

Rule-based Systems Programmability

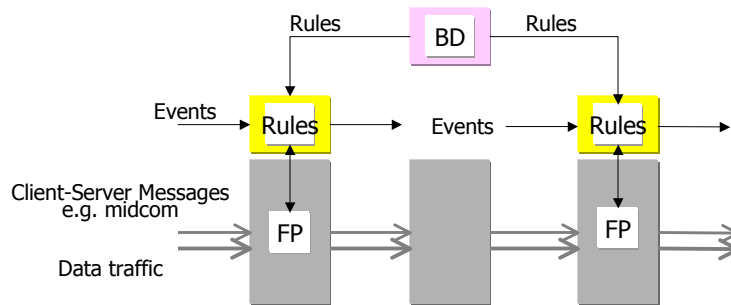
Michael Smirnov
smirnow@fokus.fraunhofer.de

Roadmap

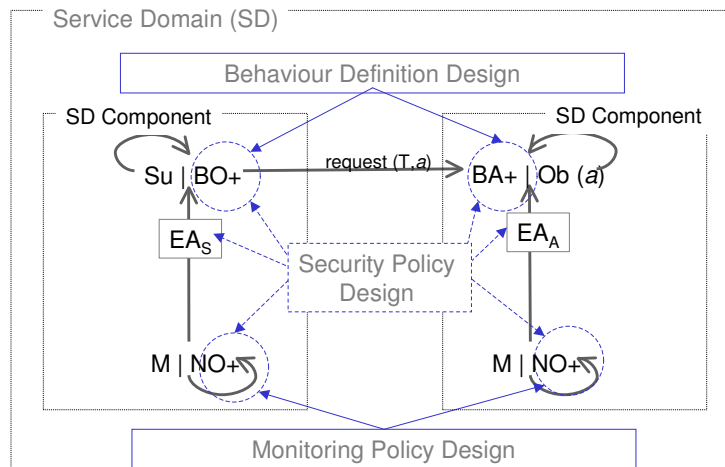
- What's a rule-based system?
- Behaviour definitions: safe and secure
- Run-time updates
- Self-organisation of rule-bases
- Impacts and new architectures

What's RBS?

- RBS behaviour is defined by openly exposed rules rather than by 'hidden' state transitions (as in client-server)
- RBS is typically a control plane implementation that controls invocation of Functional Primitives (FP), it has zero functionality
- RBS is typically securing FP invocation, we want to make RBS securely programmable



Common framework



Legend: Su - subject, Ob - target object; M - monitor; BO - behaviour obligation;
 BA - behaviour authorisation; EA - event authorisation; NO - notification obligation

Safe and secure behaviour updates

- Ruling behaviour
 - Filtering, classification, SLA conformance, etc.
 - Can we also define new behaviours for run-time service creation?
- Programmability:
 - Creation of new services
 - **Service:** A Logical Element that contains the information necessary to represent and manage the functionality provided by a Device and/or Software Feature. A Service is a general-purpose object to configure and manage the implementation of functionality. It is not the functionality itself [DMTF]
 - Responding to the unexpected
 - Designing security policies and a framework for policy management that will allow necessary access to systems and data in previously unplanned ways, and by persons and systems not normally permitted to do so [URL <http://crue.isi.edu/research/report.html>]

Run-time updates

- Need to change rule-base at run-time
- Conflicts with already installed rules
 - Feasibility conflicts
 - Local and global conflicts
- Possible approach: conflict resolution rules
 - Convention? Yes, but allowing proof-based engineering
- Automatic conflict resolution is possible
- Do we need to change the world?
- Need also to detect and resolve automatically rules violation
 - Rule-base auditing!

Self-organisation

- The need to have high performance of rule bases
 - E.g. rule heaps optimised for usage frequencies
- Rule injector has to be synchronised with a rule-base
 - Rule life-cycle management is not feasible
 - Too many potential rule injectors
 - Want to have a radically distributed system
 - Theory does exist (e.g. Adaptive Huffman, FGK)
 - Need negative modalities for rules
- Security and safety:
 - Fine grained access control:
 - Event authorisation
 - Behaviour authorisation and obligation
 - Notification authorisation

Impacts and new architectures

- Even more promises in non-conventional infrastructureless networking
 - wireless sensor, ad hoc, 'ambient' networking, etc
- New architectures: role-based architecture (RBA)
 - Role as a network service, controlling FP
 - *event-action* type of reasoning
 - role-based addressing
 - group communication within network control plane