

High-confidence Software for Cyber Physical Systems

Sherif Abdelwahed
Mississippi State Univ.
Starkville, MS
sherif@ece.msstate.edu
www.ece.msstate.edu/~sherif

Nagarajan Kandasamy
Drexel University
Philadelphia, PA
kandasamy@cbis.ece.drexel.edu
www.ece.drexel.edu/~kandasamy

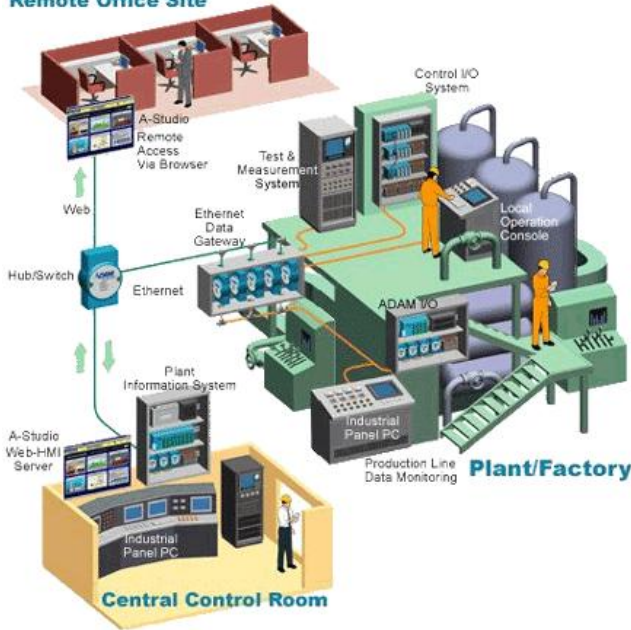
Aniruddha Gokhale
Vanderbilt University
Nashville, TN
a.gokhale@vanderbilt.edu
www.dre.vanderbilt.edu/~gokhale

Mississippi State
UNIVERSITY



Traits of Cyber Physical Systems

Remote Office Site



- Network-centric, dynamic, large-scale “systems of systems”
- Service-oriented architecture of distributed collaborating services
- Stringent *simultaneous* QoS demands, e.g., “never die,” time-critical, secure.
- Highly diverse, complex, integrated & autonomous application domains
- On demand computing needs

Key Requirements for High Confidence Software

- **Trustworthiness** - delivering multiple, simultaneous QoS
- **Autonomicity** – self healing, self configuring, self optimizing
- **Analyzability** – amenable to validation and verification

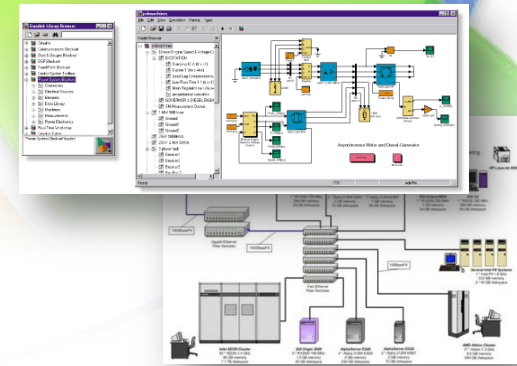
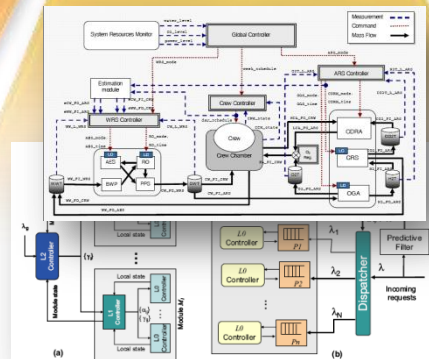
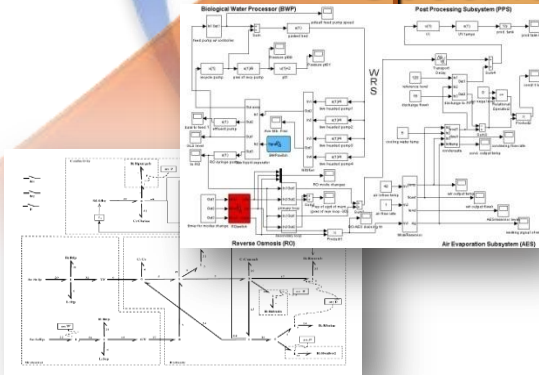
The Model-based Approach

High-confidence Software
for Cyber Physical Systems

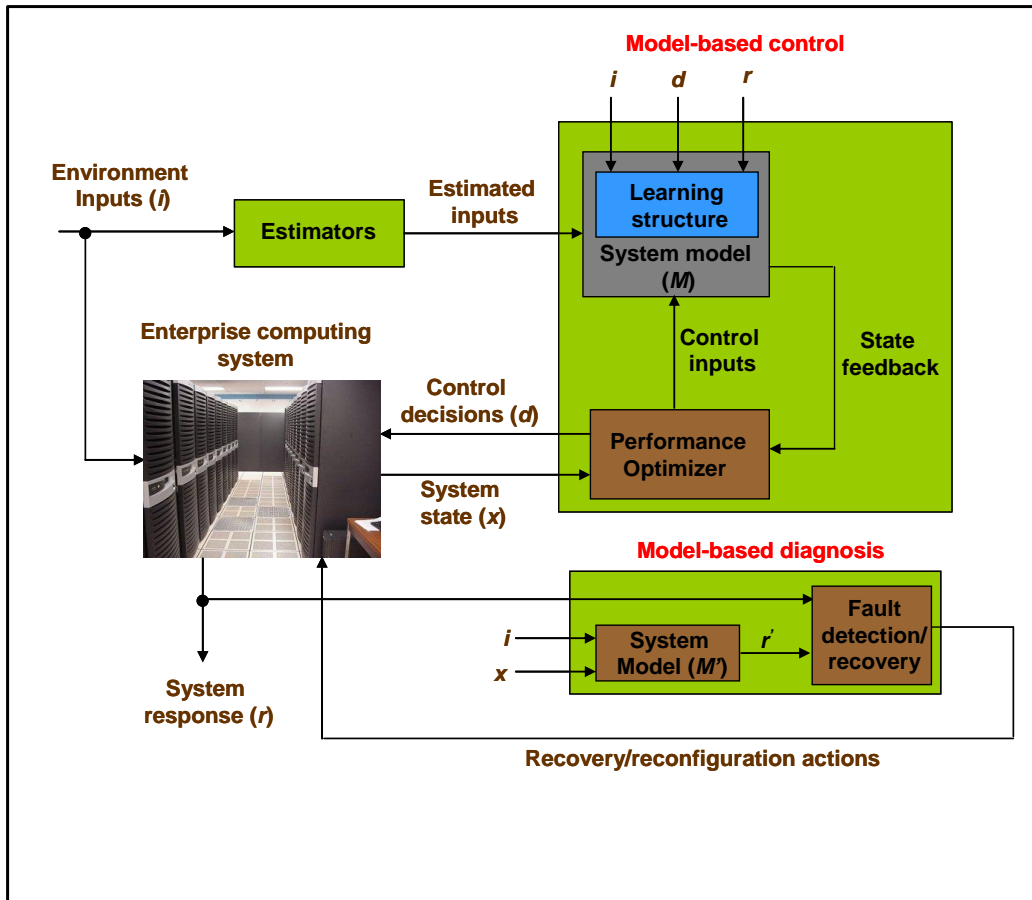
Operational
Models and
Specifications

Model-based
Adaptation
Technology

Integrated
Development
Infrastructure



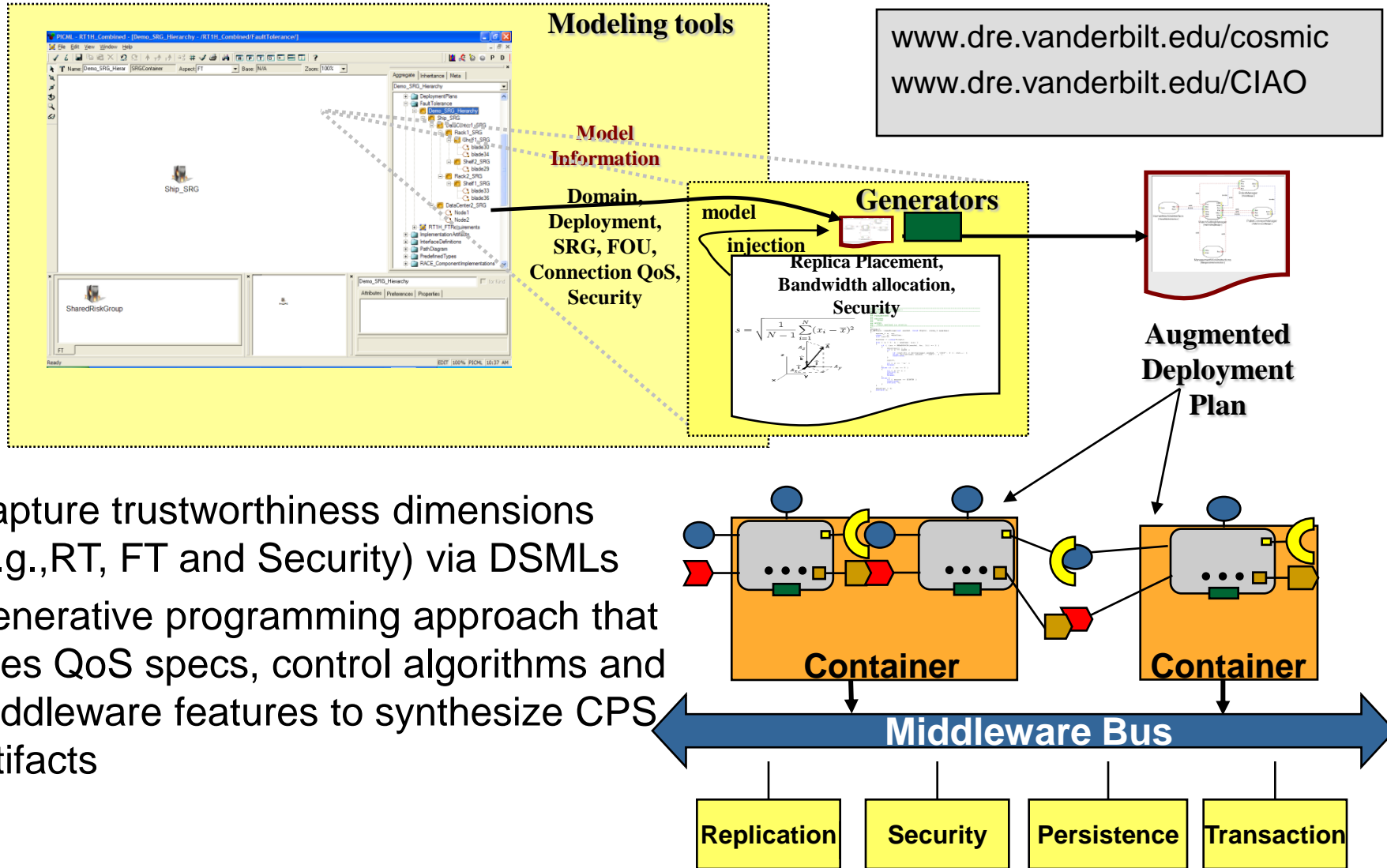
Step 1. Control-based Framework for Self-management



- System management tasks are posed as control/optimization problems and solved under dynamic and uncertain operating conditions
- Online parameter tuning and model-learning techniques can be integrated within the control framework to improve the quality of partially specified system models as well as adapt to changes in the system model itself over time
- Diagnosis algorithms will detect, isolate, and estimate the state of corrupted hardware and software components using concepts from continuous and discrete-event diagnosis, and consistency-based causality analysis.

Focus is on developing algorithms to realize incorruptible and self-healing CPSs via a combination of control and diagnostics

Step 2. MDE Tool Chain

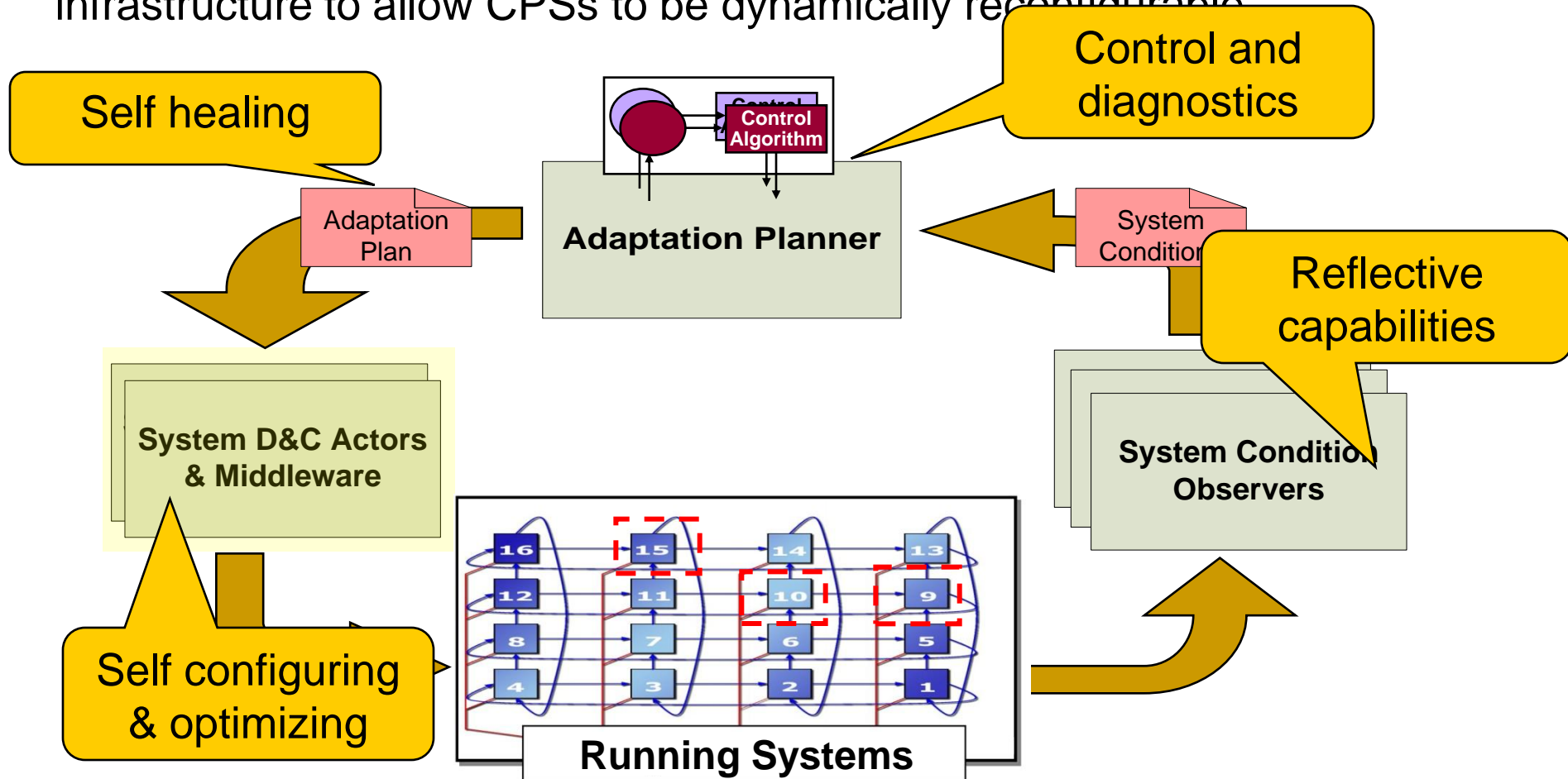


- Capture trustworthiness dimensions (e.g., RT, FT and Security) via DSMLs
- Generative programming approach that uses QoS specs, control algorithms and middleware features to synthesize CPS artifacts

Focus is on resolving accidental complexities and automating system configuration, deployment, adaptation and conducting analyses.

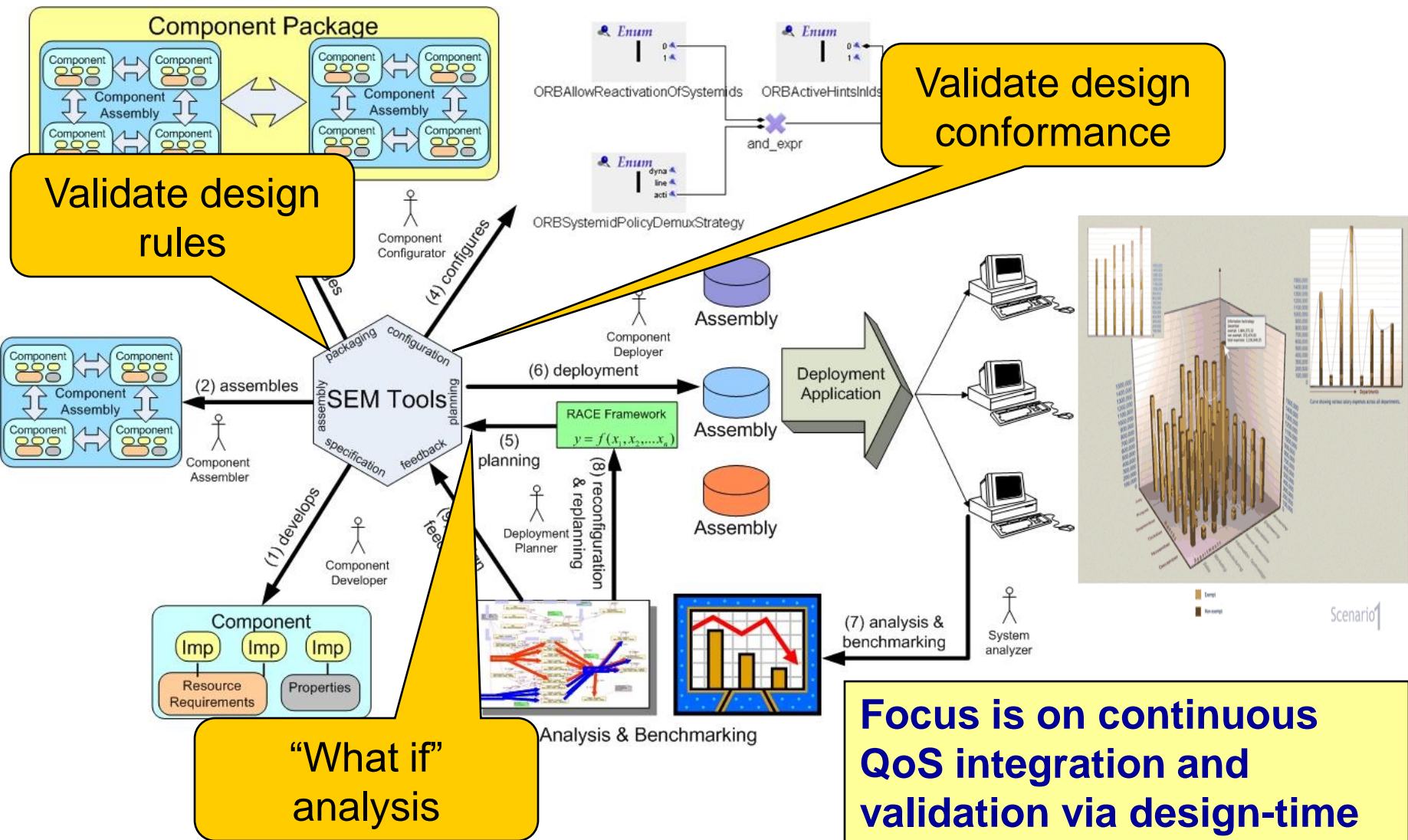
Step 3. Trustworthy Middleware Framework

- Decouple system adaptation policy from system application code & allow them to be changed independently from each other
- Decouple system deployment framework & middleware from core system infrastructure to allow CPSs to be dynamically reconfigurable



Focus is on realizing a scalable, trustworthy runtime environment.

Step 4. System Execution Modeling Tools



www.dre.vanderbilt.edu/cosmic
www.dre.vanderbilt.edu/CUTS