

Challenges in Developing a Software Architecture Evolution Tool as a Plug-In

Jeffrey M. Barnes and David Garlan
Institute for Software Research
Carnegie Mellon University

Slides and paper available for download at:
<http://www.cs.cmu.edu/~jmbarnes/papers/topi13.html>

Background

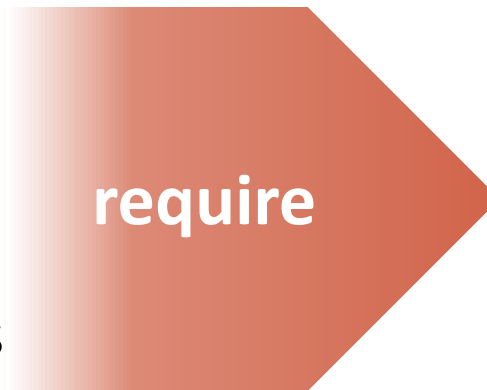
- *Architecture evolution* is central to software development

new technologies

new frameworks

new requirements

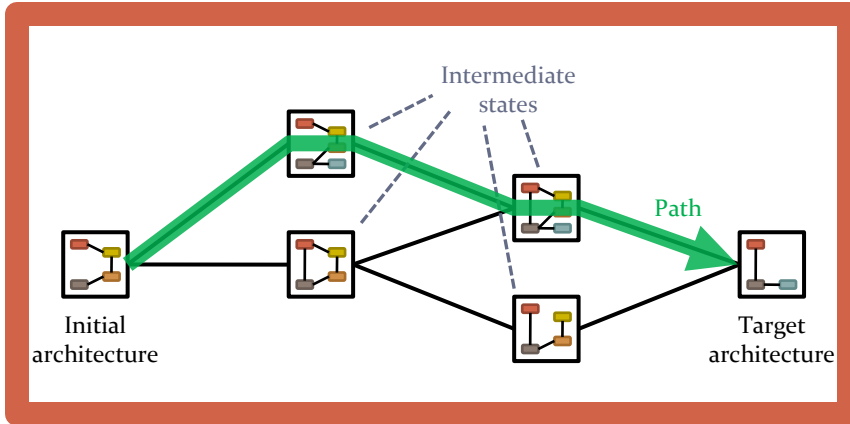
new market opportunities



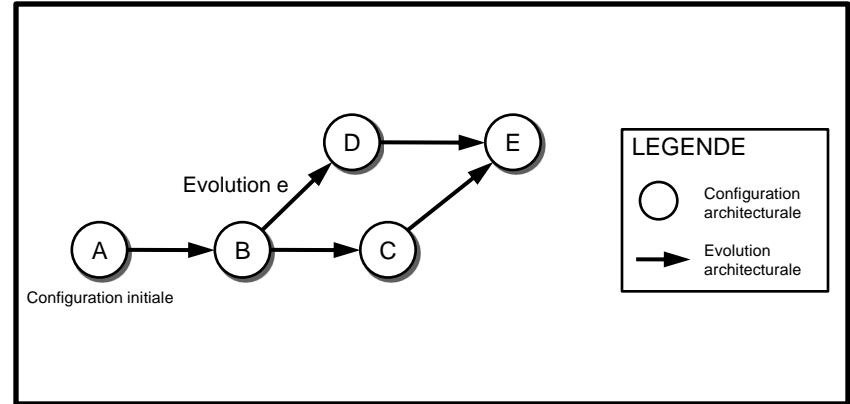
architectural
change

- At present, software architects have few tools to help plan and execute evolutions

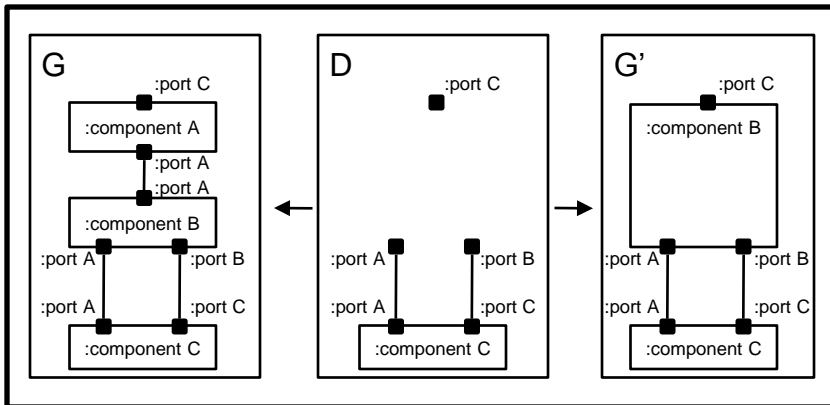
Background



J. M. Barnes, D. Garlan, et al.



O. Le Goer, D. Tamzalit, et al.

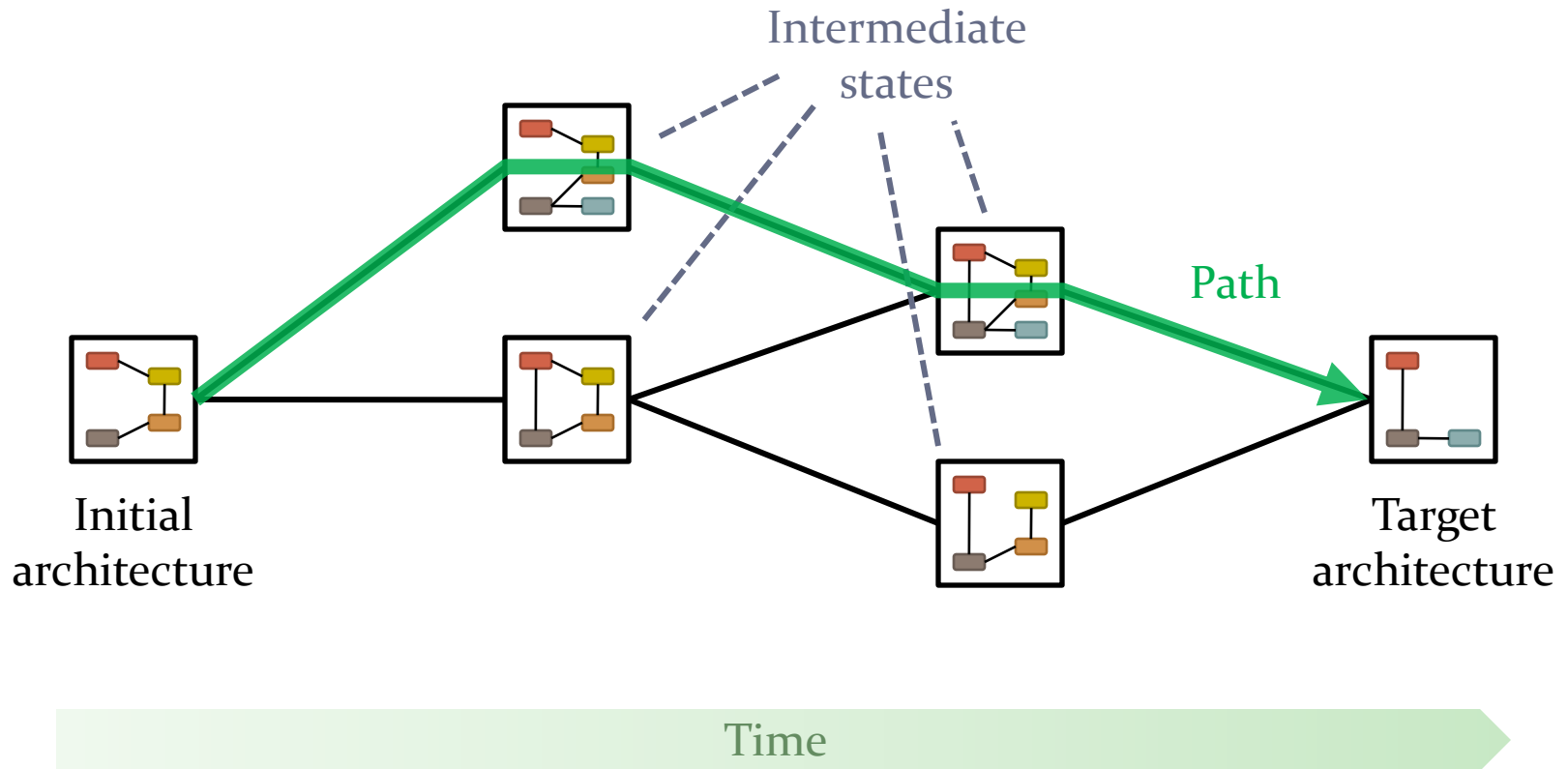


L. Grunske

		Release 1	Release 2	Release 3	Release 4	Release 5
Path #1	Cumulative value	75	132	175	216	243
	% of total value	30.86%	54.32%	72.02%	88.89%	100.00%
	Cumulative cost ($Ic_n + Rc_n$)	30	54.72	65.72	67.72	89.64
	% of total implementation cost	40.54%	71.72%	86.58%	89.28%	118.47%
Path #2	Cumulative value	0	0	47	126	243
	% of total value	0%	0%	19%	52%	100%
	Cumulative cost ($Ic_n + Rc_n$)	21	33	43	58	74
	% of total implementation cost	28.38%	44.60%	58.11%	78.38%	100%

N. Brown et al.

Our Model of Architecture Evolution

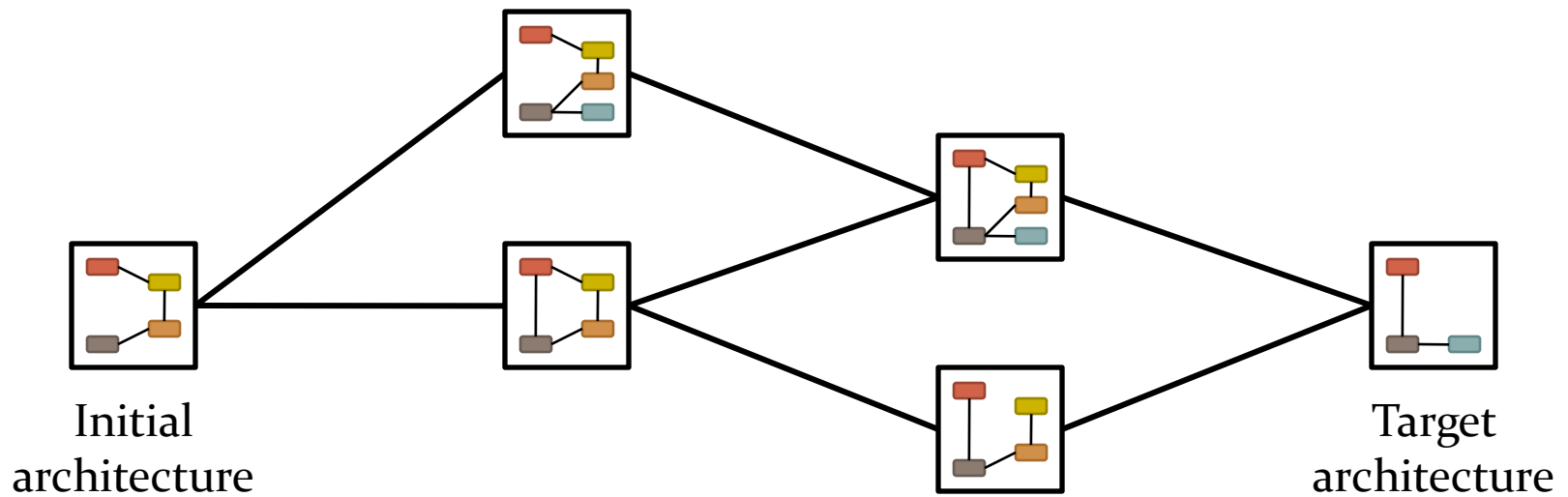


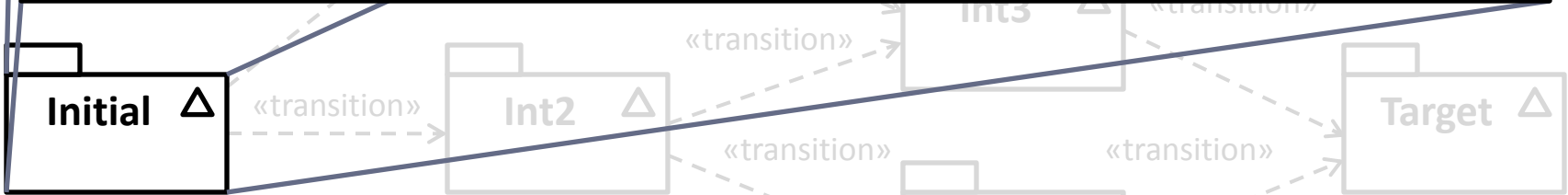
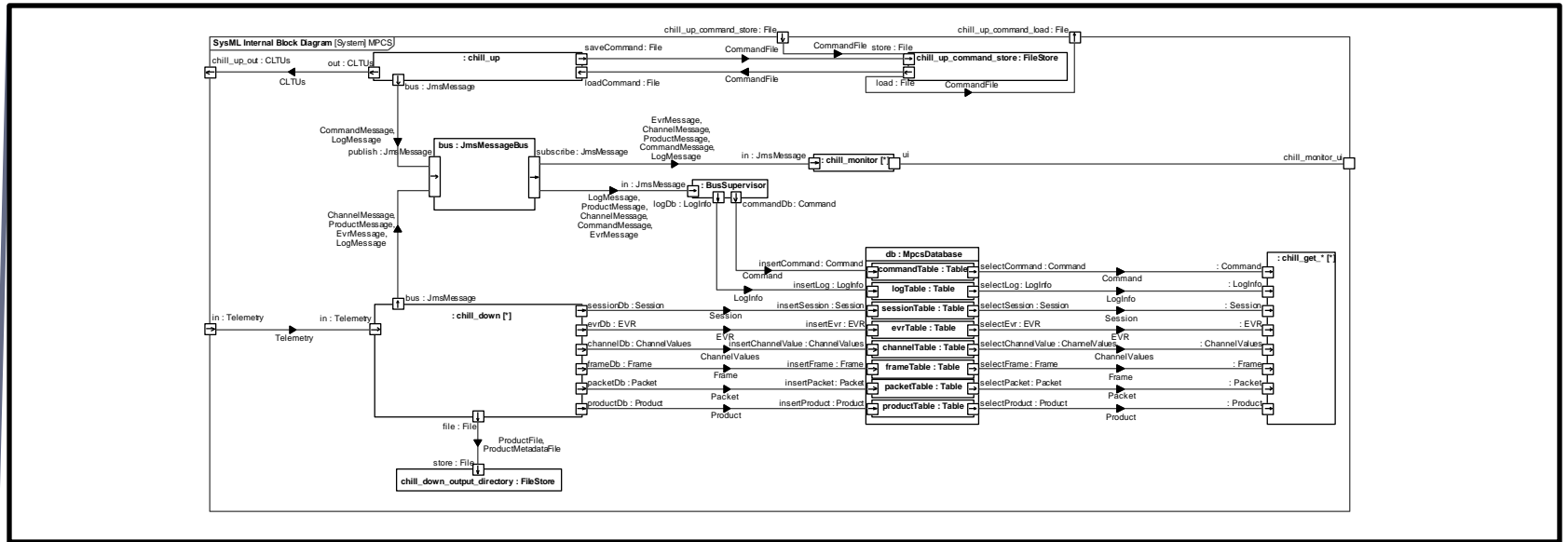
Our Previous Tooling Work

- Previously* developed a partial prototype as part of a case study at NASA JPL
- Case study aim was to model an evolution at JPL, using modeling languages and tools in use there
- Developed facilities to support modeling architecture evolution in MagicDraw
 - Representational conventions for modeling architecture evolution in UML
 - Transformation macros to effect evolution operators
 - UML models of particular intermediate states

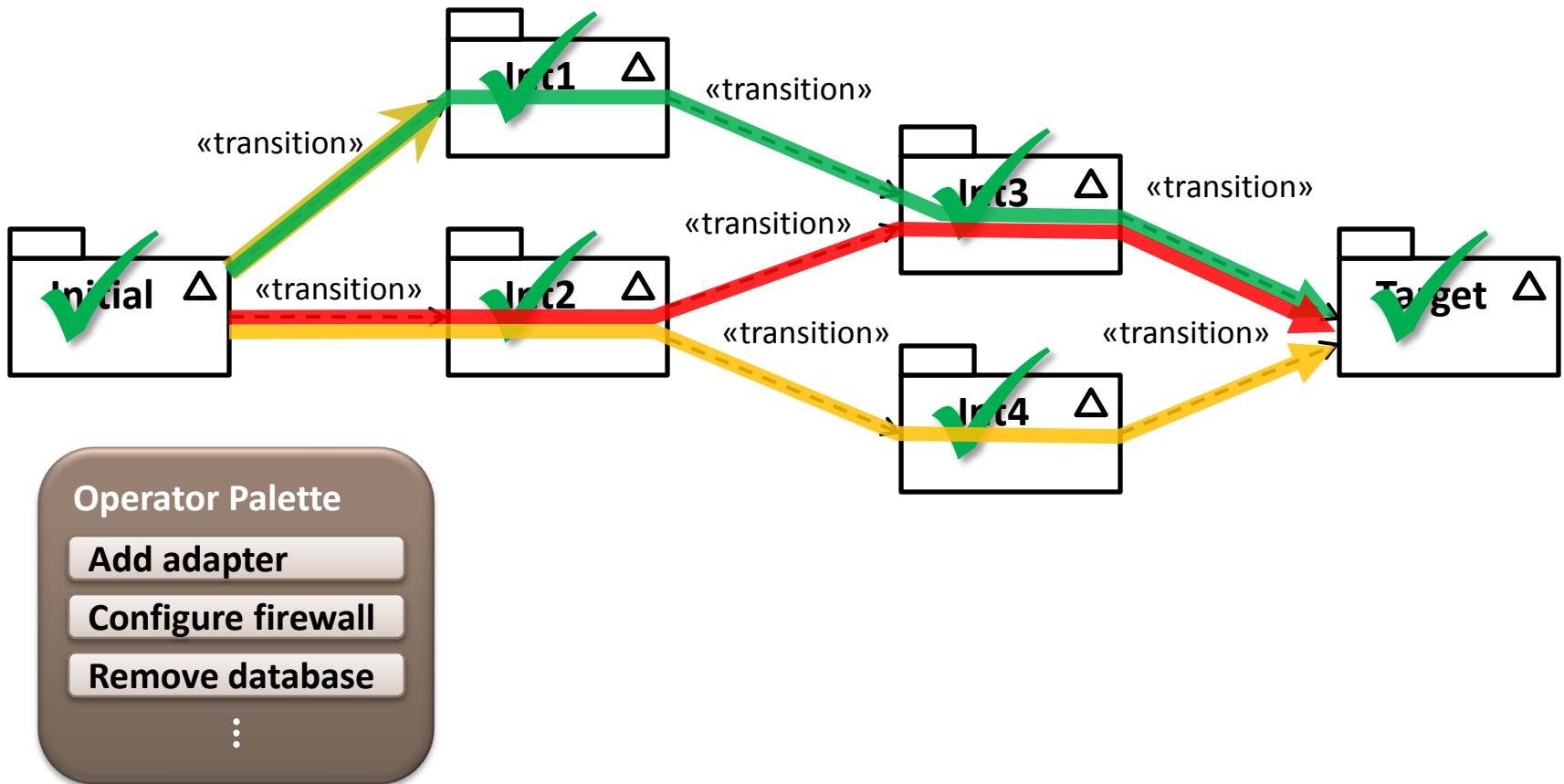
*J. M. Barnes, “NASA’s Advanced Multimission Operations System: A case study in software architecture evolution,” in Proc. QoSA’12, 2012, pp. 3–12.

Vision: A Plug-In for Architecture Evolution Planning





Vision: A Plug-In for Architecture Evolution Planning



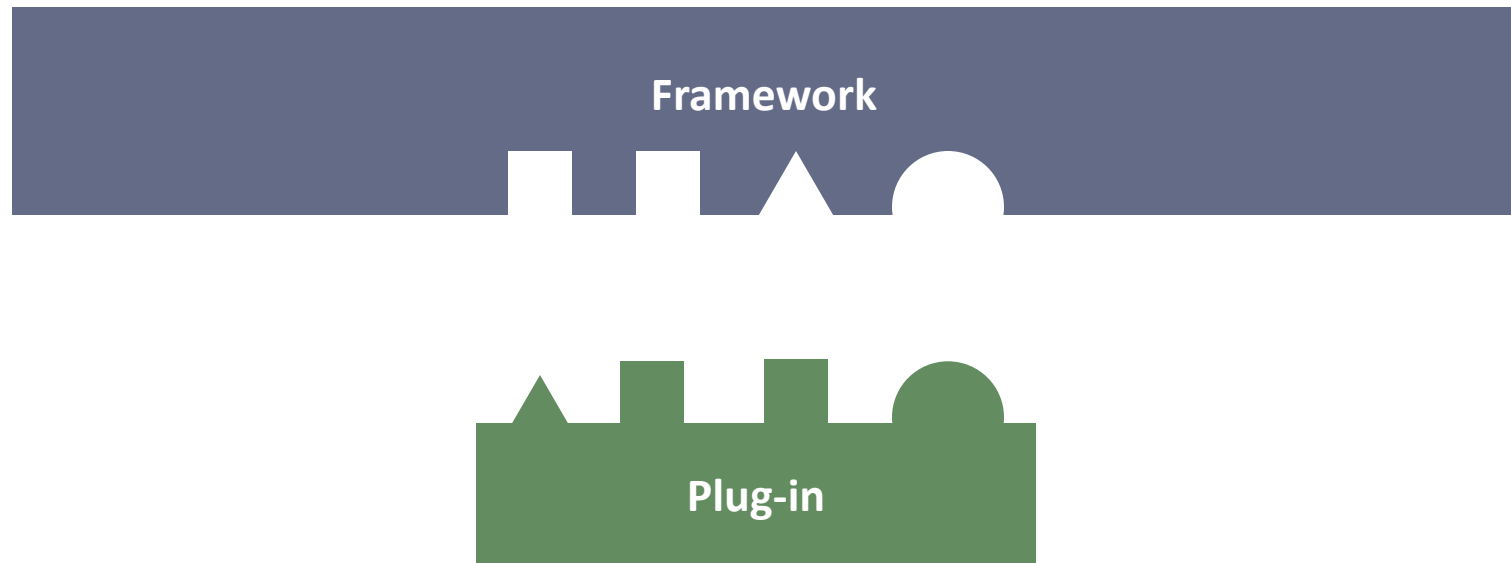
Development Process

- Development was carried out by a team of two master's students, directed by the authors
- Prototype development proceeded in stages designed to produce “feature prototypes”

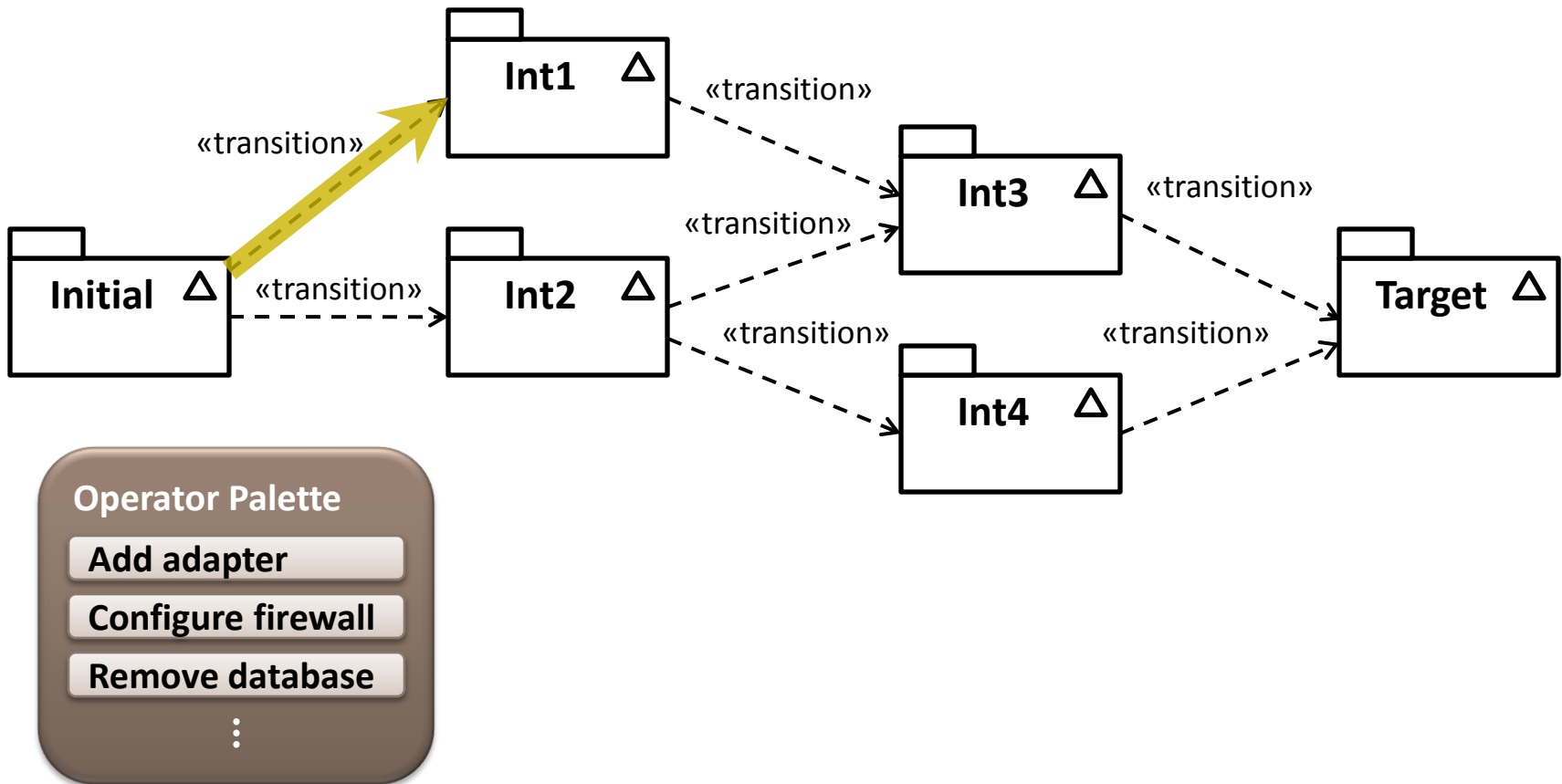
Feature Prototypes

1. Model manipulation
 2. Identification of evolution paths
 3. Interface for applying operators
 4. Operator parser
 5. Metadata handling
 6. Positioning of presentational elements
 7. Evolution path constraints
- Final phase: Integration

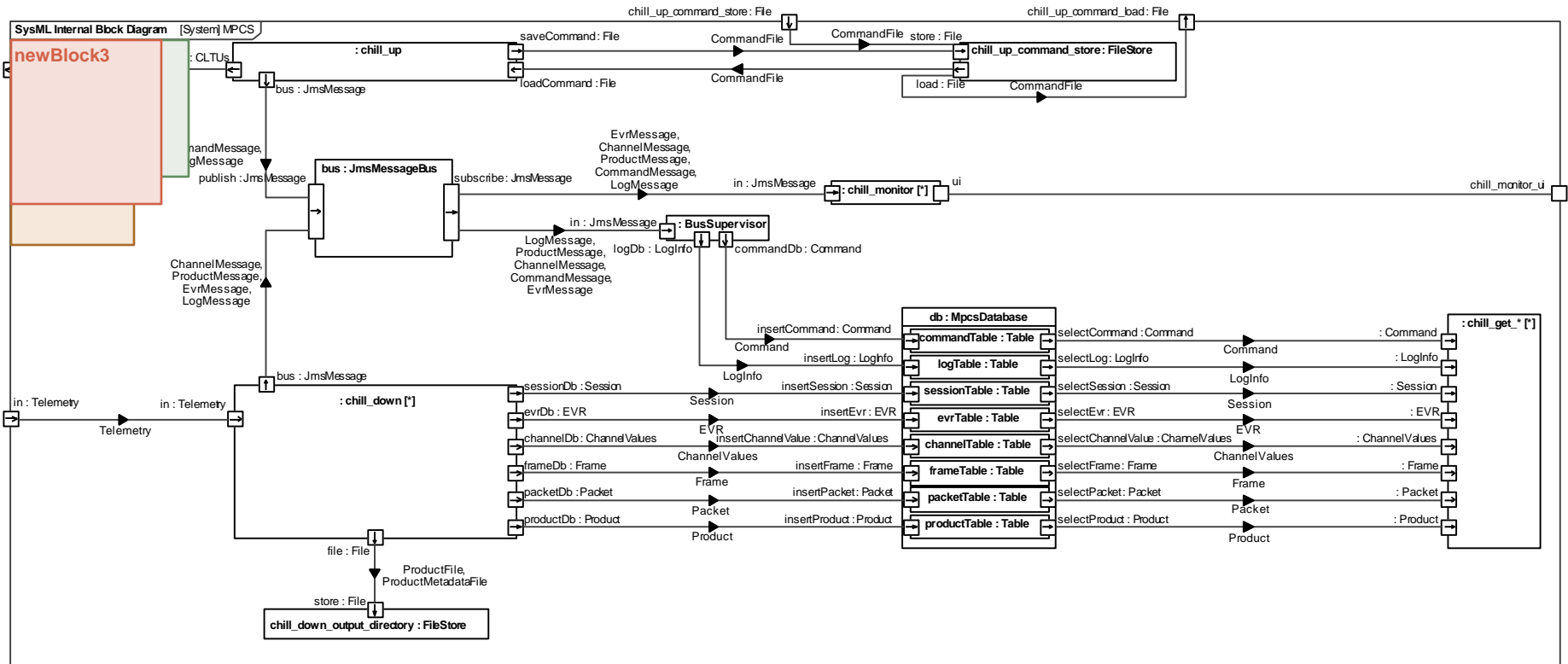
Challenge: Understanding the Kinds of Variation that the Framework Supports



Challenge: Controlling the User Interface



Challenge: Manipulating Presentational Elements



Conclusion

- Learned a number of lessons about factors that can simplify or complicate development of an architecture evolution plug-in
- Many of these lessons have broader application
 - Especially for other tools that must compare and analyze multiple software architectures
- Still lots of future work to do on tooling issues for architecture evolution