

Self-Aware Systems NEED Models at Run-Time

Dr. Christopher Landauer
Topcy House Consulting
topcycal@gmail.com

Models @ Run-Time Workshop at ICAC
18 July 2016

We argue that the title is a minimum necessity for significant self-awareness

For us, the term “self-aware” also means “self-adaptive”,
and both are part of the term “reflective”:

Not only can the system **collect information** about its own structure and behavior,
it can **examine** that information to **decide** how to **modify** the behavior,
retain and **analyze** it over extended time periods,
adjust its decision processes to “optimize” it towards given success criteria,
and even **change the success criteria** to respond to changes detected
in the environment.

Self-Modeling Questions

The questions then become: How does a system
learn about its own structure and behavior?
make decisions about its own structure and behavior?
decide on changes to its own structure and behavior?
record and use the decisions and their consequences over time?

For us, all of these are about models or involve or require models.

Our engineering design problems are:

how extensive these reflective mechanisms need to be for an application,
how much we want the system itself to construct and modify the models, and
how much of the system behavior is to be driven by the models.

Models cost time for development and execution,
which makes this set of engineering problems central to the success
of self-adaptive systems in complex environments.

If we make this extensive use of models, we are brought to the notion of
using the models to define the behavior.

These are self-modeling systems.

Other Self-Modeling Properties

There are other abilities that we think will be necessary:

The “Get Stuck” Theorems show that the system will eventually need to change its own internal representations.

The theorem is based on the fact that we can count how many structures of a given size can exist, so representing behavior in a sufficiently complex environment will eventually need to use so many expressions that they will become too large for the system to use either effectively or soon enough.

“Behavior Mining” is the process of recording and analyzing the system’s own behavior to make descriptive or prescriptive models.

“Knowledge Restructuring” is the process of rearranging the hierarchical knowledge structures (e.g., ontologies) to reduce dependencies and duplication.

“Constructive Forgetting” is the process of simplifying knowledge structures by conflating and even removing some terms and relationships.

“Model Deficiency Analysis” is the process of evaluating the effectiveness of a model (not just success and failure, but also performance, resource implications, and interference with other models).