An Introduction to Promise Theory

part I

Mark Burgess
Pervasive, autonomous computing

- Personal consumer electronics - autonomy
- Multiple services, no central control
  - Today: in data centres and server rooms
  - Tomorrow: in walls and malls
- Agents are autonomous in their policies
Questions?

How do we (can we?) arrange for a consistent policy in a distributed system?

How do we manage (control?) devices and rulesets in a distributed system?

Hard to answer – requires some notion of cooperation between nodes. But who is in control?
Policy conflicts

• Assume that all nodes will follow a consistent set of rules (no conflicts)
  – Assumes they have common goals?

• Often policy is dictated from a central location
  – Is this this obvious solution to harmony?

• What about distributed systems?
  – Pervasive computing
  – Devices have many masters/mistresses
  – Cooperation with policy is voluntary
Bad habits...

• We tend to think in terms of hierarchy and centralization
  – Hard to break out of this mo(u)ld
  – If you build everything on a presumed truth, you are unlikely to exceed it

• Start at too high a level
  – End up tripping over our shoe-laces
Protocol and timing

- Worry too much about ordering
- Event-Condition-Action (ECA)
  - Triggered fire-fighting
  - Policy in farmer or egg?
- Forget about the sequences
  - Timing should matter less if we have stability
Logics?

• (Modal) logic is often used to try to prove whether nodes will have conflicts
  – Doesn't tell us how to fix things
  – Doesn't tell us how to set things up in the first place
  – How do we formulate propositions to be proven?

• Logic = verification
  – Not a “constructive” method
  – Need an architect, not an art critic
Start again

• Try to begin with the extreme viewpoint
  – Make it primitive and constructive
  – Make it intuitive and distributed (graphs)
  – Make it analyzable (labelled graphs)
  – Make sure it is tied to existing research on autonomy

• Base it around snapshots/epochs in time
  – Sequential interaction seldom interesting for policy
  – Define what you want not how to get there
  – Triggered events – model as services
Don't forget autonomy

- *Voluntary* cooperation
- Each atom is an autonomous agent
  - You cannot force one to do anything
  - It decides for itself
  - Want centralization? Build it from the bricks!
  - Direct relationship to cooperative game theory (economic cooperation)
Promise theory

• A combination of graph theory and set theory
  – Exceeds “Boolean” logic - constructive
  – Handles constraints on possible behaviours
  – Handles relationships, not just true/false rules
  – Service model: “I promise you service S” (SLA)

• Directed graph
Typed/labelled graph

• Promise types:
  – Service promises (promise to constrain behaviour)
  – Cooperative promise (promise to do the same as)
  – Usage promise (promise to make use of)

• Atomicity rule:
  – Only one promise of a given type per pair:
  – Broken promise => two different promises

\[ X \neq Y \]
Example
Consistency

• Any node is free to promise anything
• But when is it breaking its promises (SLA)?
  – Basic promise $S$
  – $C(S)$ and $U(S)$ tell us about structural dependency
  – Use these to reconstruct cooperative structure
Other examples

- Cfengine is an autonomous agent – how do we model it?
Summary

• Integrates logic, planning and heuristics
• Promises describe steady equilibria
• Time is only a distraction – we want stability
• Very easy to use – lots of graph theory results
• Lots of theorems to be proven (re-enter logic)
• A theory for pervasive peer computing

“I will speak daggers to her, but use none
My tongue and soul in this be hypocrites”

--WS (Hamlet)