, when all the pattern instances being used for its inputs have been assigned pattern output values.

Figure 3: Trying 0 as an Output Value

An Attempt to Generalize AI – Part 2: Planning and Actions

Reverse the changes just made to the hierarchy.

Then, try $\text{Output}_n=1$ using the same procedure as was just used for trying $\text{Output}_n=0$.
(See Figure 4 on page 15.)



Key

- **EFS** Evaluation function score previously provided as an input.
- **EFS** Predicted evaluation function score input.
- Bottom-level pattern instances corresponding to external inputs/outputs that have already occurred. Each of these is "fixed": Its pattern output value was assigned, permanently, when the relevant input/output event occurred.
- Pattern instances which depend, directly or indirectly on bottom-level pattern instances for inputs/outputs which have already occurred. Each of these is also "fixed": Some of the pattern instances that depend on future inputs/outputs are shown blue here, because the value of an imminent output has been set as if it had already happened.
- Bottom-level pattern instances corresponding to external inputs/outputs that have yet to occur. Each of these has not yet been "fixed": The pattern output value will be assigned when the relevant input/output event occurs.
- **1** Value being tried for an imminent output. The red color is because, as far as the hierarchy is concerned, this output has now occurred.
- Pattern instances which depend, directly or indirectly on bottom-level pattern instances for inputs/outputs that have yet to occur. Each of these has not yet been "fixed": Its pattern output value will be assigned, permanently, when all the pattern instances being used for its inputs have been assigned pattern output values.

Figure 4: Trying 1 as an Output Value

Compare the predicted EFS values for $Output_n=0$ and $Output_n=1$. The output value with the greater score is the more desirable output value.³

Actually make $Output_n$ occur with the most desirable output value. Assign the value of the pattern output for the pattern instance corresponding to $Output_n$ with the relevant value, to indicate that it has now occurred.

and that is it.⁴

³ This assumes that a higher score corresponds to greater desirability.

⁴ Actually, there are some variations of this method, which I will not go into here, but what was just described is the main principle. I have described this kind of approach to planning of actions previously, with more detail, in the following article: Almond, P. (2007). Planning As Modelling in AI: A New Version. *paul-almond.com*. <http://www.paul-almond.com/PlanningAsModellingNew.pdf>. (Also available at <http://www.paul-almond.com/PlanningAsModellingNew.doc>.)

4 How it Works

4.1 The *real* planning process

Because the hierarchy predicts future inputs/outputs, it can be used to predict the system's future situation. We can use this to test the expected results of making an output with some value. It is as simple as it sounds: Tell the hierarchy that the output has occurred with some value (even though it has not yet occurred really), and then ask it how desirable the future situation seems.

The process as described above for selecting an output may seem to be the main process by which the system plans – and it may seem simplistic if it is supposed to capture all the complexity of human planning. Could it really reduce to something like that? What I will now say may seem strange. The process of action selection that I just described is not the “real” action selection process, but is just a means to an end. The real, deep action selection process actually occurs within the hierarchy itself – within the model. The system is using modeling for its really deep planning.

What are the grounds for thinking this?

First imagine that the system is controlling a car in a simple videogame: It has to drive the car along the road, moving left or right to stay on the road.

Imagine that the system has a history of playing this game competently. (Do not worry about how it got this history, but just pretend it has it.) Pattern instances corresponding to previous inputs/outputs in the hierarchy describe a history of moving the car correctly – going left when left was correct and right when right was correct.

Suppose we are *not* using the action selection process described previously, but instead we just look at the hierarchy's predictions of the value of an imminent output, and actually make the output with this value.

We want to know whether some imminent output, $Output_n$, should be “left” or “right”. We look at the probability for the pattern instance corresponding to this output, and it tells us that it is more likely that the output will be “right”. We select this as the output, so $Output_n$ actually occurs with $Output_n=Right$. We then tell the hierarchy that the output has occurred with this value, so the pattern instance for $Output_n$ has its pattern output value fixed with “Right”.

What we just did would actually be a good way of selecting the output. The prediction for $Output_n$ was based on everything that has been happening in the past – on the history of inputs/outputs encoded in the hierarchy's fixed pattern instances – and this history is one of playing the game competently. The system's predictions will be based on the patterns relating all these inputs/outputs. Whatever patterns, relating inputs to outputs, correspond to playing the game competently, will determine the system's

predictions of its own behavior. The system will predict future behavior for itself that continues the previous pattern of competent behavior relating inputs to outputs: The system will predict competent behavior for itself.

When inputs have indicated that the road is bending right, the system will have tended to have made outputs to move right, and when inputs have indicated that the road is bending left, the system will have tended to have made outputs to move left. Any predictions for future outputs will be based on a continuation of this pattern into the future. If inputs have just been received indicating that the road is bending right, the hierarchy will predict an output of “Right”, because that is what has happened in the past. This example is only simple, but the same principle would apply if the pattern relating previous inputs and outputs were more complex.

All this means that, assuming a perfect modeling system and looking at things simplistically, if the system’s pattern instances already contain a history of competent behavior, basing the system’s behavior just on predictions of imminent outputs, would simply continue this behavior.

4.2 Why the Action Selection Process is Needed

If what I just said is true, why is the action selection process I described needed at all? Why not just base all the system’s behavior on predictions of what the hierarchy is going to do next. There are two problems with this idea.

4.2.1 Problem 1: Initially, there is no history of suitable behavior.

In the above discussion, I asked you to assume that the hierarchy contained a history of competent behavior, and not to worry about where this comes from. In reality, the system would not start with any such history: It would start with nothing. This would prevent it from behaving competently in the first place, with such a simplistic approach: It would need a history of competent behavior before it could behave competently to produce such a history. It would be in a catch-22 situation.⁵

4.2.2 Problem 2: Random drift would occur.

No probabilistic modeling system, like the one proposed for the hierarchy, will be perfect. The predictions of future behavior will have some error. The behavior that such system produces would, at best, be *almost* as good as the behavior that would have been extrapolated from previous behavior by perfect modeling. This behavior would become part of the system’s behavioral history, contributing to predictions of behavior that will be used to generate further behavior, causing it to deteriorate slightly, and so

⁵ Also, as will be discussed shortly, modeling will produce most of the learning, but this will need a behavioral history with a pattern of improvement, and the action selection process is needed to establish this, unless such a history pre-exists in the system.

on. This feedback loop would continue, with nothing to halt the deterioration in behavior, until eventually the system is behaving randomly.

4.2.3 The Same Problem

Both of the above problems are special cases of the same problem: The driving of such a simplistic system is tautological, there being nothing to drive the system, beyond its own history, in the direction of improvement. Something outside the hierarchy is needed to give the system's behavior any preferred direction. This need is met by the action selection process previously discussed.

4.2.4 How the Action Selection Process Gets the System Started

The action selection process has the role of driving the system's behavior from outside. It does not need to drive the system to a great extent: The hierarchy is still doing the main work. All it needs to do is start the process off by causing the behavior of the system to improve initially, and then prevent random drift.

Suppose now that we have a system which uses the action selection process described previously.

Initially, the system has no behavioral history. The action selection process relies on predictions of inputs corresponding to future EFS values, but there is nothing on which to base these predictions. The system's initial behavior will be random.

After the system has been behaving randomly for a while, simple patterns will start to emerge between various actions and various effects and later EFS values: It will be apparent that doing certain things, in certain situations, causes low EFS values to occur later, and doing other things, in certain situations, causes high EFS values to occur later. The action selection process will now start to give meaningful predictions: Some output values will be determined to be better than others because they really are better, and these output values will be selected. The system is now behaving non-randomly, though its behavior is still simple.

The history of previous inputs/outputs provides a context, and that context is one of slightly non-random, slightly organized behavior. When any new output value is being assessed for desirability, it is being assessed within the context of all these previous actions: The hierarchy will assume that the previous behavior is a guide to what will happen in the future, and any specific output that is being considered will cause EFS predictions dependent on how it fits in with this previous behavior. (This issue of context is important: Outputs are not being assessed in isolation.) The fact that the action selection process is now selecting output values within the context of a history of slightly improved output values means that the outputs selected will become slightly better. This will in turn improve the behavioral history, improving the behavioral context for later action selection, and so on.

4.2.5 Learning as Modeling

The system as described above may seem to depend on the external action selection process for any improvement in its behavior, the modeling capabilities of the system just being used to determine the consequences of any action. If this were the case, the system's learning capabilities would be very limited, because the action selection process itself is very shallow. However, the action selection process is only there to impose, externally, a direction on the system. The real learning will come from the hierarchical modeling system itself.

How does this happen? I have already described how the action selection process, working with the hierarchy, can produce at least some improvement in behavior. This means that, after some time, the system will have a history of *improving* behavior: The pattern instances corresponding to previous inputs/outputs on the bottom level of the hierarchy will describe behavior which has been gradually improving. *This improvement is itself a pattern*. When the hierarchy predicts the consequences of making a particular output, it will be looking at the output within the context of a pattern of improving behavior and outputs that best fit into such a pattern will result in the highest EFS values. The action selection process only has to cause a small amount of improvement in behavior, and the hierarchy will model an improving system on this basis. The improved behavior that results will become part of the system's behavioral history, in the bottom-level pattern instances, and this will contribute to modeling of a still more improved system – on top of the improvement that would have been modeled already. This cycle repeats, with improvement in behavior being added to the behavioral history where it leads to modeling of further improvement.

For an easy way of thinking about this, imagine the simplistic version of the system I have already described, in which the action selection process is removed and an output is selected merely by getting a prediction of it from the hierarchy, just before it is made. Assume that we somehow have a history of *improving* behavior. This improvement will have a pattern and the hierarchy will extrapolate this pattern into the future, predicting outputs which are slightly better than previous ones. These better outputs will become part of the behavioral history.

Approaches to planning of actions with some similarity to the one described in this article have been tried before in reinforcement learning, but I think that the potential power of such an approach tends to be overlooked. The approach gives a system with the power to extrapolate any small amount of learning you initially give it into continued, deeper learning, given a sufficiently general modeling system, which the first article was intended to describe. It may seem like an attempt to get something for nothing – to get the learning from nowhere – but the learning is really coming from the hierarchical model. Planning and learning have now been largely subsumed into the modeling process.

4.2.6 How the Action Selection Process Prevents Random Drift

From the above, it may seem that the action selection process is only needed to get the system going and establish an initial pattern of improvement, and that we could then dispense with it and just use the system's predictions of its own outputs to select them; however, there is also the problem of random drift. If an attempt were made to do this, small imperfections in the modeling by the hierarchy would affect outputs, which would in turn degrade the behavioral history, and in turn the system's performance, this cycle eventually causing random behavior.

As well as getting the system started off, the other purpose of the action selection process is to push the system, slightly, in the direction of improvement.

4.3 Why the Evaluation Function Score is Used as an Input

I have described treating the EFS as an input, so that each time it is computed, it is encoded as pattern outputs for one or more bottom-level pattern instances and stored in the hierarchy. To determine the desirability of making some output, the hierarchy is updated as if the output has occurred, and one or more pattern instances that will be used for future output of an EFS value is obtained to get an indication of desirability of the future situation.

Some readers may wonder why this is done. Why not just obtain predictions for conventional inputs (e.g. from vision or sound input devices) and apply the evaluation function to them to determine the desirability? The problem with this approach is that it requires the system's predictions of the future to be very specific. If the hierarchy predicts that, on average, some future situation will have a desirability of 20, this could happen in many ways. There could be a huge number of potential permutations of inputs, and trying to predict so specifically is much harder.

As an analogy, suppose that someone invents a pair of glasses which tell you how pleasant your current situation is. The idea is that the glasses tell you when good things seem to be happening to you and warn you when bad things seem to be happening to you. The glasses do this by looking at what is going on in front of you, using a camera, and applying some AI to this to generate a score, which is then superimposed on your field of view. If you find some money in the street, the glasses may display a high score. If someone punches you in the face the glasses may display a low score, warning you that you are having an unpleasant experience, so that you can deal with this.⁶

Suppose you are walking down the street wearing these glasses and a score of 80 is being displayed in your field of view. You trip and fall. As the ground rushes up to hit

⁶ I am not expecting anyone to take such an invention seriously. It is just being used to illustrate a point.

your face, you realize that the score displayed by the glasses is going to be a lot lower when your face hits the ground and the glasses realize that things are unpleasant. You guess that a score of about 15 is expected.

That score of 15 could result from a huge number of different ways of hitting the ground. To actually predict the details of the accident, in terms of inputs/outputs would be difficult – and maybe even impossible. However, that is not needed to predict how bad it will be. The predicted score is an average across this huge number of possible outcomes. You might not know exactly how you are going to experience hitting the ground, but you know it will be bad.

This is the kind of thing we are doing when we treat the EFS as an input.

5 The Self

What I have been saying here is supposed to be a description of how we might make AI work and how we may work. Some readers may have noticed an absence of any reference to concepts such as minds, consciousness or “the self” in the discussion so far. Instead, I have taken a rather mechanistic approach, just trying to describe how a system may be made to behave intelligently. There is no explicit attempt to represent anything like “consciousness” or “the self” in all this, so what of “the self”? How does it relate to all this?

The hierarchy contains patterns relating previous inputs and outputs. We might consider this to be representing how the “outside world” behaves, but any distinction between an “outside world” and an “inside world” is, from the point of view of the hierarchy’s model, meaningless. The hierarchy makes no such distinction. It simply models inputs/outputs, based on any patterns relating them, whether these patterns result from events in the “outside world” or events in the system itself.

Some of the patterns in the modeling system will correspond to things in the outside world. It may have some pattern relating inputs, corresponding to “tree”, or a pattern relating inputs over a period of time corresponding to “moving closer”. Some of the patterns will relate outputs to later inputs, so that they link actions to the consequences of those actions.⁷

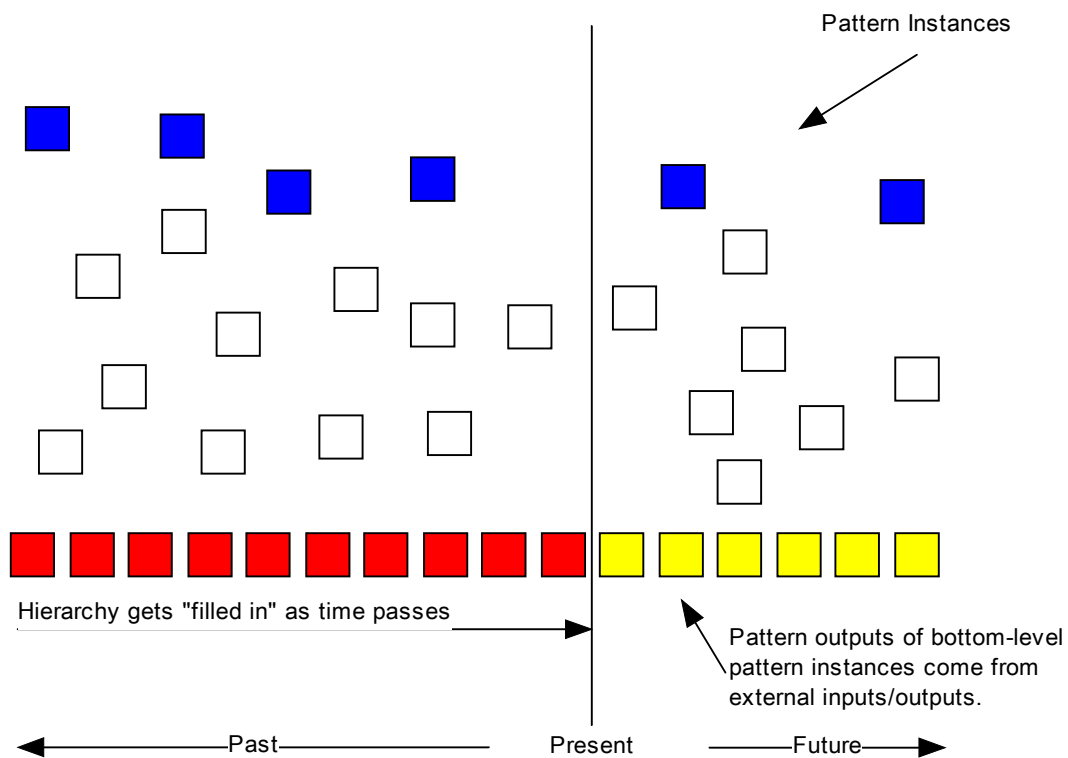
Some kinds of patterns should particularly interest us, in any consideration of “the self”. These will be patterns that relate outputs to later outputs. Patterns like this are modeling the system’s own behavior. The hierarchy will model the AI system itself. What we think of as “the self”, “consciousness” or “intentionality” all come down to the system’s representation of itself in its own hierarchical model. As well as applying to AI systems, I am saying that this applies to us. What this means (and I know it is a strange statement) is:

“You” are an object in a hierarchical model, constructed to explain “your” brain’s previous behavior.

The idea that the self is merely an object in a model has been previously proposed by Thomas Metzinger, a philosopher.⁸ (See Figure 5 on page 24.)

⁷ This should explain why a simple EF in the action process is not inconsistent with the complex, highly abstracted motivations that we experience: They are not the same thing. The motivations, desires, goals, etc. that we experience are part of the abstracted model of our behavior in the hierarchy, far above the level at which the EF works, which has emerged from the system’s complex behavior.

⁸ Metzinger, T. (2003). *Being No One: The Self-Model Theory of Subjectivity*. Cambridge, MA: MIT Press.
Metzinger, T. (2009). *The EGO Tunnel: The Science of the Mind and the Myth of the Self*. New York: Basic Books. While noting Metzinger’s work, I will stay away, in this article at least, from the philosophical issue



Key

- "The self". You are here.
- Pattern instances of other patterns.
- Bottom-level pattern instances corresponding to external inputs/outputs that have already occurred.
- Bottom-level pattern instances corresponding to external inputs/outputs that have yet to occur.

Figure 5: The Self

of whether this means that the self is not real. I have said what I think the self is, and how it gets caused in a system. There are technical issues to cover for now.

6 Chess, Computers and Humans

Chess has proved a difficult problem for computers. Although it seems that computers can now play as well as the best human players, this has been achieved after a lot of research on algorithms and the construction of computers with a lot of processing power.⁹

Chess algorithms usually perform a tree search. The possible moves are examined, the possible responses to each move, the possible responses to these and so on. The computer power and/or time needed to perform such a search increases out of proportion to the number of moves ahead that we want to look, and at a certain point it becomes unfeasible. Various methods are known for increasing the efficiency of chess tree searching, but they only help up to a point.¹⁰

Could humans really be running anything like that? I am skeptical about the idea that this is anything like how humans really play chess. Humans may consciously think like this – they may imagine the search tree for some limited depth – but I suggest that deep-level, human intuition about chess is based on a completely different method.

I think that the way humans play chess is much more like the kind of process described in this article. When a human player is faced with different choices for a chess move, the player's brain hypothetically assumes that each possible move is actually selected as the move. The consequences of this are then modeled using a hierarchy similar to the one described in the previous article, using patterns which have previously been found useful for describing the progression of chess games. In doing this, the chess player's brain is not just modeling the other player: It is modeling itself. If the chess player happens to consciously think about things like tactical sequences of moves or tree searches, all of this is happening within parts of the model that happen to deal with the chess player's own behavior. It may be that a rethink is needed about the basic idea of how to get computers to play games before computers can do well at games such as *Arimaa* which are difficult to deal with using tree searches.¹¹

⁹ It should be noted, however, that efforts in computer chess at the time of writing are concentrated on software, rather than hardware, using more efficient algorithms than those used with IBM's *Deep Blue* in 1997.

¹⁰ Heinz, E, A. (2000). *Scalable Search in Computer Chess: Algorithmic Enhancements and Experiments at High Search Depths*. Vieweg Verlag. Chapter 0, pp11-18.

¹¹ <http://arimaa.com/>.

7 Conclusion

The previous article in this series described a probabilistic, hierarchical modeling system. No explanation was given of how this system would be made to plan actions, but such an explanation has now been provided.

The method relies on the ability of the modeling system to predict future inputs from the history of previous inputs/outputs which are stored on the bottom level of the hierarchy as pattern instances with fixed (permanently set) values.

In this method, an evaluation function (EF) continually computes an evaluation function score (EFS), based on recent inputs. This score is an indication of the desirability of the situation. The EFS is treated as an input to the hierarchy, with pattern instances being continually used to encode it. These pattern instances are part of the bottom level of the hierarchy on which predictions are based. When an output is imminent, the different values for the output are tried hypothetically, with the pattern instance corresponding to the output being updated as if it had occurred with each value. In each case, the effects of this are propagated through the hierarchy. Probabilistic predictions for pattern inputs that, in the future will be used for input of the EFS are then obtained from the hierarchy, giving a prediction of the EFS, and therefore an indication of future desirability of the situation, in each case. The output value which seems to lead to the most desirable future is selected and the output is made with this value.

A hierarchy like this would, in principle, work for some time without such an action selection process, given a pre-existing history of competent behavior stored in the hierarchy: When an output is imminent, a value could be selected just based on what the system indicates that it is about to do. The above process is necessary, however, because there is no pre-existing history of competent behavior, and the problem of random drift needs resolving.

Learning occurs when the system's behavior improves over a period of time. There is a pattern to the change in behavior as it improves, and the hierarchy will model this pattern and incorporate it into its predictions when determining the consequences of making an output, so that output will be considered within the context of the system's improving behavior.

There is no explicit feature of this system to provide anything like "consciousness" or a "self". Instead, these are objects in the hierarchical model, like everything else. In this case, they are objects relating behavior to later behavior. They can therefore describe such things as "intentionality". This can be summarized with the following statement.

"You" are an object in a hierarchical model, constructed to explain "your" brain's previous behavior.

A consideration of this kind of system should show why it has been so difficult to get computers to play chess well. The intuitive thinking that humans use in a game like chess will not be based on tree searches, but on a process like that just described.

I have now described both how the hierarchical model works (the previous article) and how it can be used to plan actions (this article). A general overview is now, therefore, complete. I am suggesting that this is how AI systems should be constructed and that it is how brains work. The method described here is basic. Improvements may be possible. I have proposed a method that works from outside the hierarchy, just interacting with the bottom level of the hierarchy – the pattern instances corresponding to inputs/outputs – but we should maybe not completely rule out, at this stage, a planning method which involves doing things, especially for planning, at higher levels of the hierarchy.

This series of articles is still far from complete. In the first article I described what a pattern specification is supposed to do, but not exactly how it would be implemented – and I admit not having full details on this. The issue of efficiency needs discussing. The hierarchy, as described so far, would involve every input/output that occurs being stored permanently as a pattern instance with a fixed value, as well as every pattern instance with a value derived, directly or indirectly, from previous inputs/outputs. It is unlikely that the human brain works in this way, and it is impractical for an AI system. In a later article, I will describe how this can be dealt with by building *forgetting* into the system. The system, as currently described, represents every pattern instance in a pattern explicitly. This is also impractical, and ways of dealing with it will be discussed later. The system described so far, when considered as a possible explanation of how brains work, leads to a plausible explanation of dreaming, and I will also discuss this.

8 Bibliography

Almond, P. (2007). Planning As Modelling in AI: A New Version. *paul-almond.com*.

Retrieved 27 February 2010 from

<http://www.paul-almond.com/PlanningAsModellingNew.pdf>.

(Also available at <http://www.paul-almond.com/PlanningAsModellingNew.doc>.)

Almond, P. (2010). An Attempt to Generalize AI - Part 1: The Modeling System. *paul-*

almond.com. Retrieved 27 February 2010 from <http://www.paul-almond.com/AI01.pdf>.

(Also available at <http://www.paul-almond.com/AI01.doc>.)

Heinz, E, A. (2000). *Scalable Search in Computer Chess: Algorithmic Enhancements and Experiments at High Search Depths*. Vieweg Verlag. Chapter 0, pp11-18.

Metzinger, T. (2003). *Being No One: The Self-Model Theory of Subjectivity*. Cambridge, MA: MIT Press.

Metzinger, T. (2009). *The EGO Tunnel: The Science of the Mind and the Myth of the Self*. New York: Basic Books.