

Policy Based Management
*Thoughts and Observations from a
Network Management Perspective*

John Strassner
(john.strassner@intelliden.com)

John – Industry Requirements



DEN-ng vs. the World...

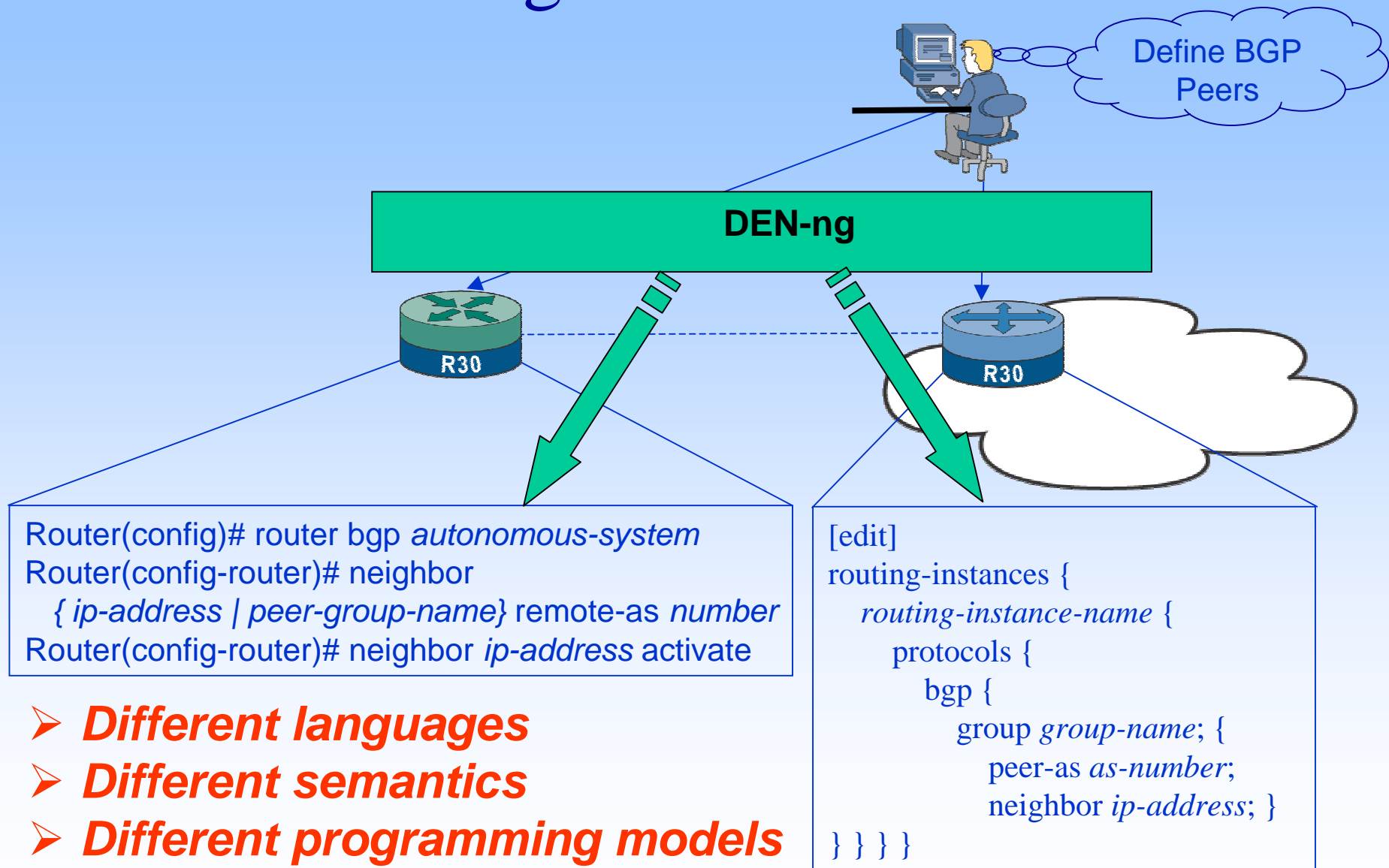
Our Subject...



Our Subject...



Network Management is a Mess



- **Different languages**
- **Different semantics**
- **Different programming models**

Goals, Shmoals...

- The fallacy is that people think that there is ONE policy...
- ...WRONG!
 - “John gets GoldService”
 - Is perfectly reasonable for business analysts
 - Is perfectly meaningless to a NOC technician
 - Will never happen for me (but I diverge...)

The Policy Continuum

Business View: SLAs, Processes, Guidelines, and Goals



System View: Device- and Technology-Independent Operation



Administrator View: Device- Independent, Technology-Specific Operation



Device View: Device- and Technology-Specific Operation



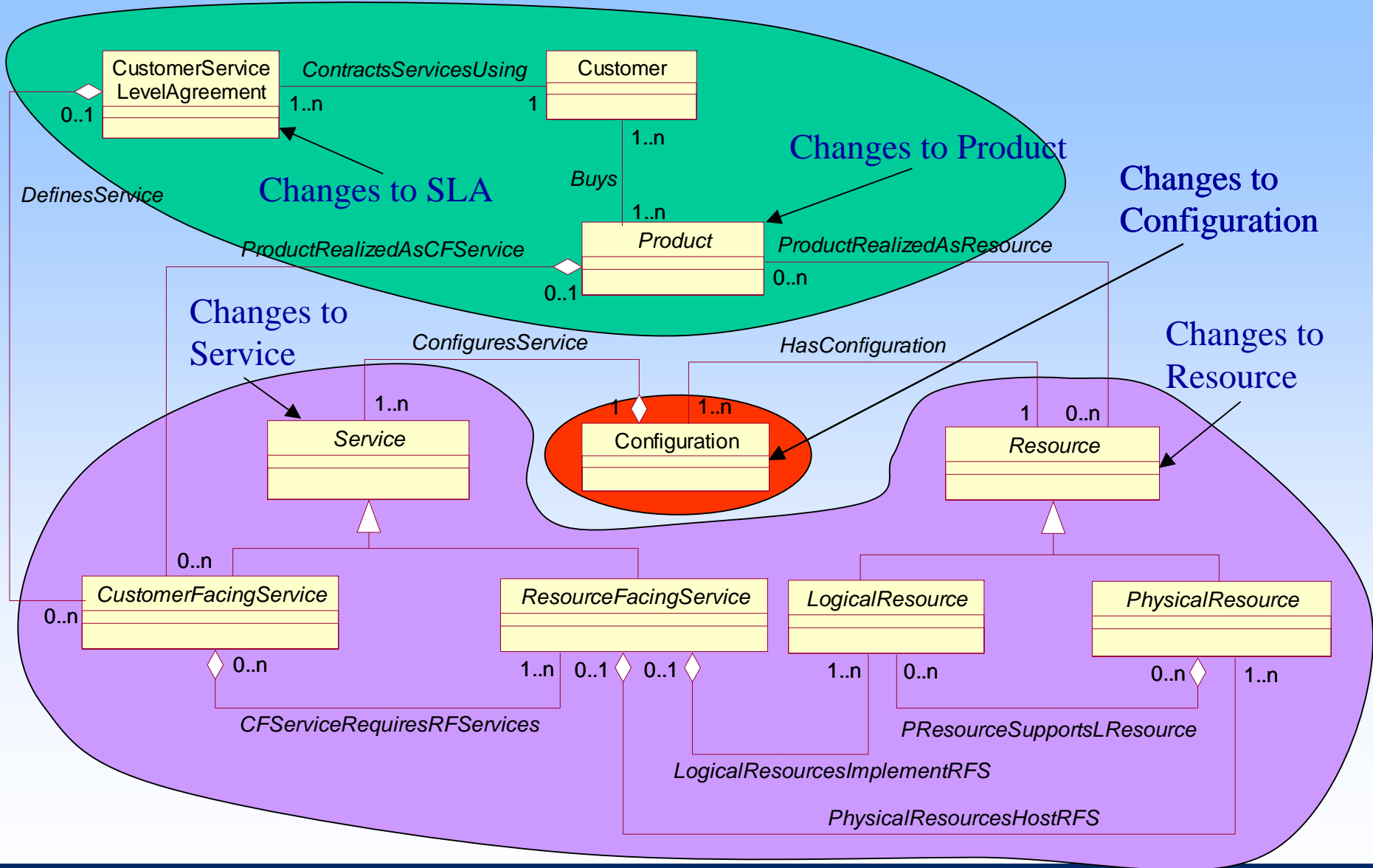
Instance View: Device-Specific MIBs, PIBs, CLI, etc. Implementation

Morris Asked (Too) Many Questions

- Policy Specification (ECA and permit/deny)
 - Maybe...but they need to be understood by **heterogeneous** PDPs, PEPs, PXP, etc.
 - Which is the problem with a **single** Policy Language
- A goal isn't a policy – a policy is used to govern behavior that realizes the goal
- AI techniques have their place, but they are not going to be used in a Telco environment!
- Agents and active networks are a good research topic, but would YOUR network admin use them?

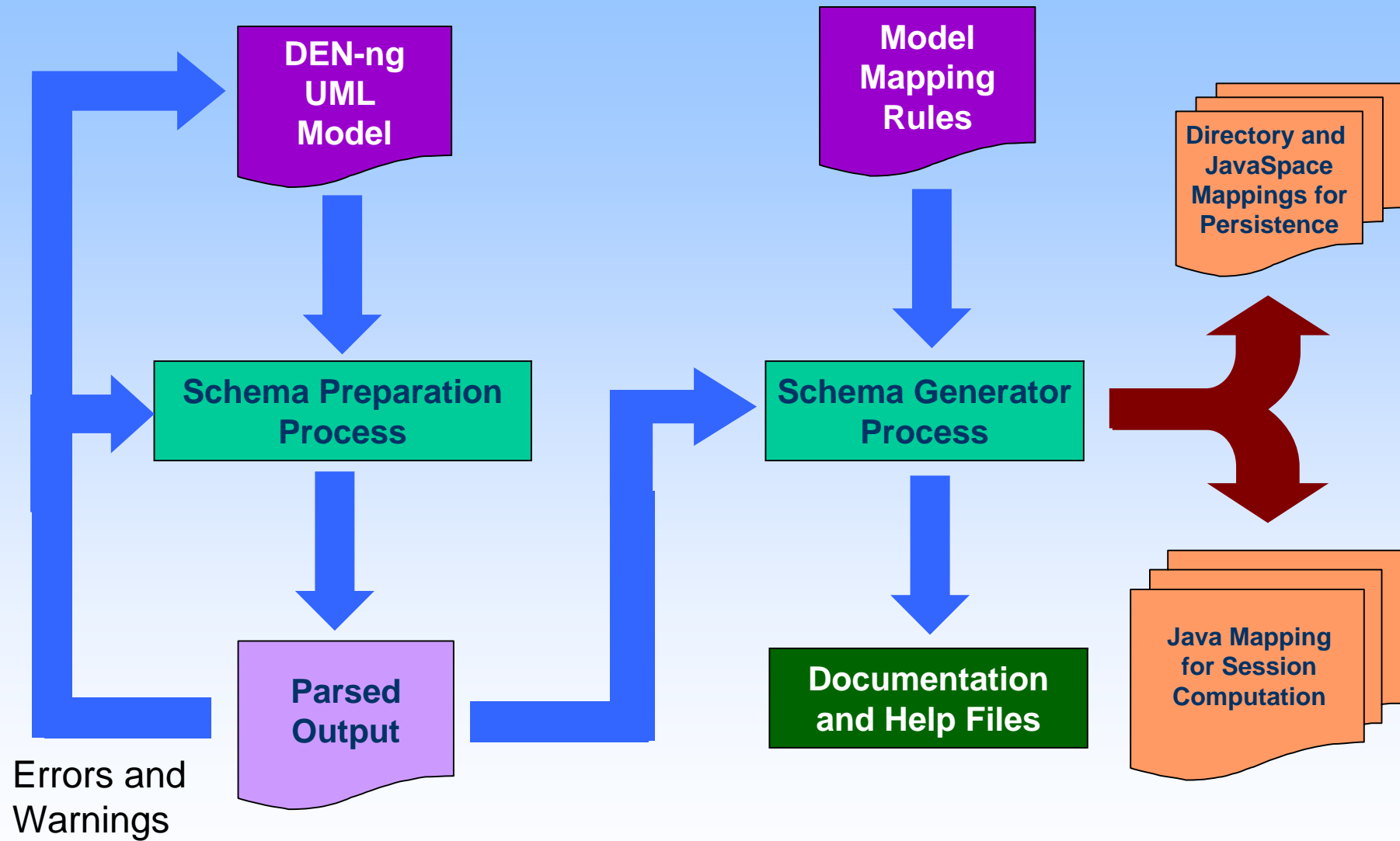
But We Have a More
Important Problem

We Always Forget About the Business...



This is hard, so it must be
automated

DEN-ng Model Driven Code Generation



But Now, the Real Problems

- Policy is a paradigm-shift
- Political-economical-social considerations
 - Everyone's traffic is the most important
 - Lack of OO, UML-compliant, scalable models that have been tested by industry
 - DEN-ng is arguably the first of these
 - Lack of skilled people
- Industry and Academia must be reunited
 - It's the same problem, but needs both perspectives to be solved correctly
- Other than DEN-ng, we haven't addressed
 - How it is used (capabilities, constraints, context)
 - How information is invoked (CONTRACTS!)
 - Policy is MORE than a static class diagram!

Summary

- There are as many policies as it makes sense to the users of the system
- Instead of specifying a universal language
 - We really need to specify the behavior in terms of capabilities, constraints and context
 - We need to formalize behavior using Contracts
- Policy isn't widely deployed because there are few similarities between policy-aware components and systems
 - But that doesn't mean, Give Up!
- ***Academia and Industry need to be reunited***