From adaptive systems to adaptive spaces

Dagstuhl seminar on resilient and survivable networks, infrastructures and services. July 2007
Resilience and survivability seem to be inherently system-wide concepts

- Does no good to have a resilient component if the network as a whole is fragile – and indeed the term itself may have no meaning

Adaptation helps, in a broad sense

- Change behaviour in details to account for changing conditions
- …while offering a predictable and stable quality of service
- …and so dooming engineers to perpetual under-appreciation

The argument we’ll make here

- Achieving this stability requires a whole-system modeling approach
- Able to trade-off and analyse radically different issues
Autonomic is *not* the same as stable or predictable

- Getting decent feedback loops needs work
- Not all feedback is desirable

What we need

- What a system should do in response to stimuli
- A guarantee that it does indeed do this
- A way of constructing such systems compositionally
- Separate description from mechanism

---

An adaptive system should adapt within an *envelope* of acceptable behaviours

- Adaptation is “movement” within a “space” of possibilities

This same basic description is applied to general dynamical systems, as *phase spaces*

- Obtain a closed-form description of a system’s behaviour
- Mathematically well-developed (if a little head-wrecking)
The advantage of this view is that the mathematical tools are very well-developed

- Fields – define a quantity that varies continuously over another
- Especially vector fields and the associated calculus – div, curl, grad and all that
- Closely linked to differential equations, chaos, catastrophes, …
- Topological characteristics that are “desirable” in whatever sense

But it’s not all rosy

- Computers are often discrete, and it’s not clear that a lot of vector field theory applies (but it might…)
- Implies we can define adaptive spaces ahead of time, which isn’t obvious (but is advantageous if we can…)
- No necessary link from semantics to implementation – descriptive, not prescriptive (but this might not be bad…)

This has been used in, for example, coverage calculations in WSNs by Ghrist and da Silva
Drivers typically don’t brake hard enough, or brake too hard.

Heating makes brakes behaviour non-ideally. Distance sensor can “add pressure” to the brake pedal to improve performance.

A network, and the components within it, can be described in terms of their effects on packets

- Arrival curves, delivery curves, packet loss, …
- Implications in terms of delivery, bandwidth, fairness, isochrony etc can be analysed
- Adding sensors provides finer observation

So we are looking at describing these curves in a common framework to find the adaptive space of a network

- May be under-constrained, so several solutions are acceptable
- Different trade-offs represented by different regions of the multi-dimensional space
Construct the adaptive space of the network and any software running on it

- Capture possible contextual constraints as “dimensions” of the space
- Responses form the “co-dimensions”
- Characterise the properties we want to maintain/avoid according to the topology of the space

Different strategies for addressing problems modeled as different dynamics for moving around the space

- Convergence, stability, …

Does this work? Don’t know yet, but there are reasons to hope so :-)

Modeling resilience
To understand the adaptive space of a system, and ensure that its movement within it is consistent with respect to its intended external semantics

- Characterise strategies in terms of how they move the system within its adaptive space
- Ensure we select (one of the) appropriate strategies at the right time
- “Fibre” contexts over the behaviours (strategies) they select

Ideally like an end-to-end model of all these aspects
Conclusion

This problem of whole-system modeling, analysis and understanding that is currently absorbing a lot of our attention

Treat resilience, stability and adaptivity within a framework that’ll support proper analysis

• Guarantees of behaviour generate confidence

Long-term

• Improve both analysis and development of predictably-adaptive systems
• Incorporate semantically well-founded constructions into programming tools