

Constraint-based Specifications for System Configuration



John A. Hewson The University of Edinburgh

Configuration is complex, errors are expensive

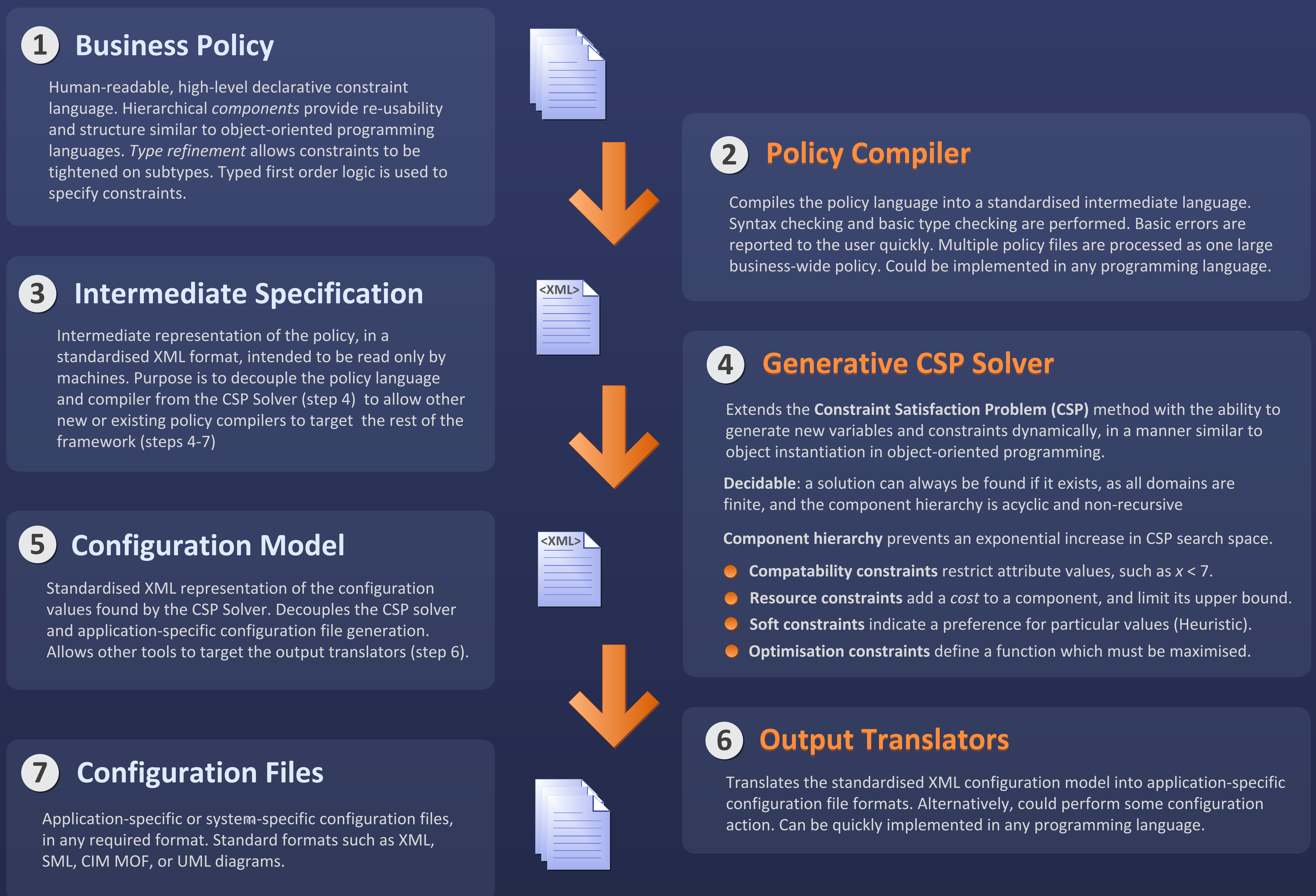
62% of enterprise network downtime is caused by **human error**
Yankee Group (2004)

\$100,000 minimum estimated loss **per hour** of downtime for 42% of US organisations
Forrester Research (April 2004)

28% of network disruptions at **Google** are caused by **configuration error**
The Datacenter as a Computer (2009)

\$2,000 estimated loss **every second** at **PayPal** in a one hour hardware outage
The Risk Factor, IEEE Spectrum Blogs (2009)

A unified, constraint-based approach is proposed



High-level configuration is a complex problem

● Autonomics

The process above is designed to be encapsulated in a self-managing *autonomic element*, allowing distributed and hierarchical control of sub-systems. No specific architecture or schema is required when re-using our framework, a major limitation with existing systems.

● Reconfiguration

When the business policy changes, the solver should find the configuration closest to the current solution, to cause minimal changes. We plan to investigate using local search and heuristics within the CSP solver.

● Federation

Policy and configuration information can come from a variety of sources inside an organisation. A constraint-based system can automatically synthesise composite constraints and detect conflicts.

● Security and Validity

As constraints are defined on each component, and the solver is computationally complete, any configuration which compiles successfully is guaranteed to be valid under those constraints. Compliance with a security policy can therefore be proved.

● About

First year PhD research. Work so far has been identifying a methodology with sufficient expressivity which is decidable, and yet conceptually simple.

● Work Remaining

Implement the above system using F#, including the design of a suitable policy language, type checker, and solver.