Reflections on 20 Years of Architecture for Distributed Real-time & Embedded Systems

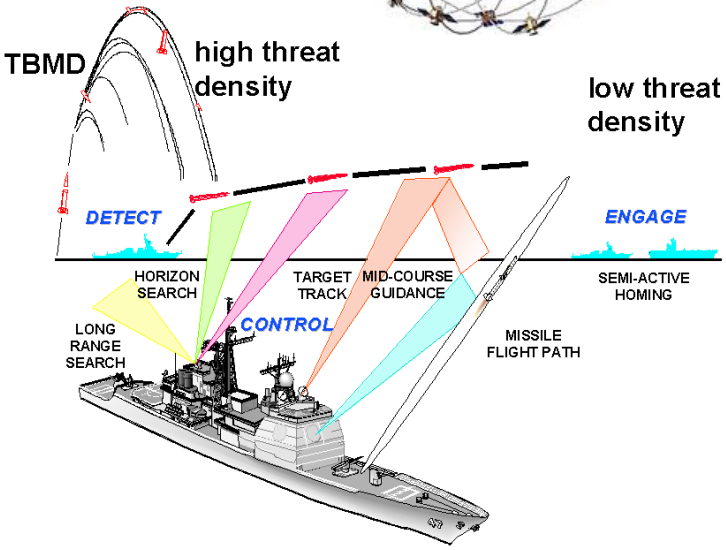
Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

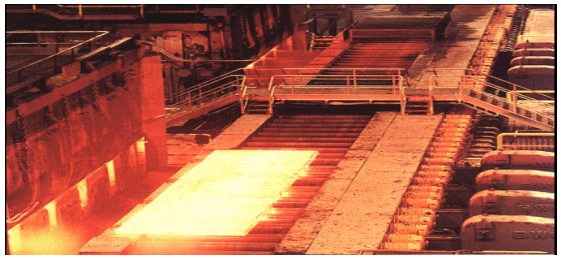
Professor of EECS
Vanderbilt University
Nashville, Tennessee

SATURN Conference, May 9th, 2012

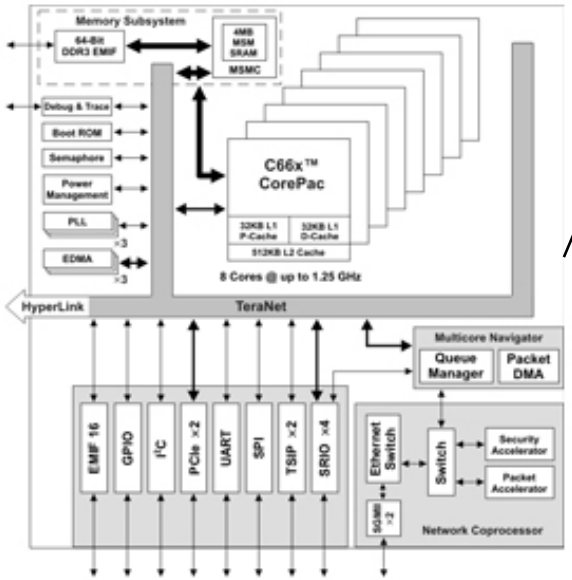
Distributed Real-time & Embedded (DRE) Systems

In DRE systems the “right answer” delivered too late becomes the “wrong answer”

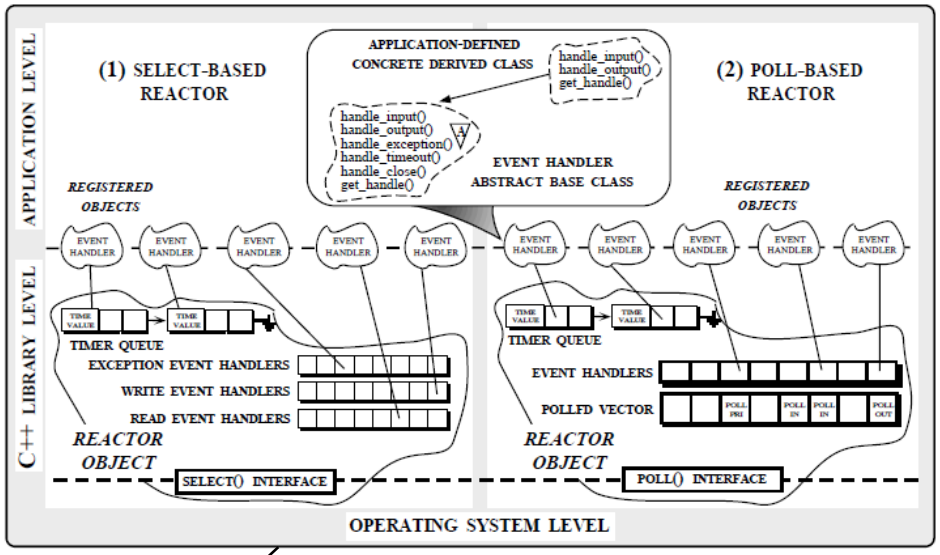




DRE System Architecture: ~20 Years Ago

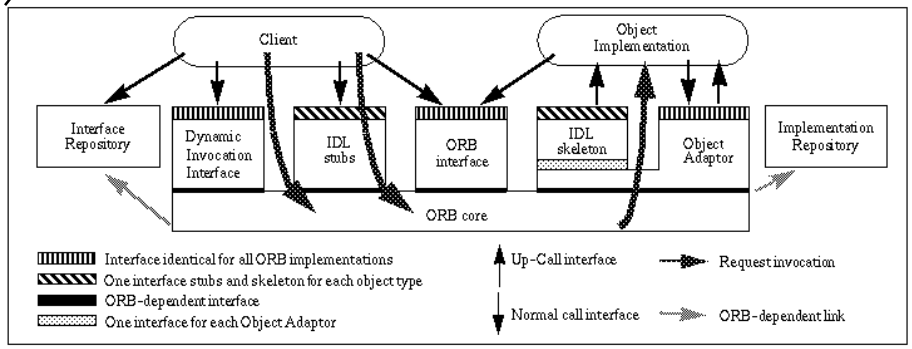


Overly hardware -centric

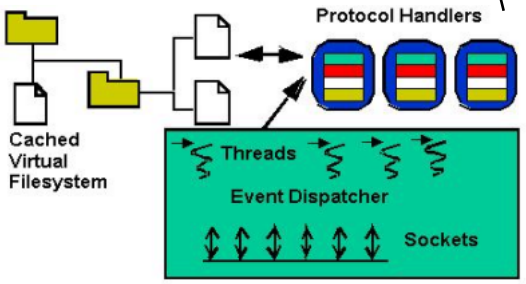


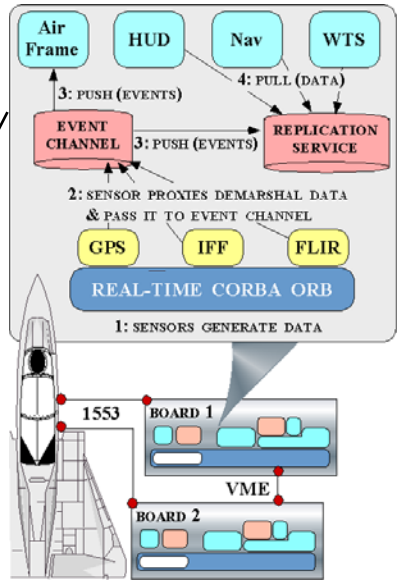
Conflated design & implementation

Abstracted away from key quality-of-service properties



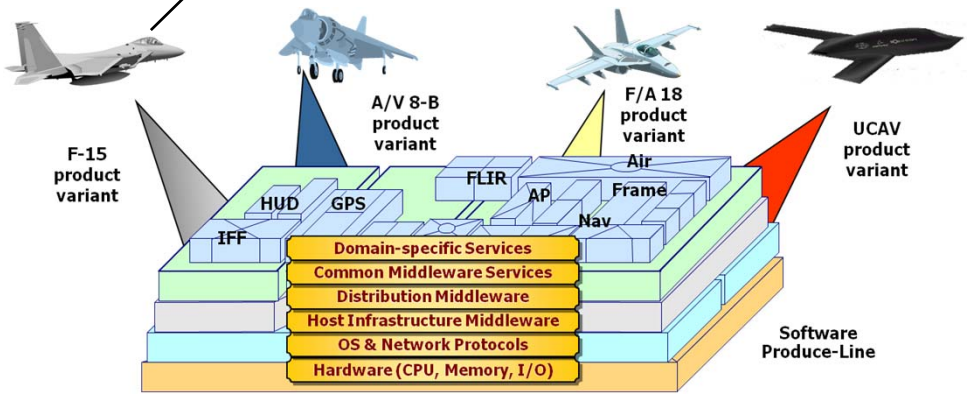
Limited reuse across domains



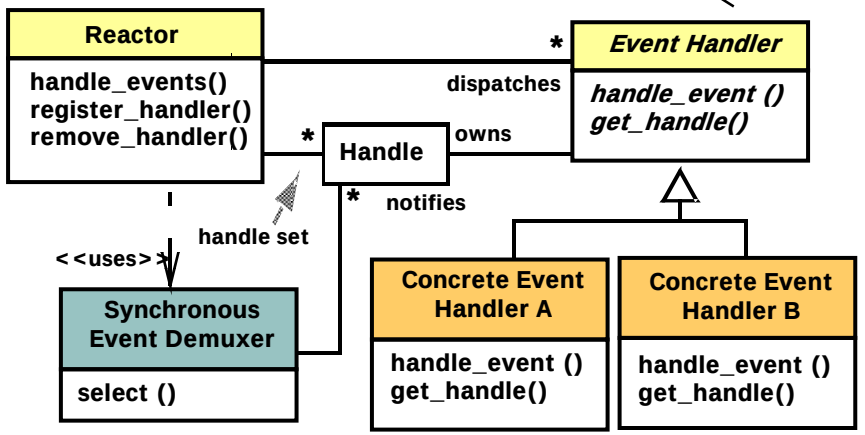


DRE System Architecture: ~10 Years Ago

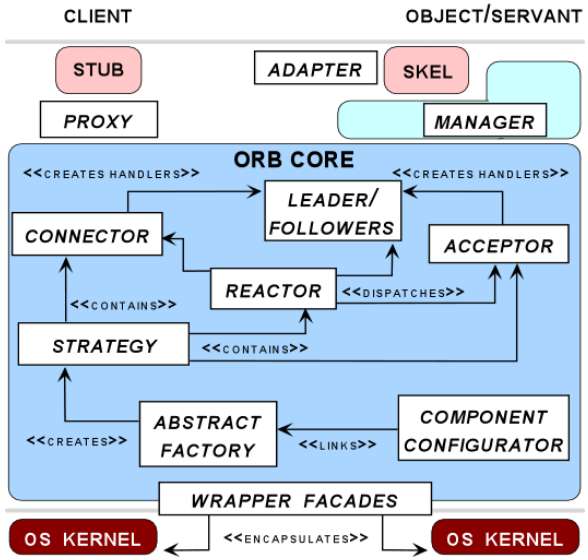
Pattern-Oriented Software Architecture



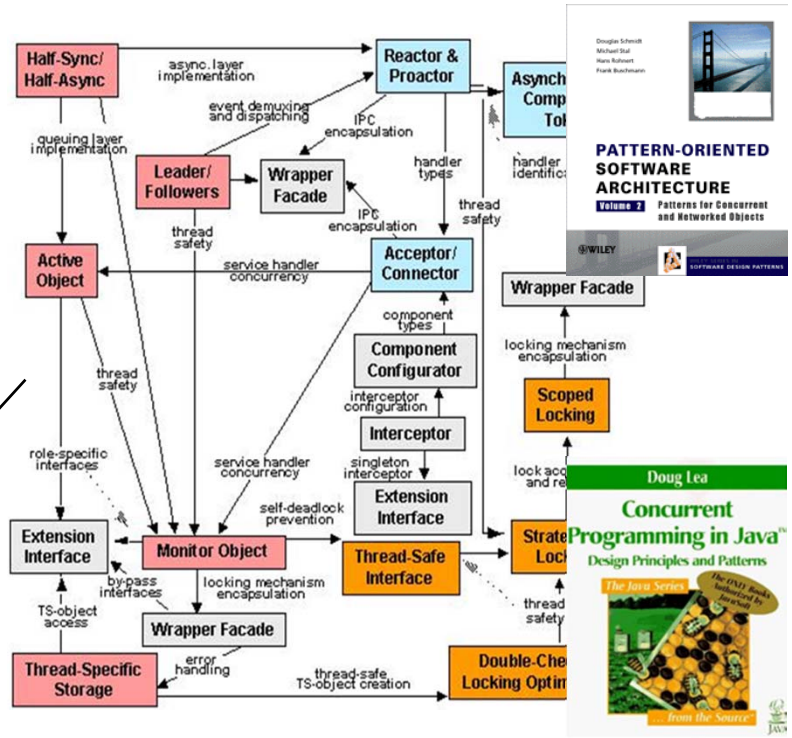
Patterns decouple design & implementation



Optimization principle patterns address key quality-of-service properties

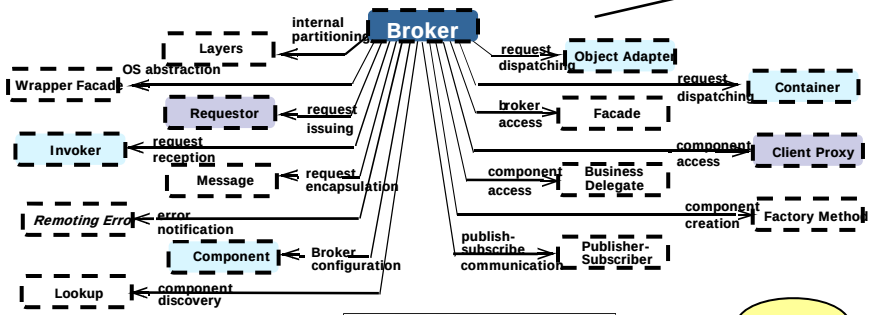


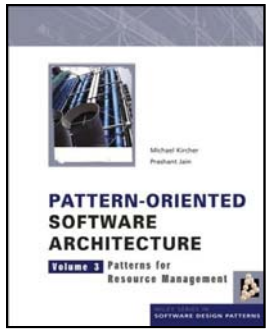
Pattern collections support multiple DRE system domains

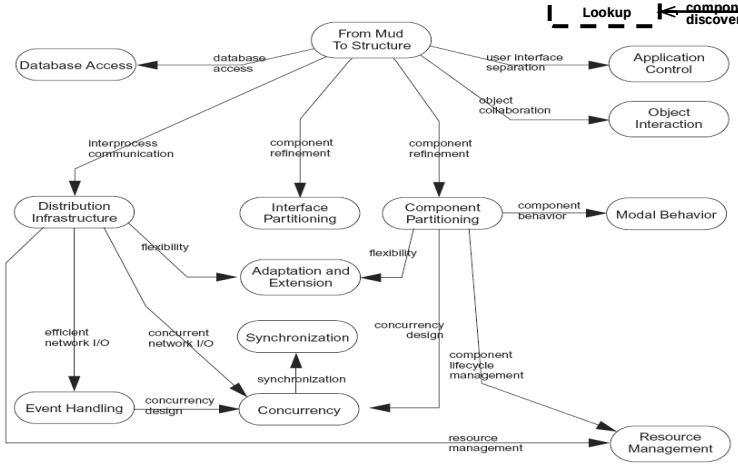


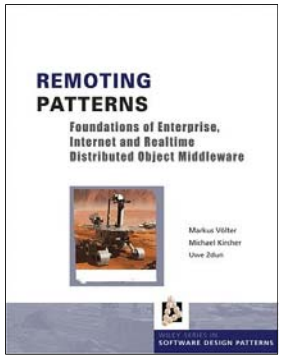
DRE System Architecture: Past ~5 Years

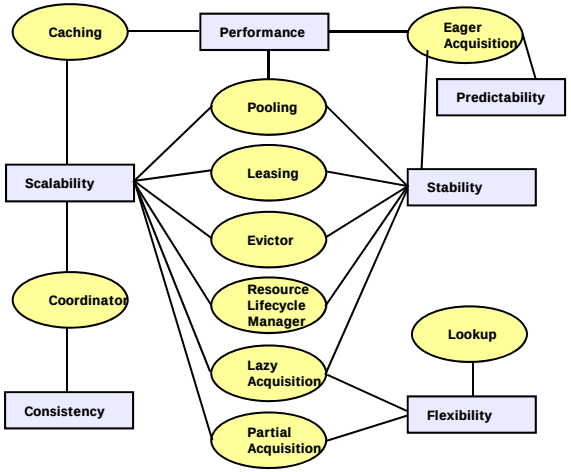
Pattern languages for distributed computing & resource management

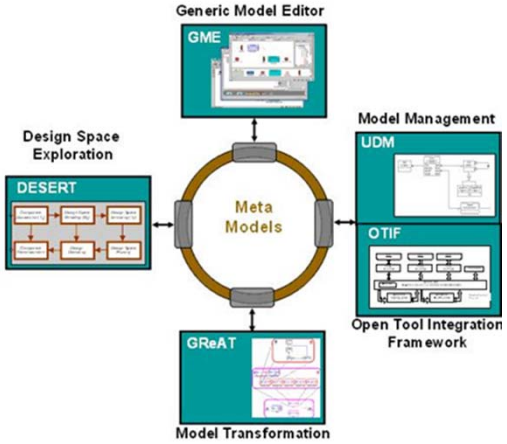




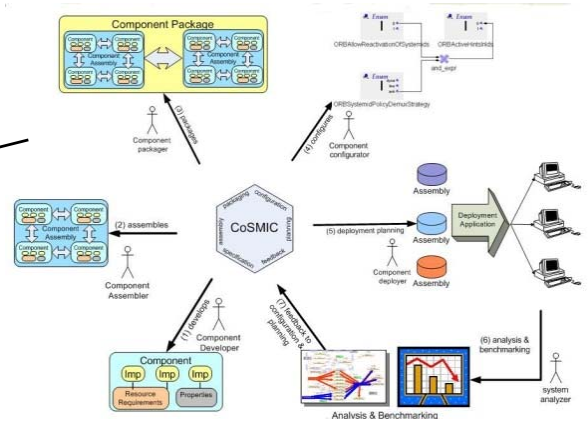






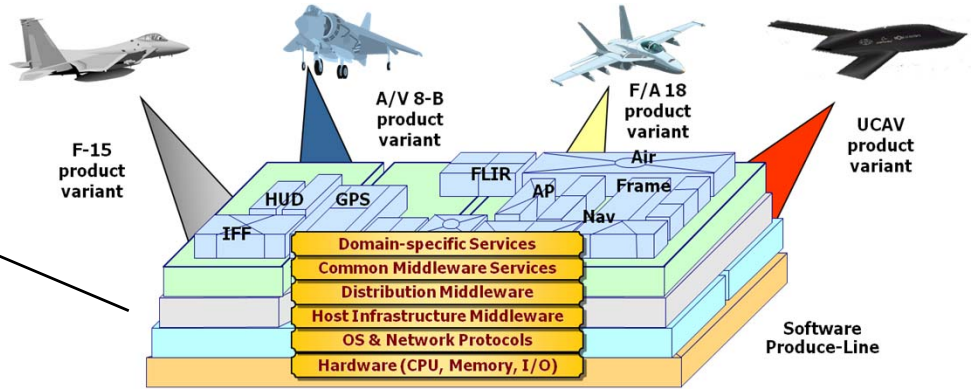


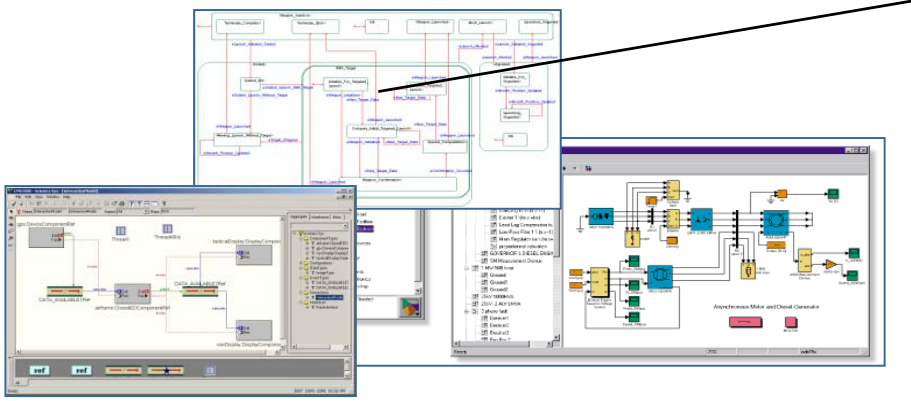
Model-driven engineering tools automate key patterns for DRE systems



Concluding Remarks

Careful study & leveraging of good patterns significantly improves architecture skills





Automation via model-driven engineering tools is helpful, but not (yet) a substitute for experience/insight

Our understanding of the key patterns necessary to architect net-centric DRE systems-of-systems is still in its infancy