IBM Research

The Policy Decade: Has it Delivered?
An Autonomic Computing Perspective

Jeff Kephart (kephart@us.ibm.com)
IBM Thomas J Watson Research Center
Hawthorne, NY, USA

"Have we delivered on what we promised?"

"Did we make the right promises?"

Or, equivalently

"Are we solving the right problems?"

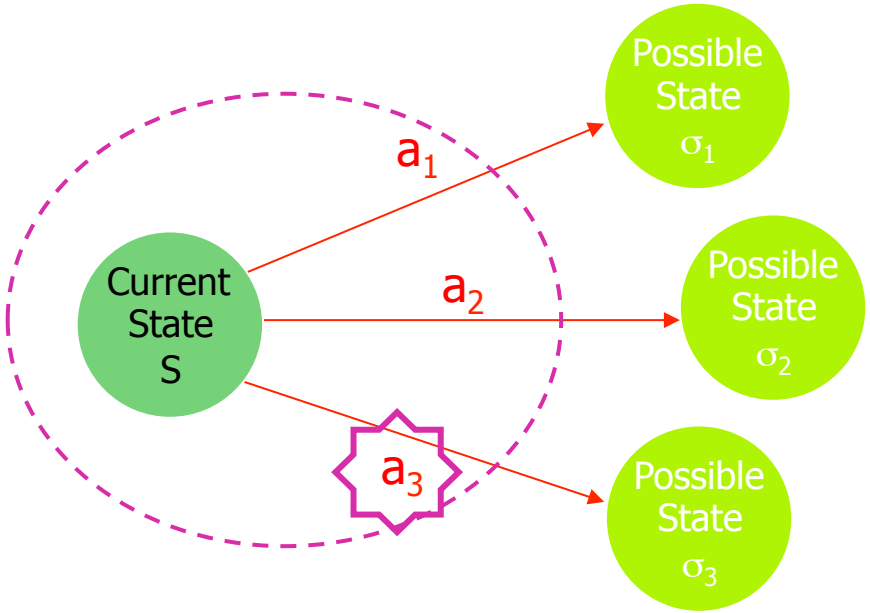
We're missing something important

- Role-based access control, policy-based network management, etc. are all worthy things to work on…
- … but autonomic computing is focused primarily on higher-level policies
  - "Computing systems that manage themselves in accordance with high-level objectives from humans."
- From an AC point of view, the policy community should focus *less* on what we want a system to *do* …
- … and *more* on what we want it to *accomplish*

Action Policy

{ON (Event)} IF(*Condition*) THEN (*Action*)

- *Event/Condition* specifies
  - *Current* state or set of states S
  - Action *a* that should be taken from state *S*
- Nothing is said about the state σ that will result from taking action *a* from state *S*
- Presumably, the rule author had an idea of what state σ would result, but there is no slot in the policy to hold such information
- There is no way to check and see whether the action resulted in the desired state
  - Let us hope that the rule author was highly knowledgeable and very careful
  - Let us hope that nothing unanticipated happened to interfere with the state or action

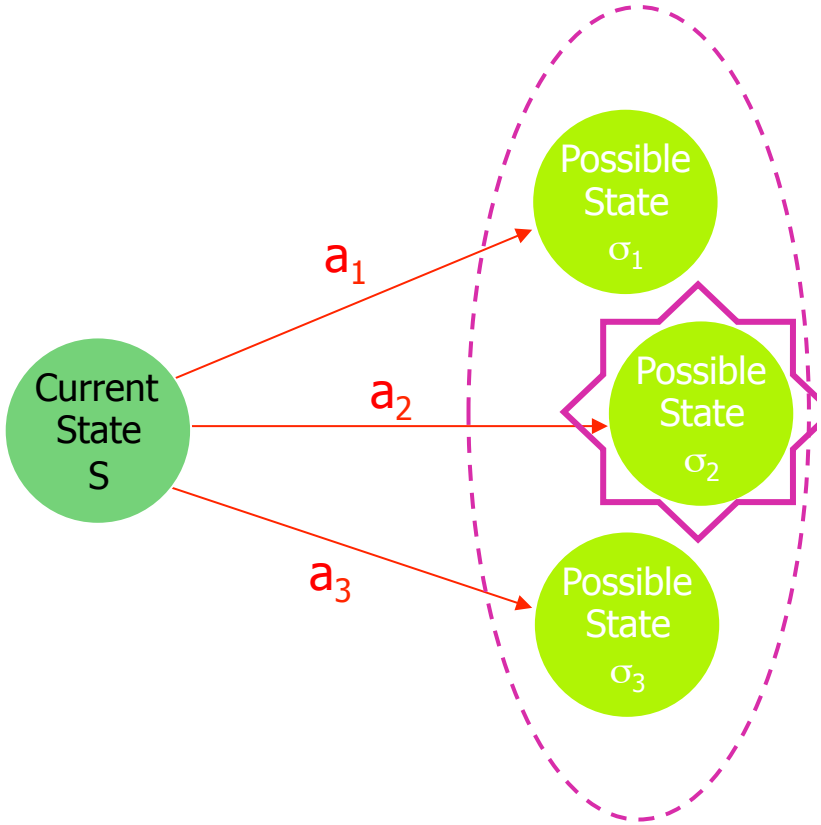


Advantage: Policy fully specifies what action to take.

Disadvantage: This seems inherently dangerous and brittle!

Goal Policy

- Specifies desired *resulting* state σ
  - Or properties that define a set of desired states, any of which are acceptable
- System computes action (or action sequence) that reaches σ from S
- This computation requires
  - A system model σ(*S, a*)
  - Planning technologies (engine, etc.)

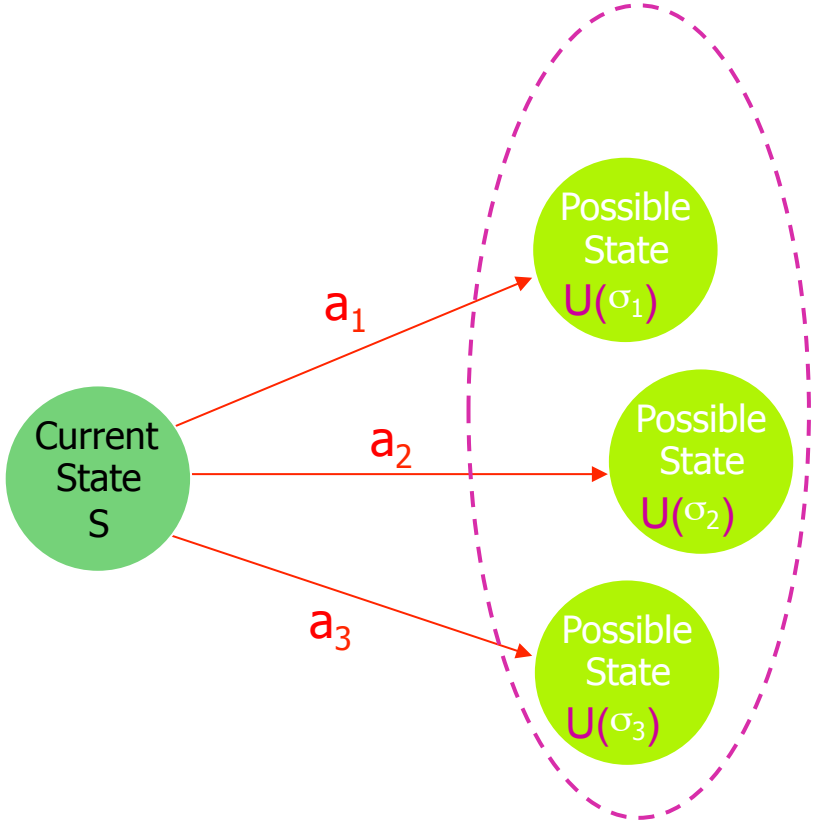


Advantage: Higher level; more flexible.

Disadvantage: Requires sophisticated modeling and planning technologies.

Utility-function Policy

- Like Goal, focus on the states you want to be in
- Assign to each state σ a real value U(σ)
- Compute state σ* for which U (σ*) is maximized
- Compute action to reach σ*
- This computation requires
  - A system model σ(S, a)
  - Optimization/Planning technologies



Advantage: Strict generalization of Goal; even more flexible (allows gradations of good and bad valuations); avoids conflicts.

Disadvantage: Requires sophisticated modeling and planning/optimization technologies, plus utility elicitation (hard)!

Thoughts and Questions

- Goal and utility-function policies are much more true to AC
- But the Policy community seems to have placed relatively little emphasis on them
- Are you content with this?
- If "Yes", is there another community I should be asking?
- If "No", then it would require Policy to reach out more to other research communities (largely AI) that work on
  - Modeling, Planning, Optimization, Learning, Elicitation
- The Policy community's work on authoring policies/rules, automated conflict detection, etc. might be relevant for planning
  - Ex. PDDL (Planning Domain Description Language) describes domains in terms of predicates and actions; describes problems in terms of objects, initial states and goal specifications