Technische Universität Braunschweig

ISF

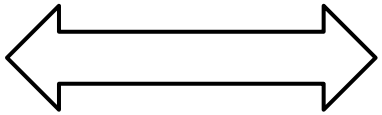
In the Tension of Software Redundancy and Variability

Sandro Schulze (COW Veteran), TU Braunschweig, COW #29

Thanks to: David Wille, Sönke Holthusen, Ina Schaefer (TU Braunschweig), Olaf Lessenich, Sven Apel (University of Passau)

What is this talk about?

"…some disciplines aim at introducing redundancy, others at exploiting it, and others still at avoiding it." (from the COW #29 web page)



Questions…to be answered

Why does redundancy exists?

Any good reasons? On purpose?

Where does it come from?

Beyond plain copy&paste?

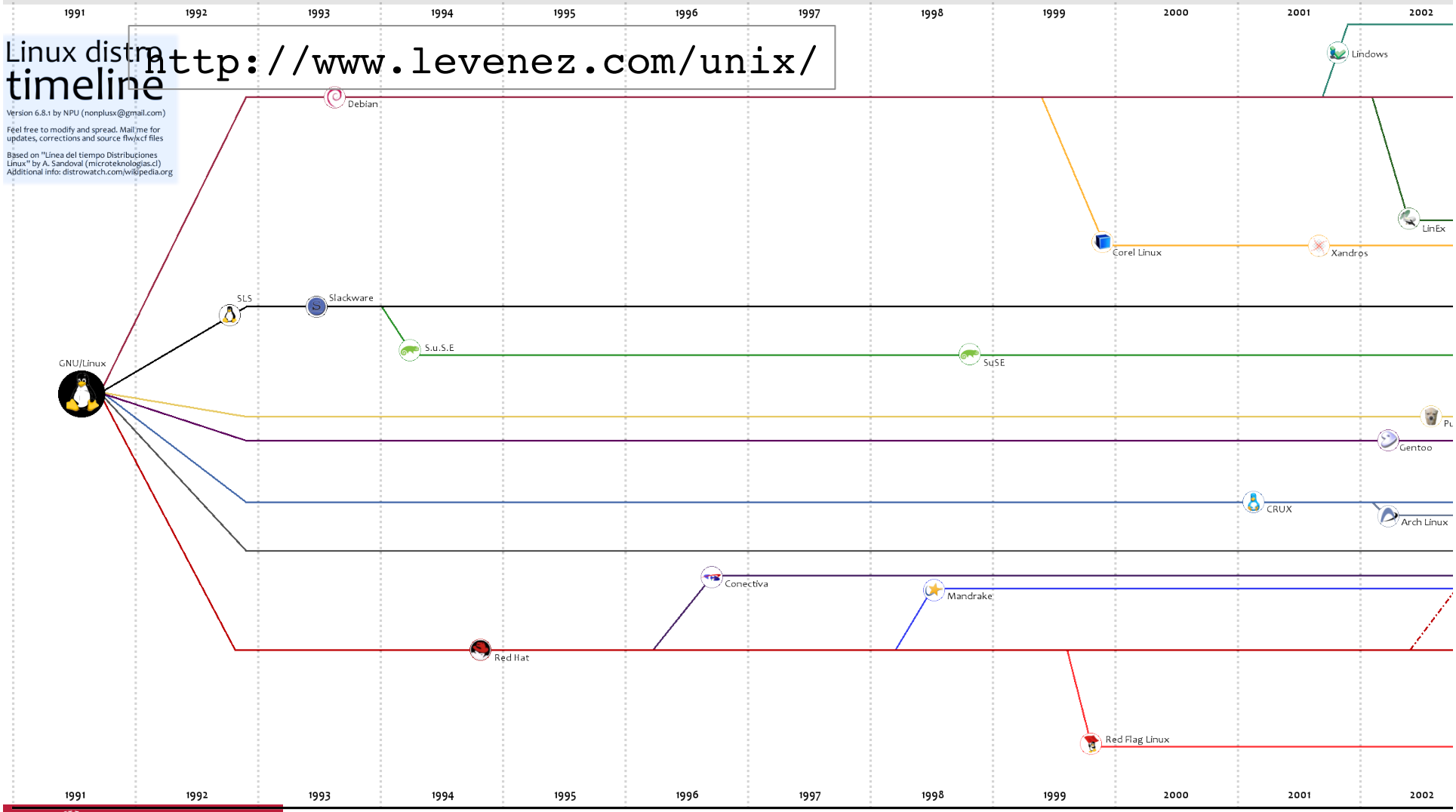
What does it tell us (under the hood)?

What is redundancy really used for…and why?

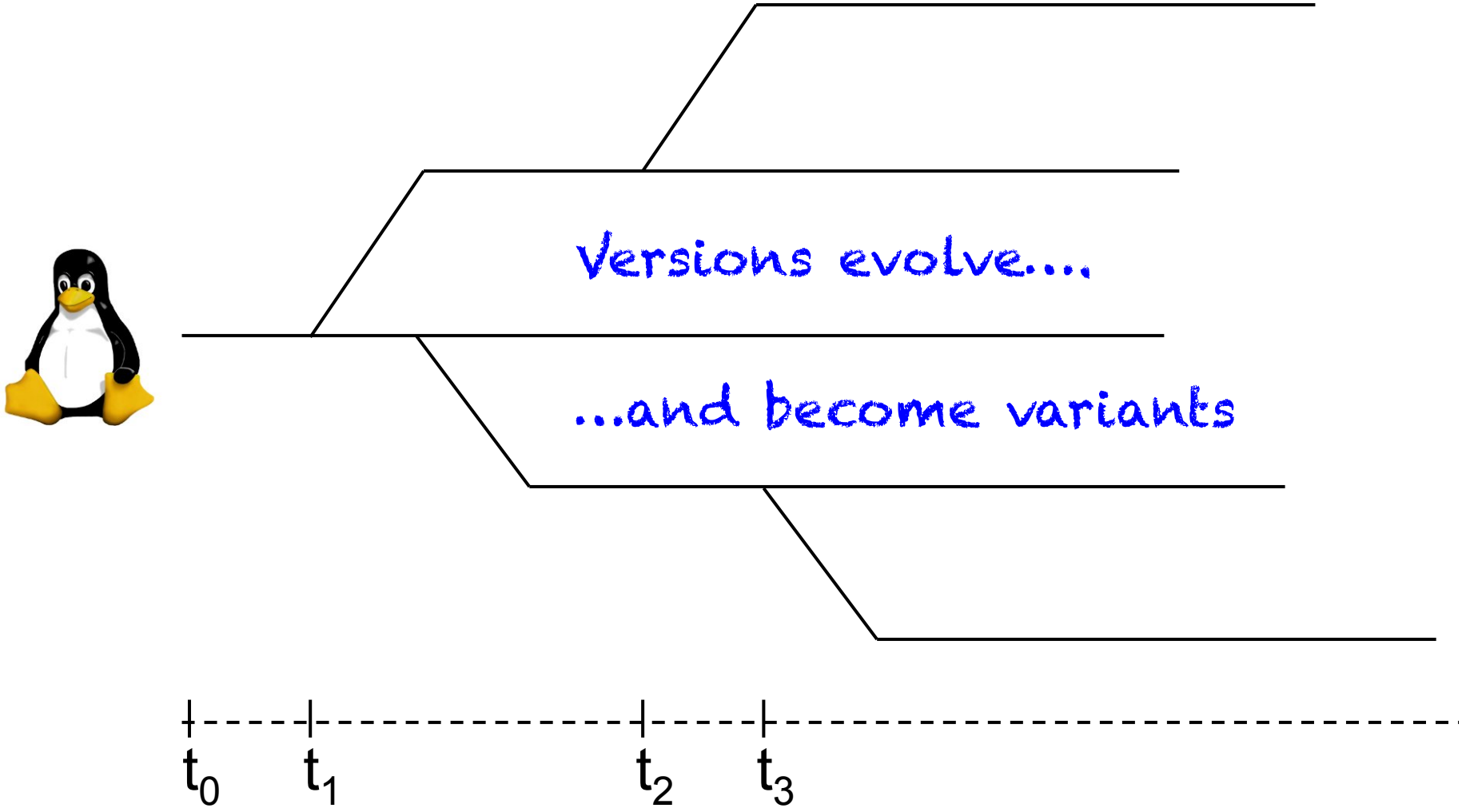
Does your mother know, you are here?

Does anybody know about all that redundancy?

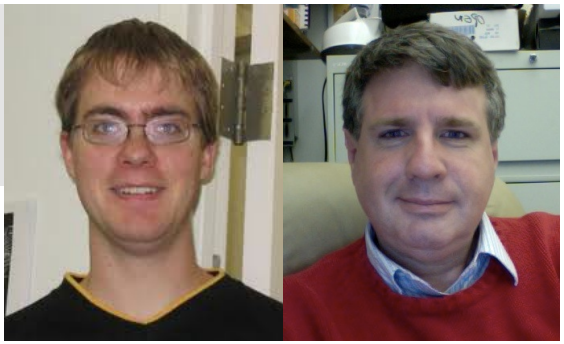
An Example of Software Evolution



Clone-and-Own



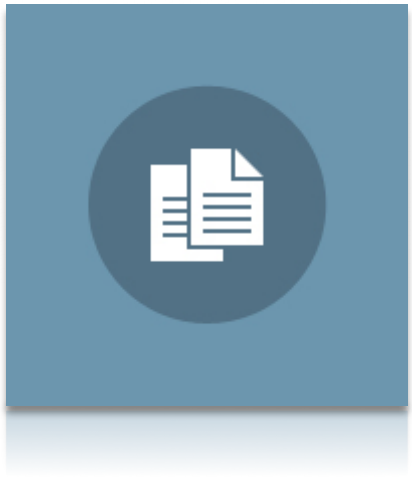
Patterns of Redundancy



WCRE 2006
ESE Journal 2008

Hardware Variation

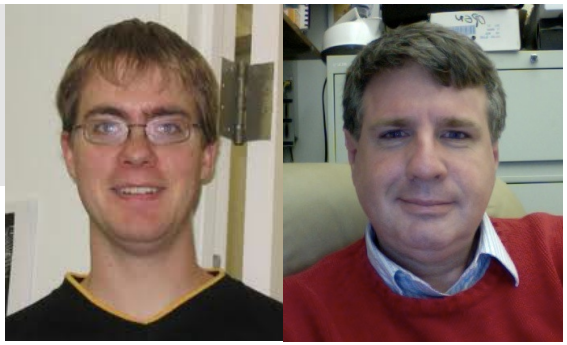
Platform Variation



API/Library Protocols

Algorithmic Idioms

Patterns of Redundancy



Replicate and Specialize Workarounds

Reliability, efficient evolution, knowledge

Maintainability, bug propagation, (missing) back propagation

Redundancy introduces Variability

10000

features

250+

distributions

http://www.distrowatch.com

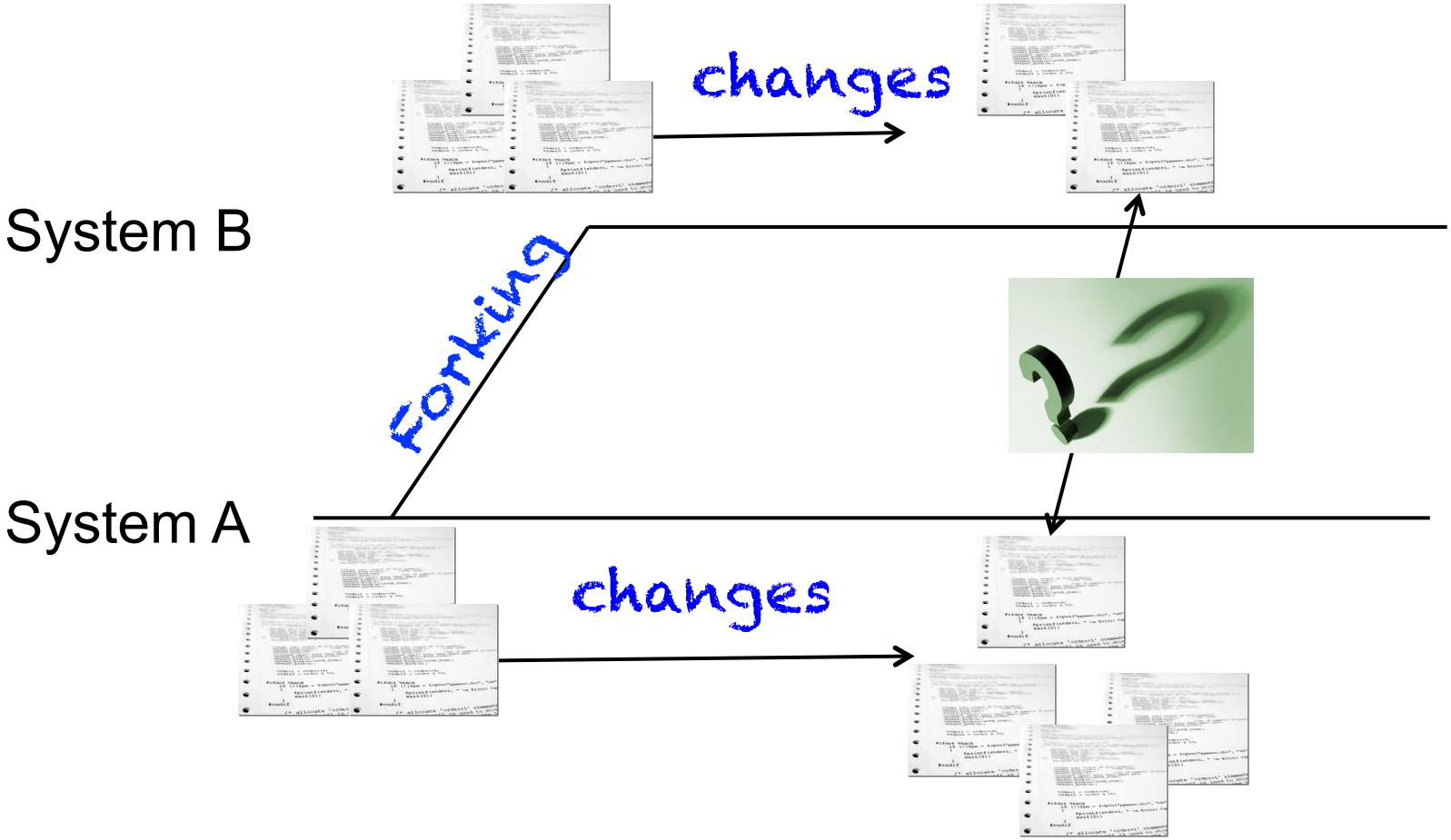
Unleashing Redundancy

λ

Provides Data → How to get there? → Provides Information

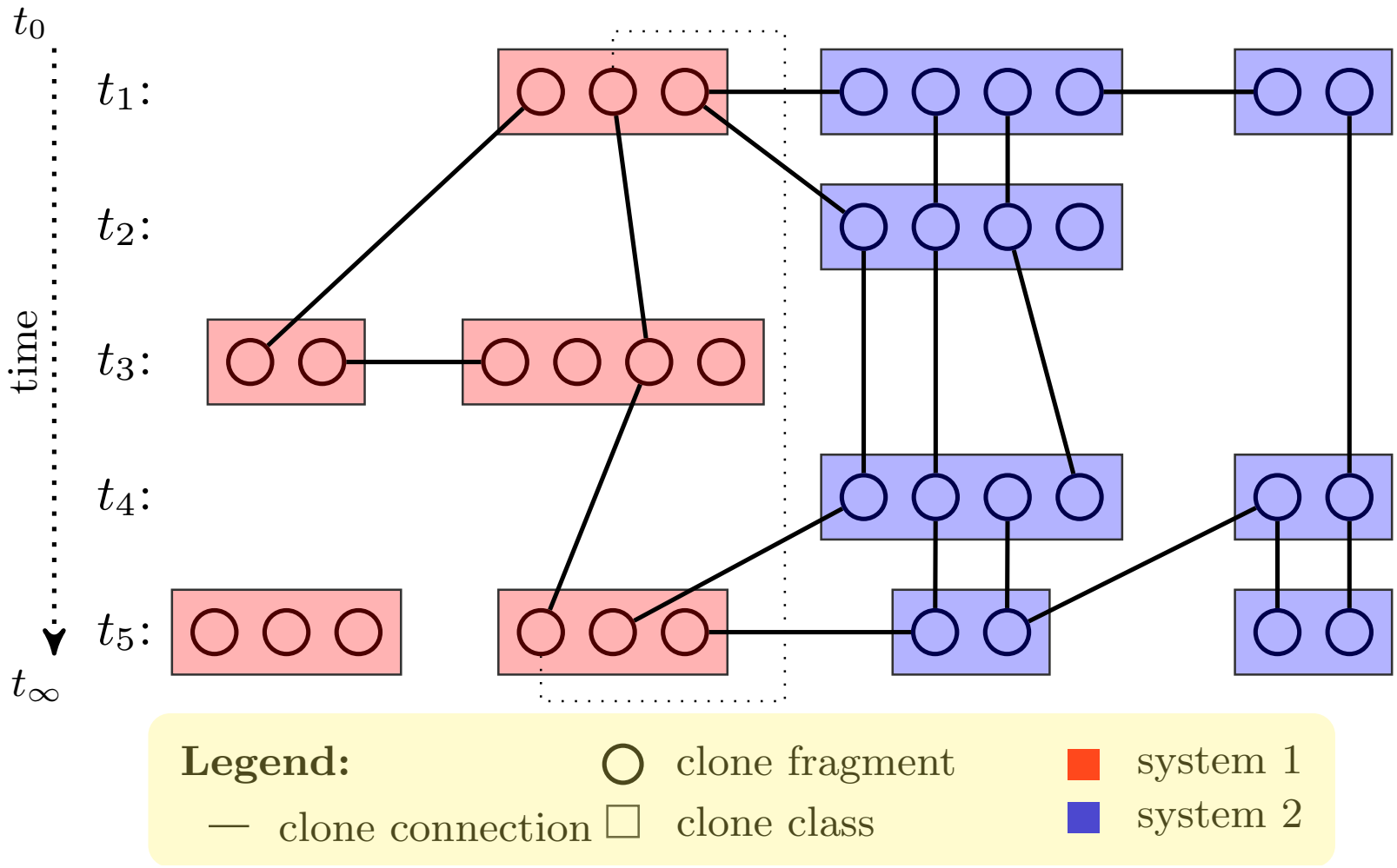
- How to add semantics to redundancy?
- From where to obtain domain knowledge?

Forking & Evolution



Which parts evolve together/independent across systems?

Analyzing Inter-System Clone Evolution



Analyzing Inter-System Clone Evolution

simplicity

vs.

precision

Gives you a fast overview
Gives you regions of interest

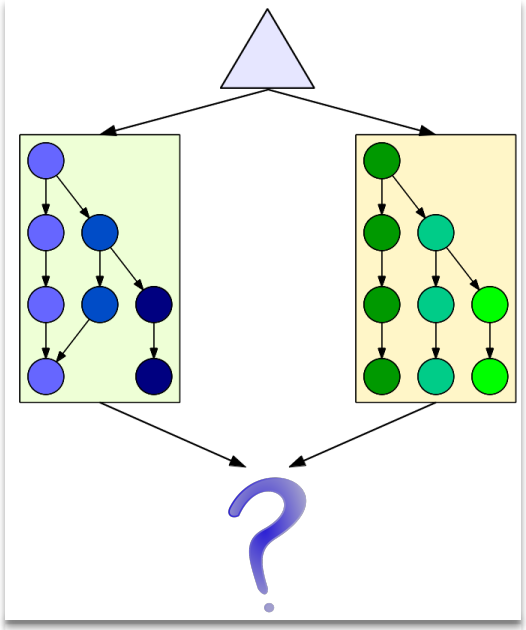
NO information about features
NO information for merging the code base of systems

Semi-Structured Merge

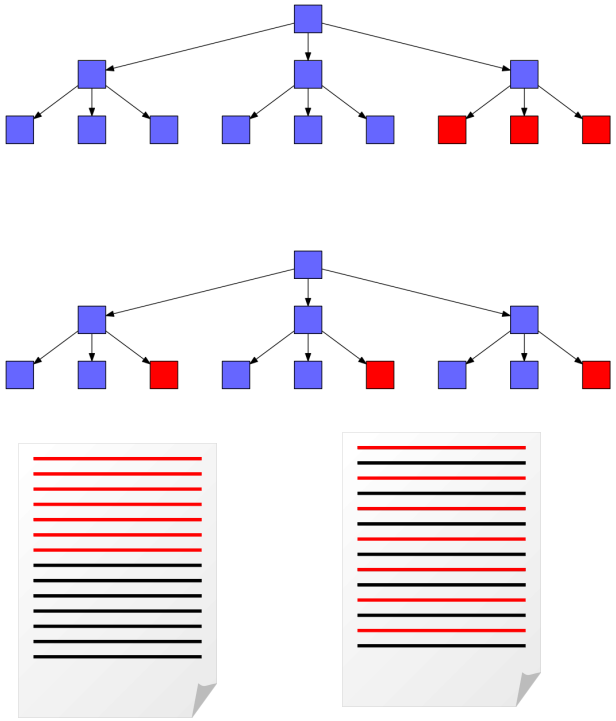
Apel et al., ASE 2012

Structured Diff on abstract syntax trees (AST)

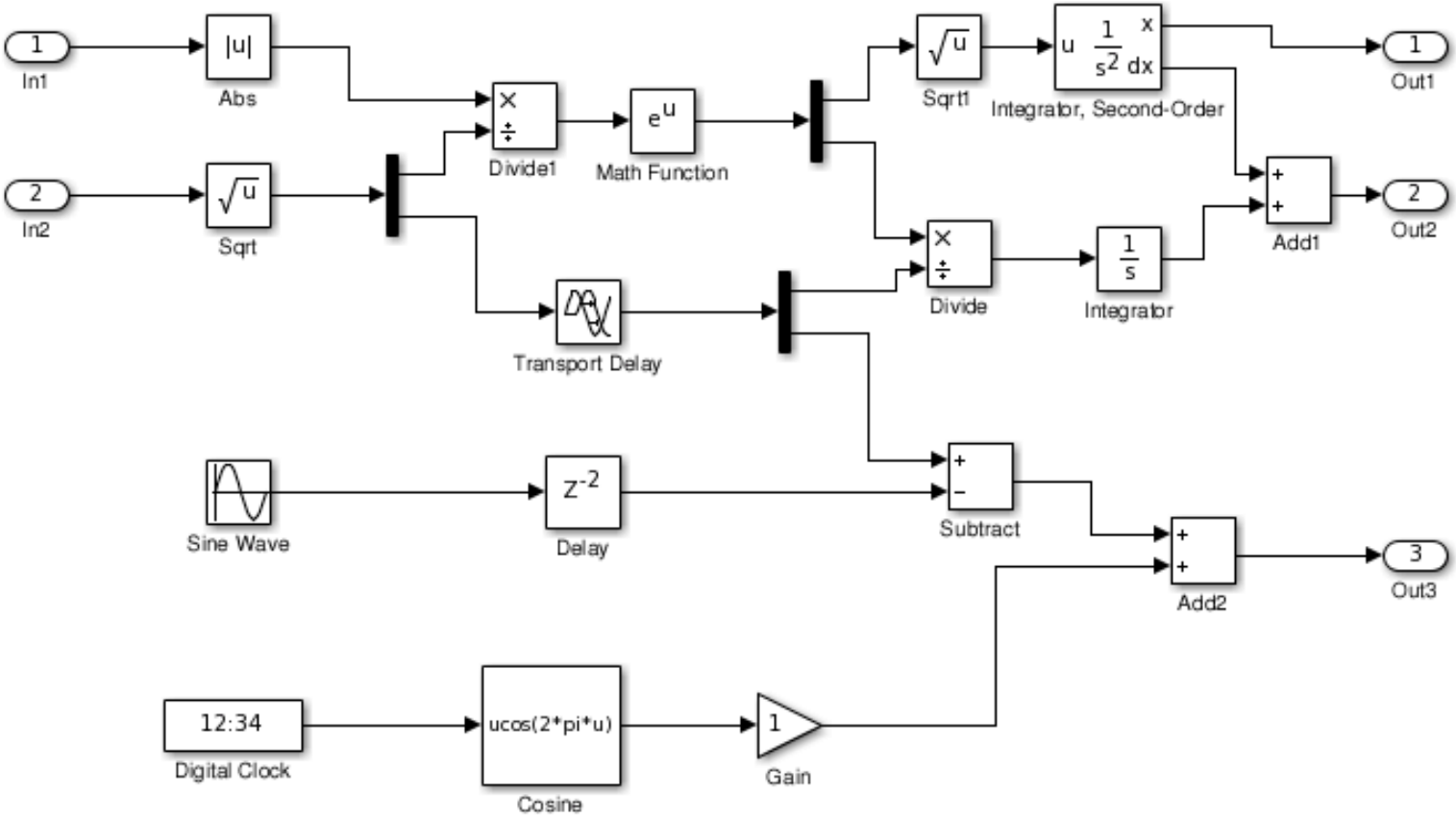
How much does this cost?



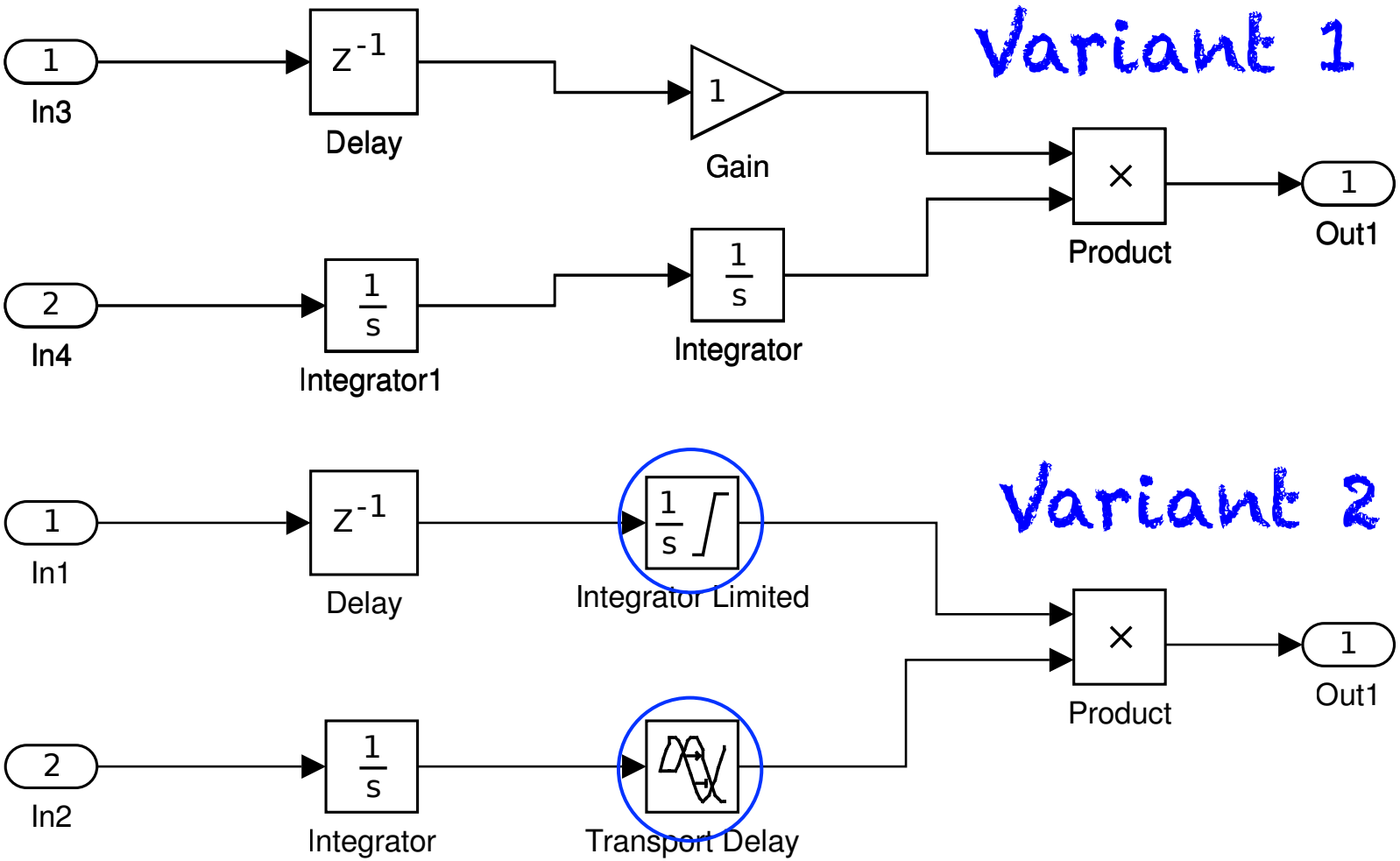
How difficult is it to merge?



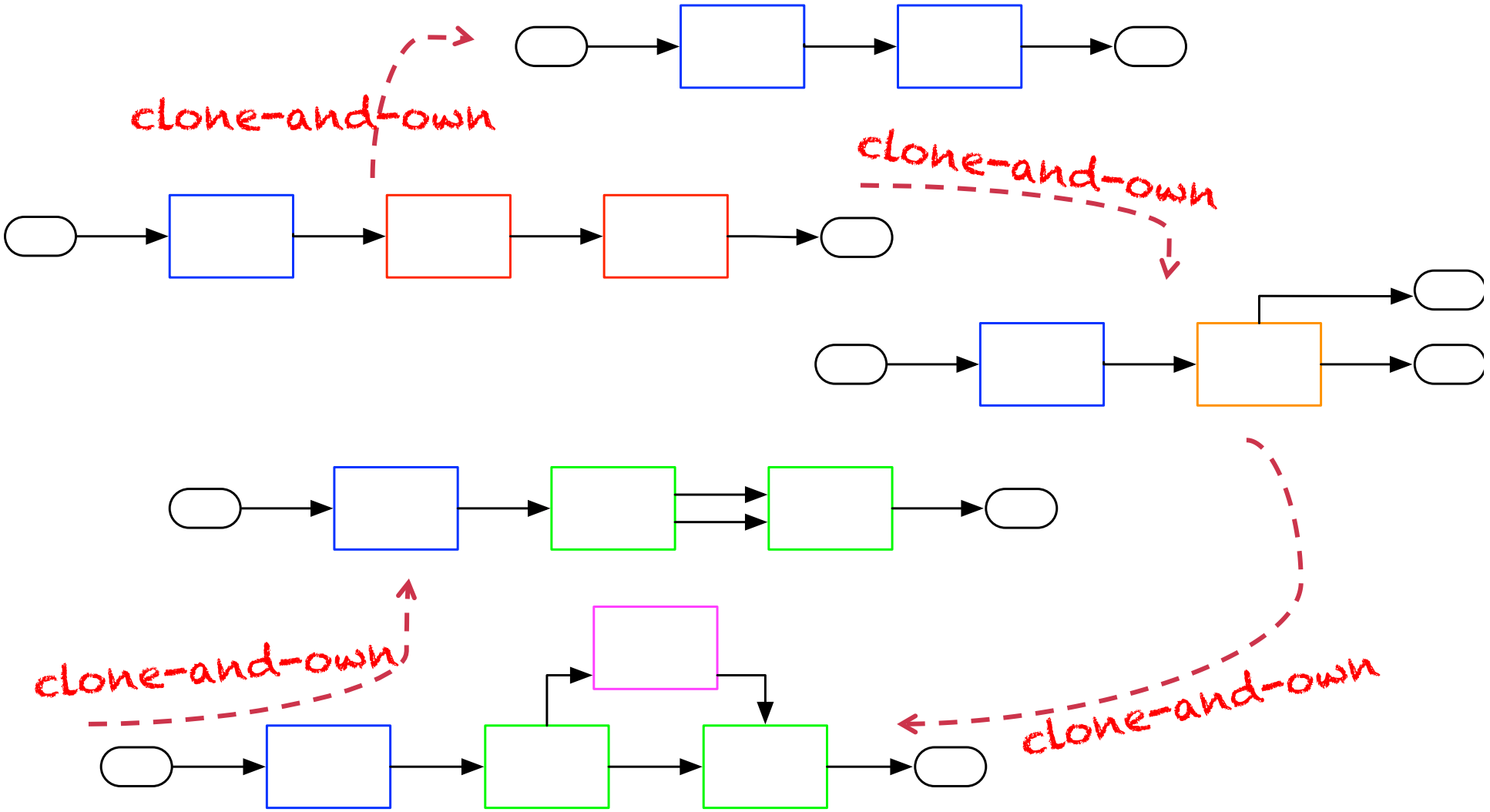
Models are software...too



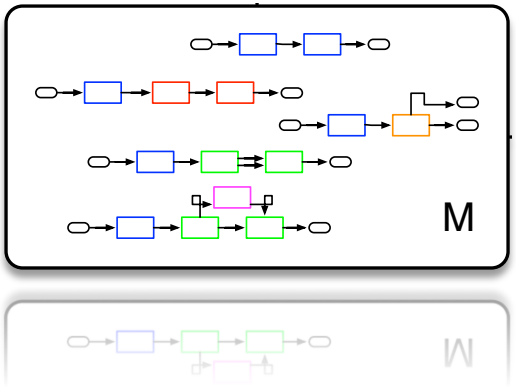
Model-Based Development Process



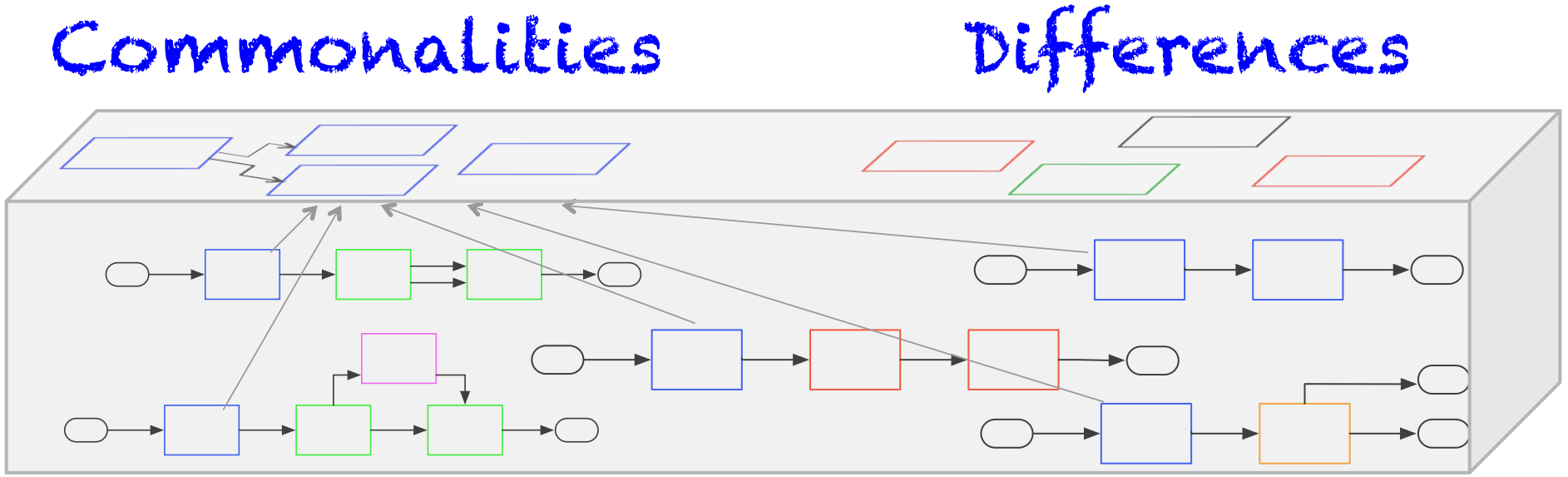
The Model Zoo



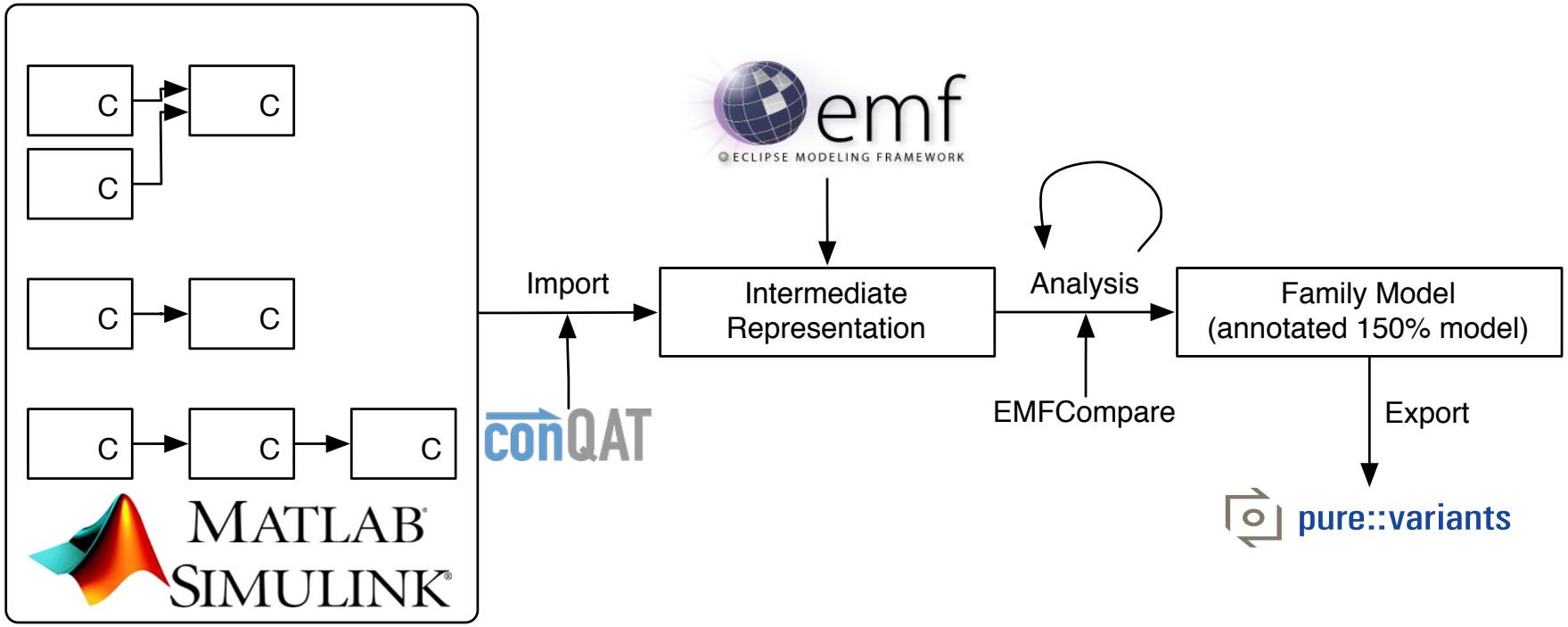
Towards A Family of Models



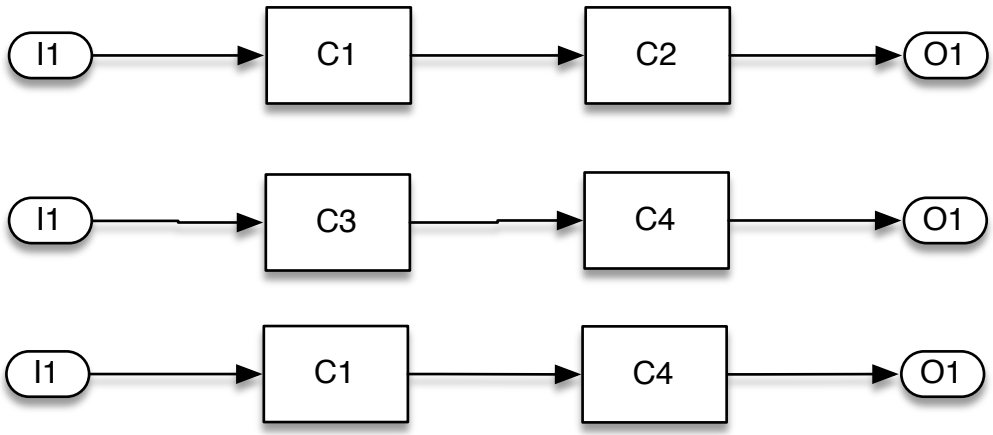
- Difficult to maintain
- Propagating changes → which models?
- Replication usually not documented



Mining Model Variability



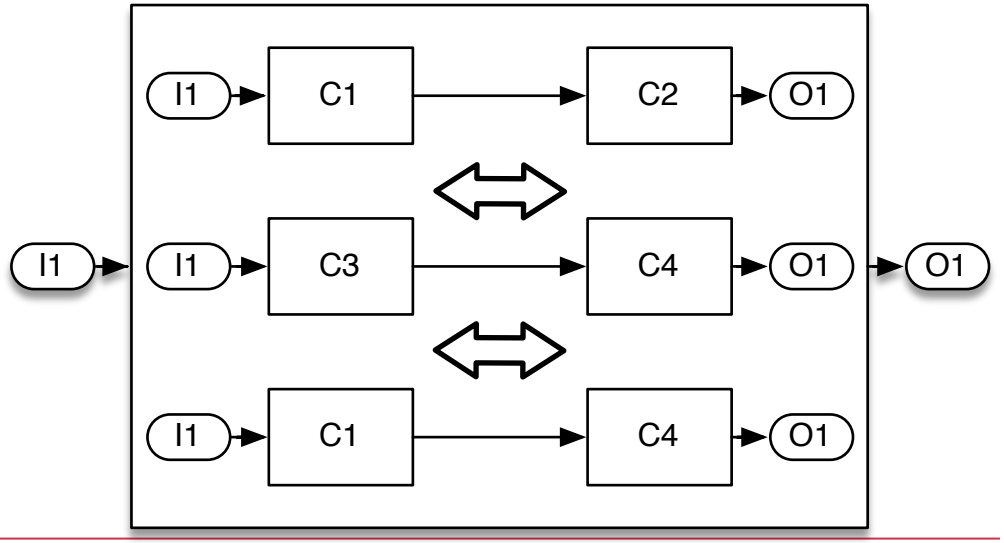
Beyond Model Diff & Model Clone Detection



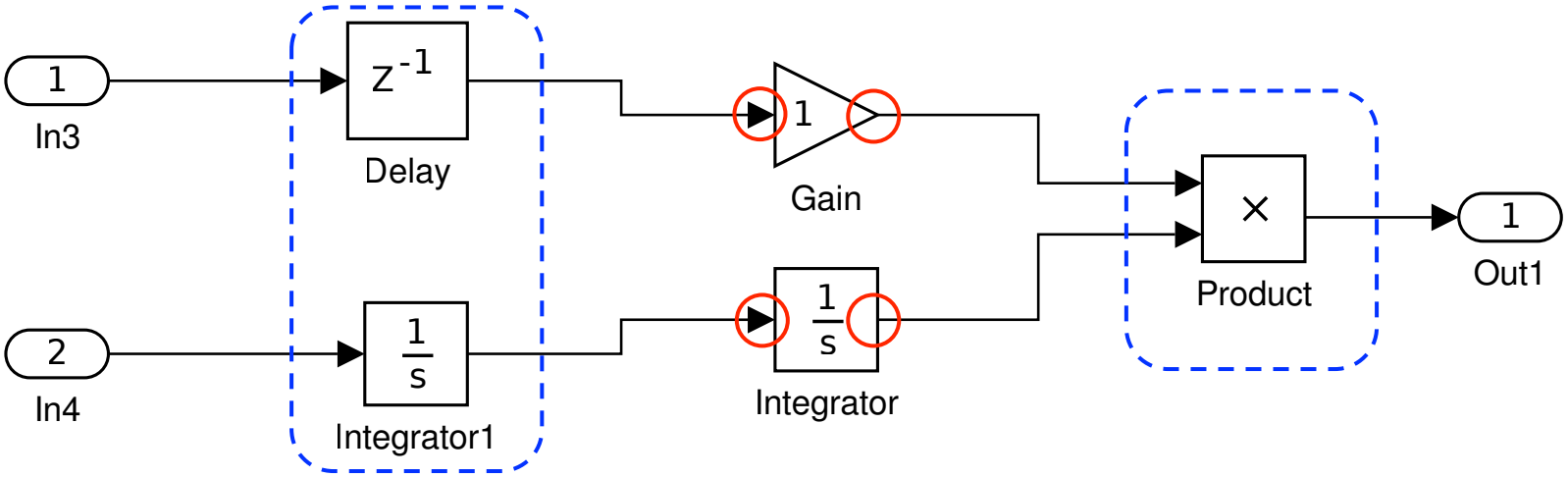
Differences and/or similarities detected by lots of existing tools

Beyond that….

…we add semantics
…put model elements in a family context



Context & Interfaces

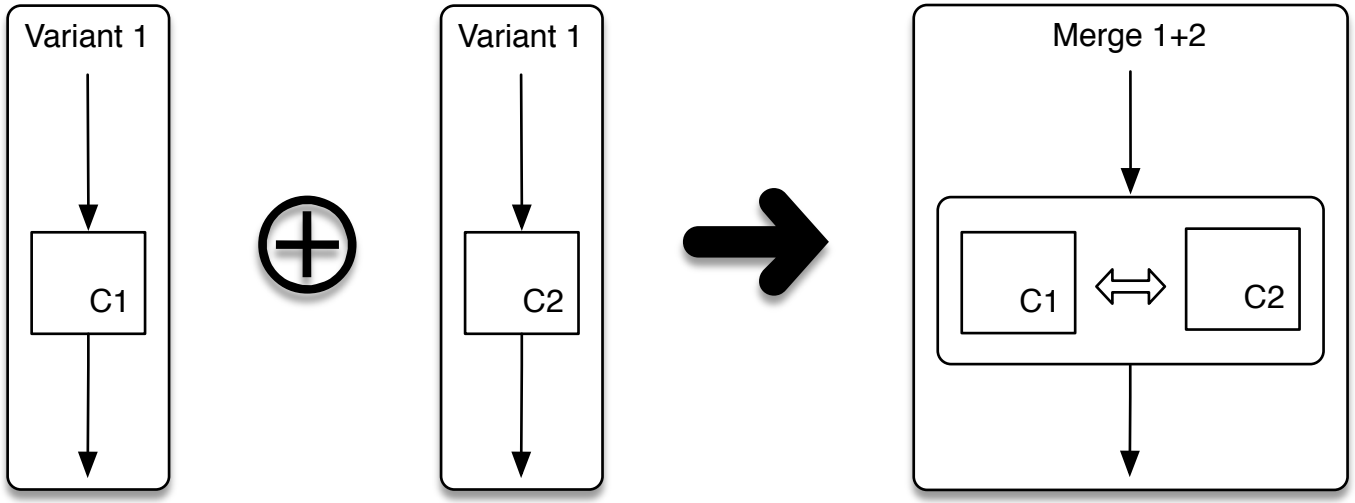


Context → model components, connected to component of interest

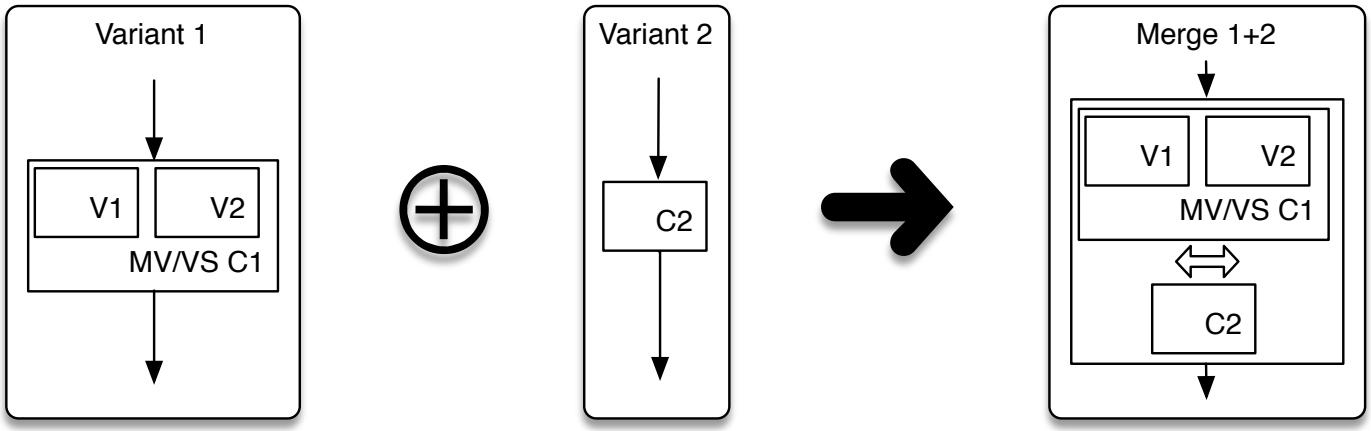
Interfaces → IN and OUT ports of component of interest

Alternatives

Different, mandatory components

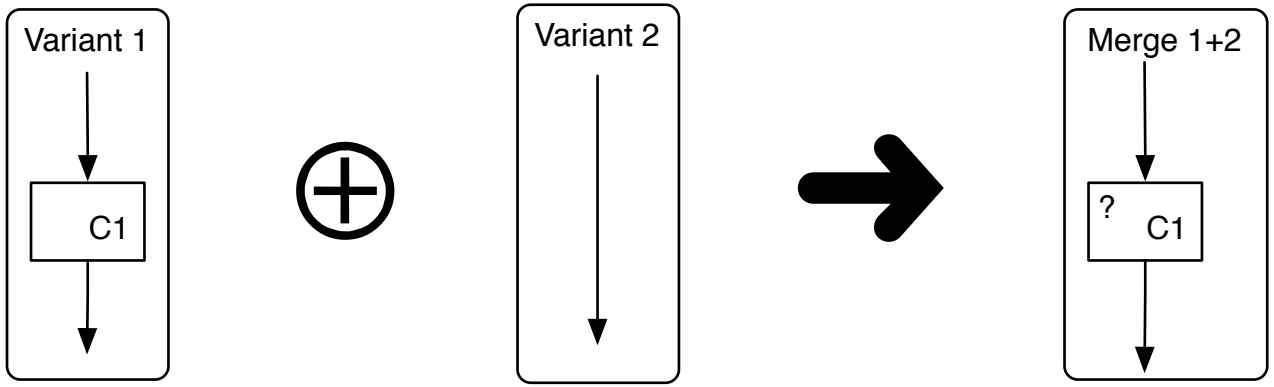


Variable sub-system and mandatory component

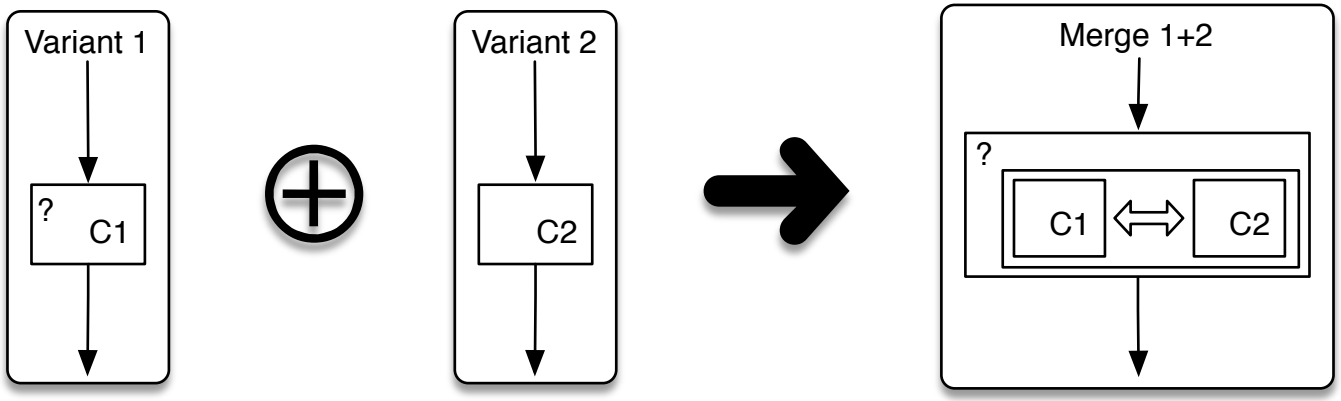


Optional Components

Optional and mandatory component

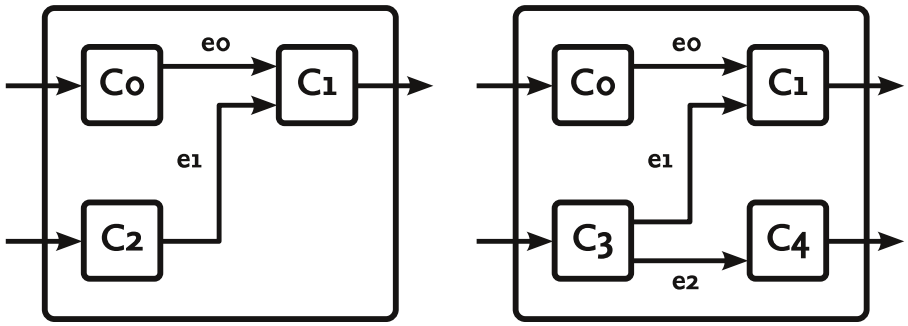


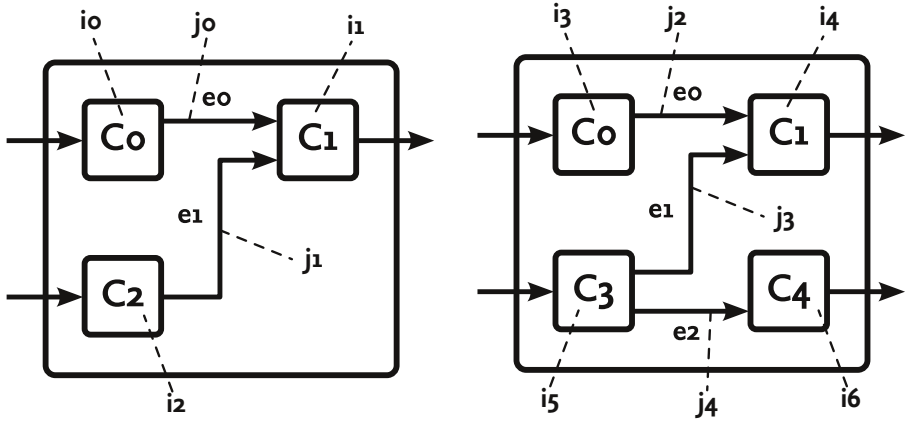
An optional alternative ;-)



Interface Variability

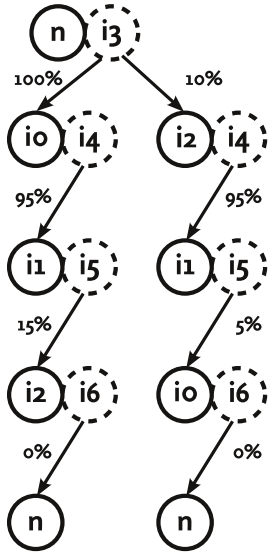
Wille et al., MAPLE 2013



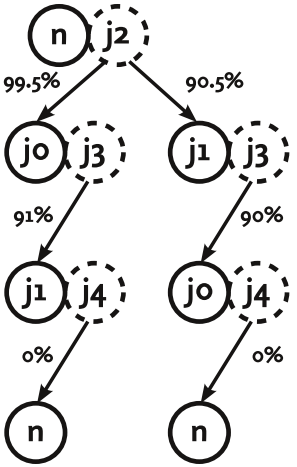


Compare trees for…

Components

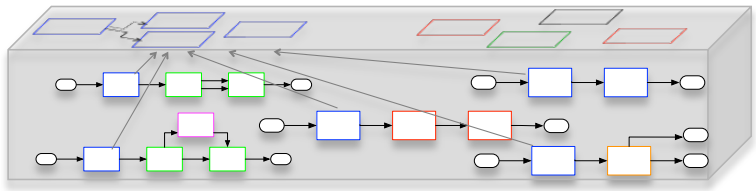


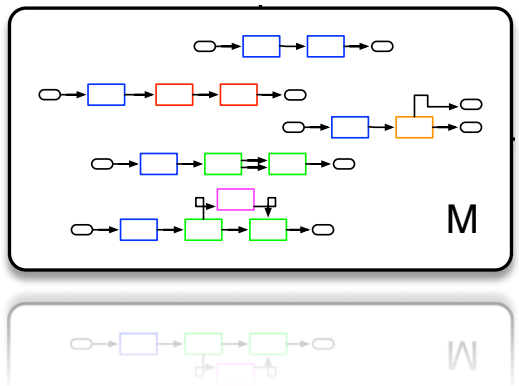
Connectors



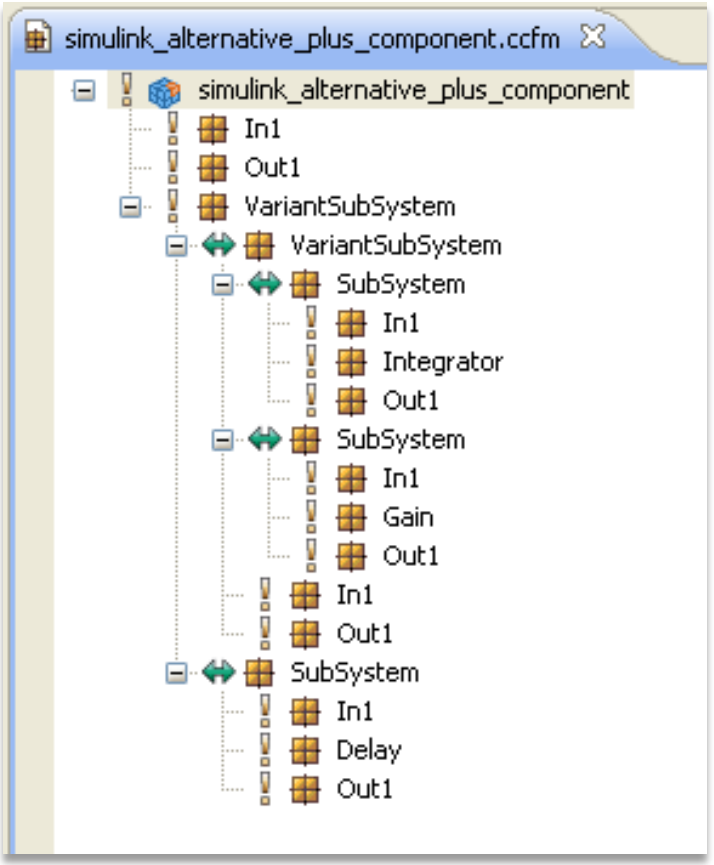
Putting the Pieces Together

Redundancy





Family Model



Summary

Questions…partly answered

Why does redundancy exists?

Where does it come from?

What does it tell us (under the hood)?

It's much (all?) about variability!!!

Does your mother know, you are here?

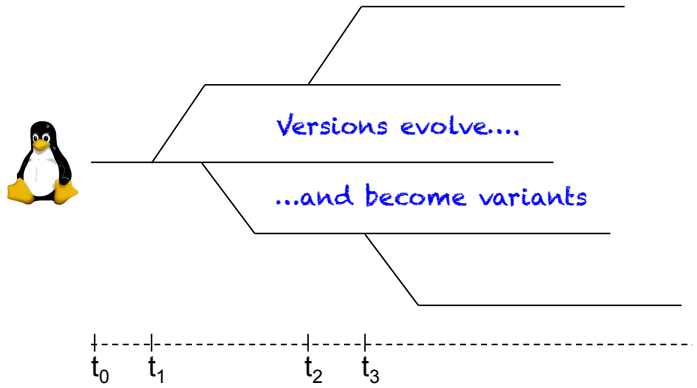
At least, she does not know the whole truth!!! But there's more…

…compositional testing and verification

…propagating patches/changes

…modular reasoning

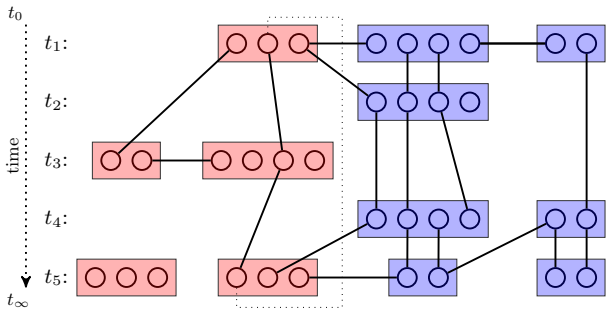
Clone-and-Own



Redundancy introduces Variability

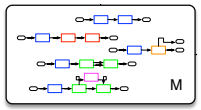
http://www.distrowatch.com

Analyzing Inter-System Clone Evolution



Legend: clone fragment — clone connection □ clone class ■ system 1 ■ system 2

Towards A Family of Models



- **Difficult to maintain**
- **Propagating changes → which models?**
- **Replication usually not documented**

Commonalities Differences