A Model-Based Design Perspective on Challenges and Opportunities in Automated Software Certification

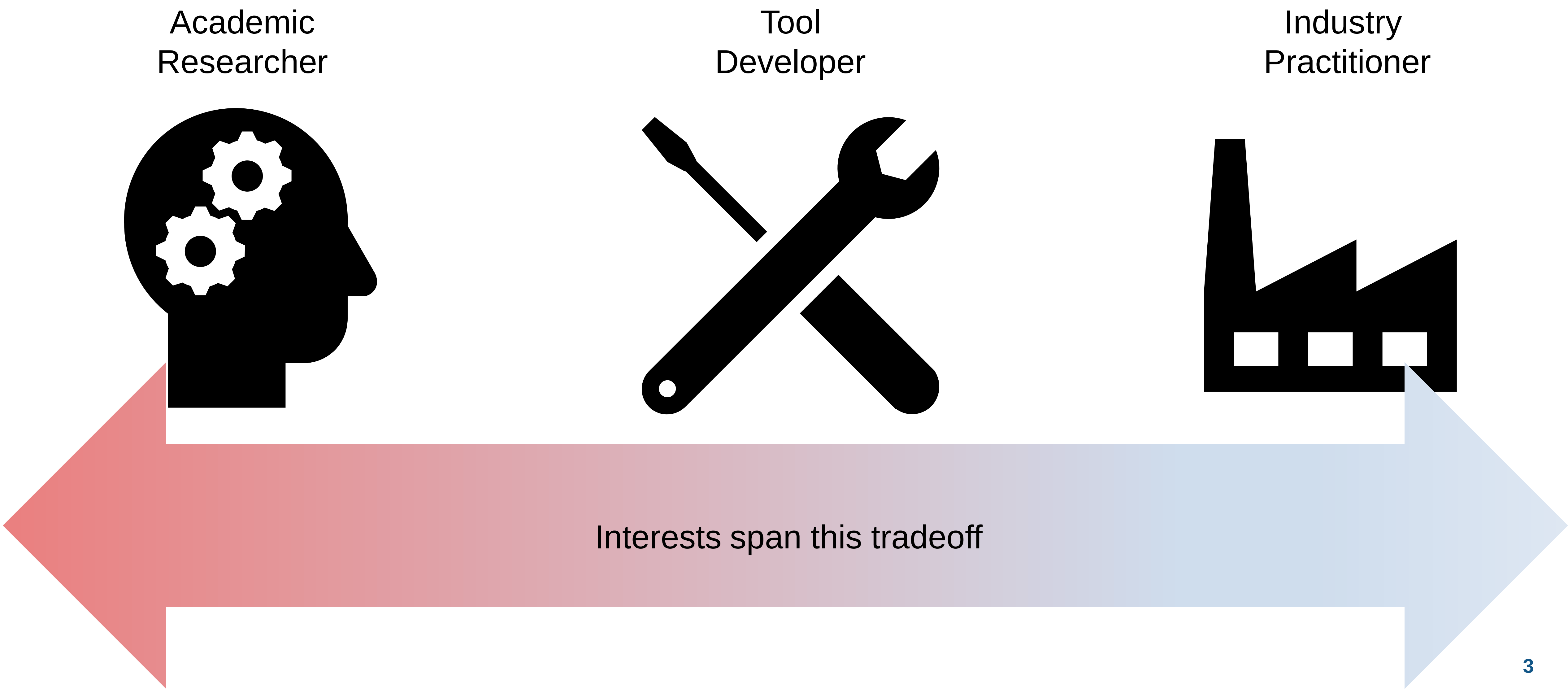
Akshay Rajhans, PhD
Principal Research Scientist
arajhans@mathworks.com
https://arajhans.github.io

20th Software Certification Consortium Steering Committee Meeting, Annapolis, MD, May 2019
THEME: TO WHAT EXTENT CAN AUTOMATION HELP IN CERTIFICATION?

About me

- 'CPS' Practitioner before it was called CPS
  - Embedded controls for diesel engine applications
  - Programmable logic controller for industrial automation
- CPS Research at the intersection of
  - Model-based design and analysis
  - Formal methods
  - Software and system architecture
- CPS Research Scientist at MathWorks

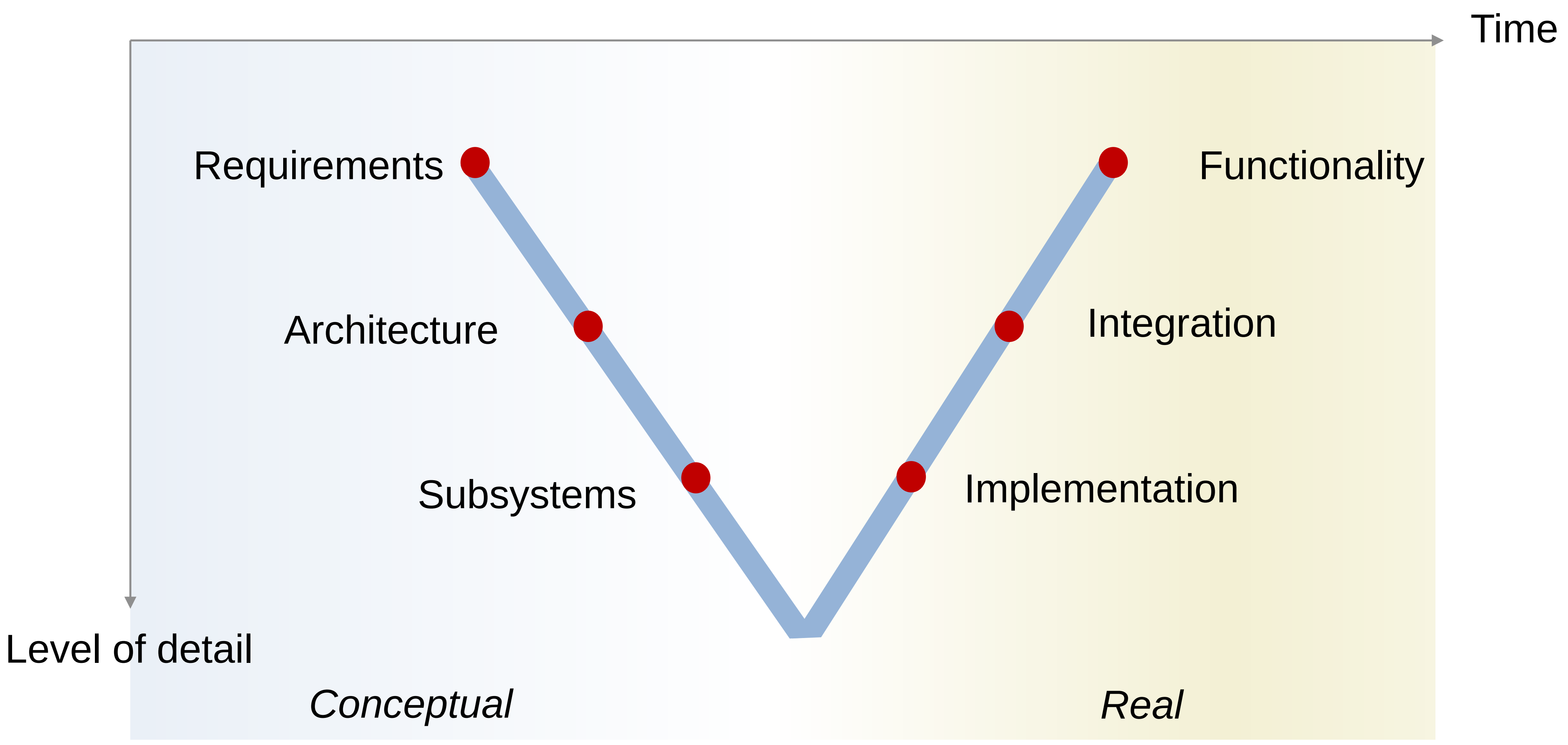
Perspective shaped by my personal career trajectory



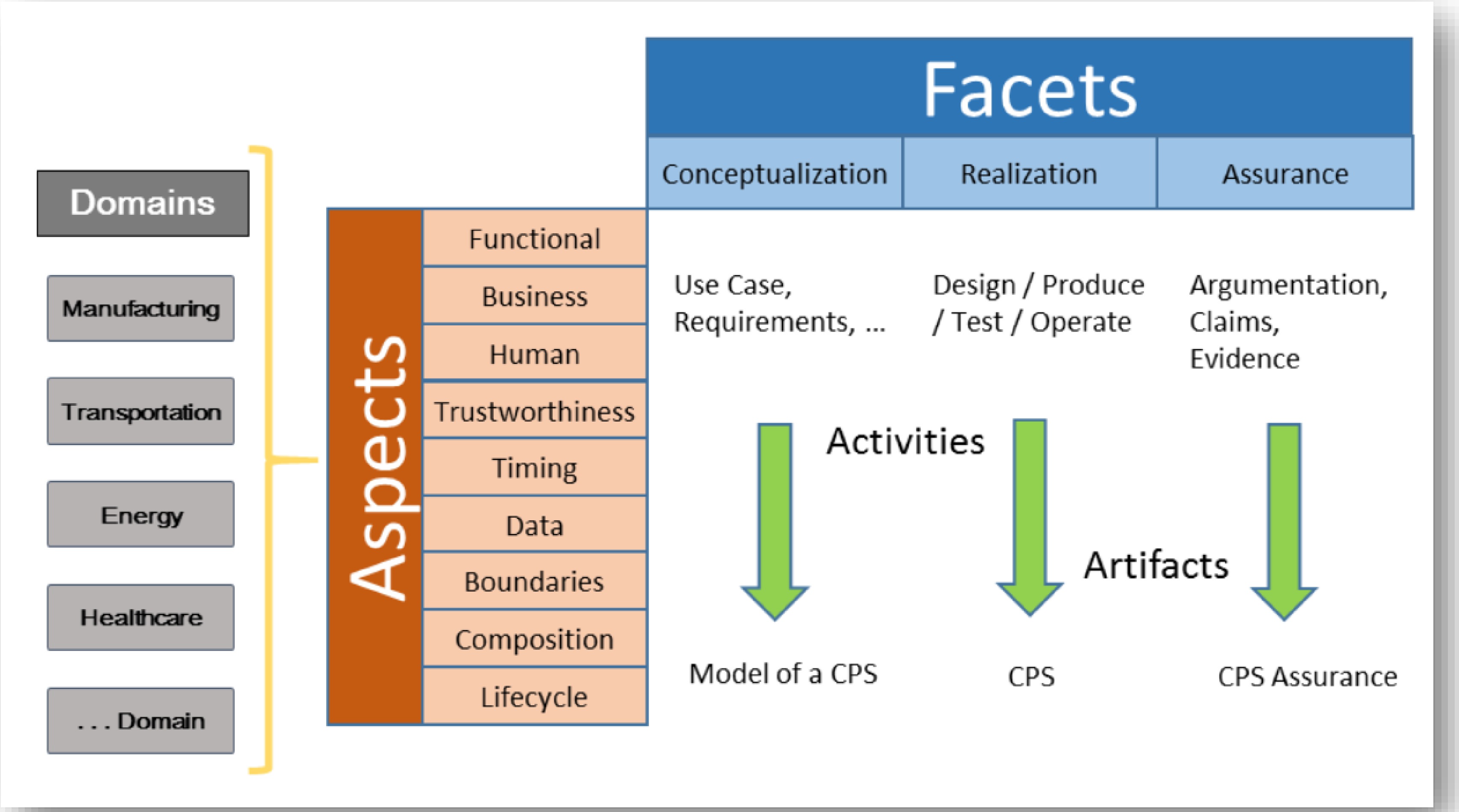
Outline

- Introduction
  - Model-Based Design context for Software and System Development
- Two new recently-released products/features related to my research
  - Architecture Design and Analysis
  - Formalizing Specifications
- Conclusion
  - Link back to the Model-Based Design context

Model-based design 'V'

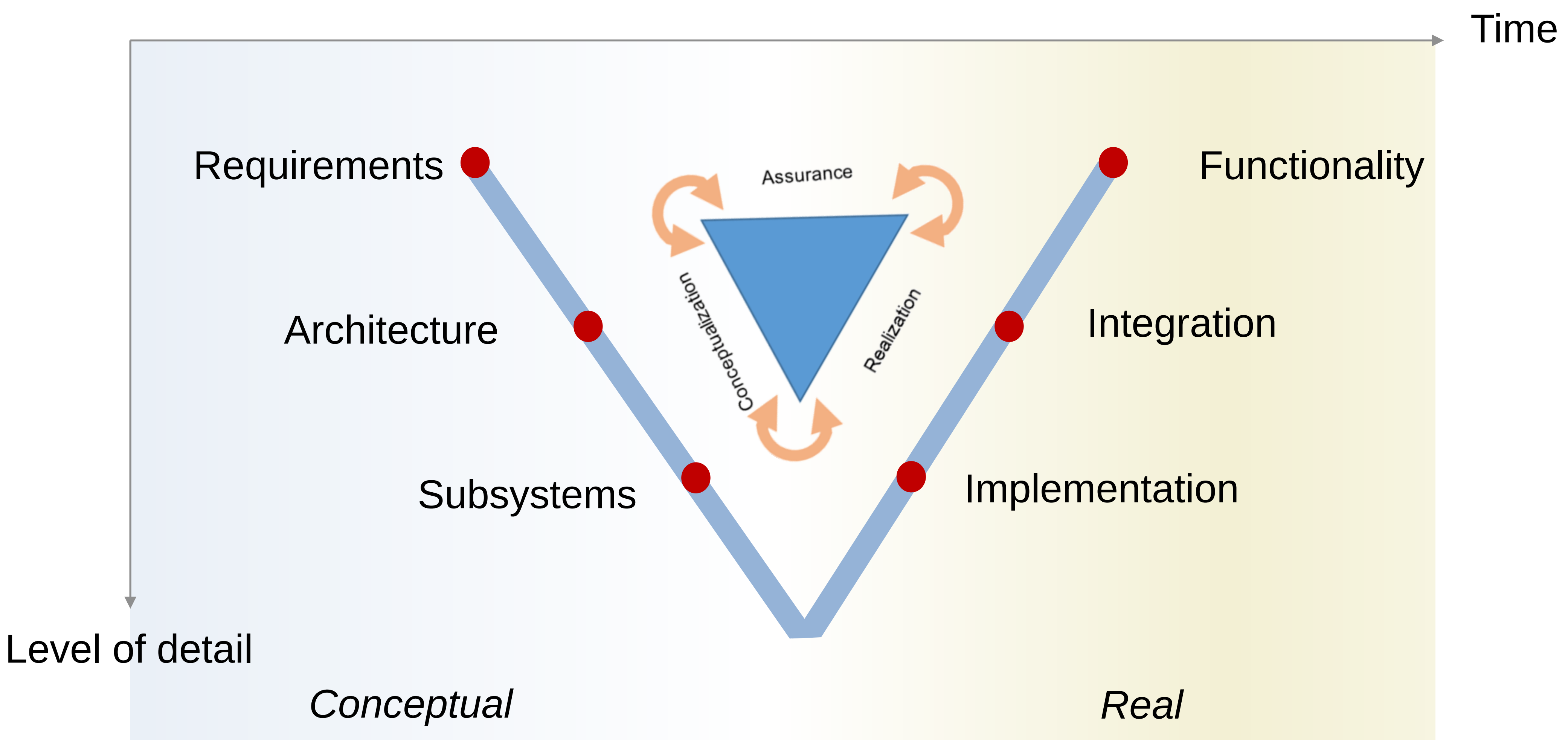


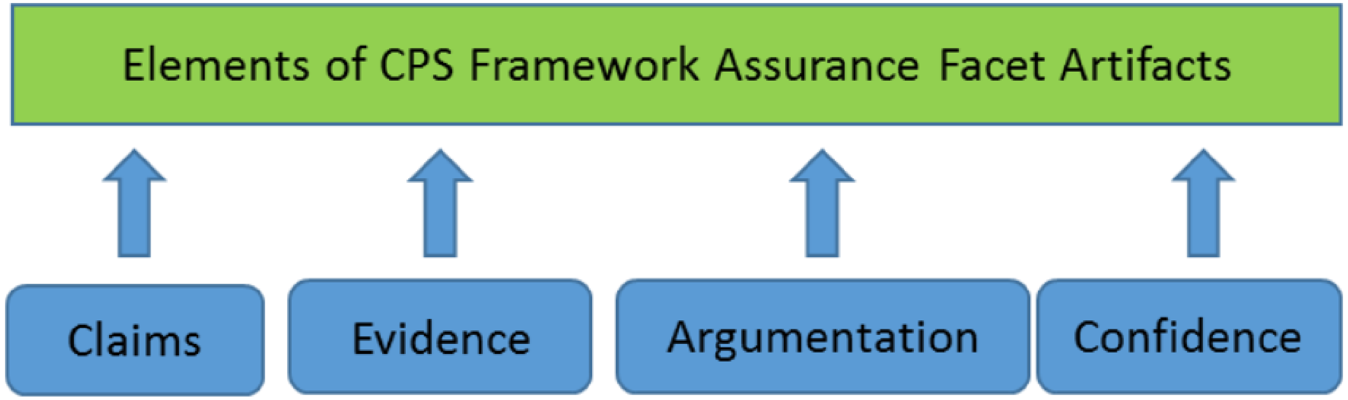
NIST CPS Framework



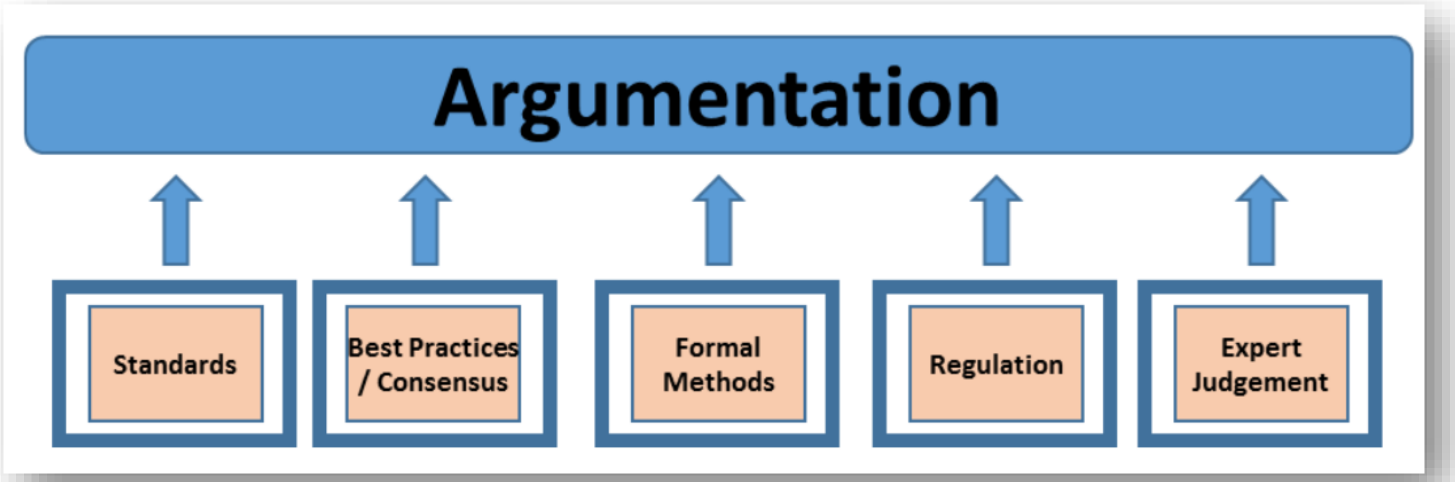
https://www.nist.gov/publications/framework-cyber-physical-systems-volume-1-overview

Model-based design ‘V’

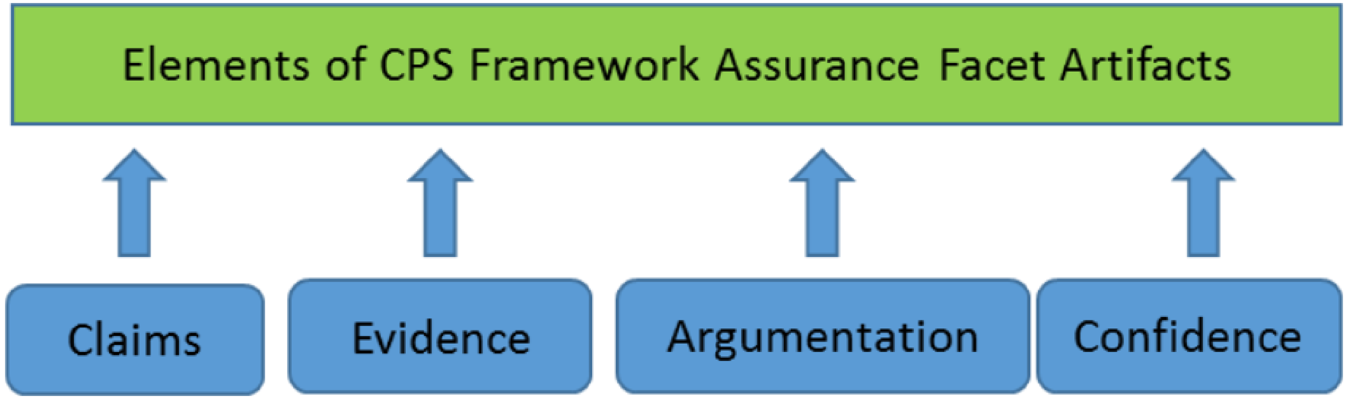




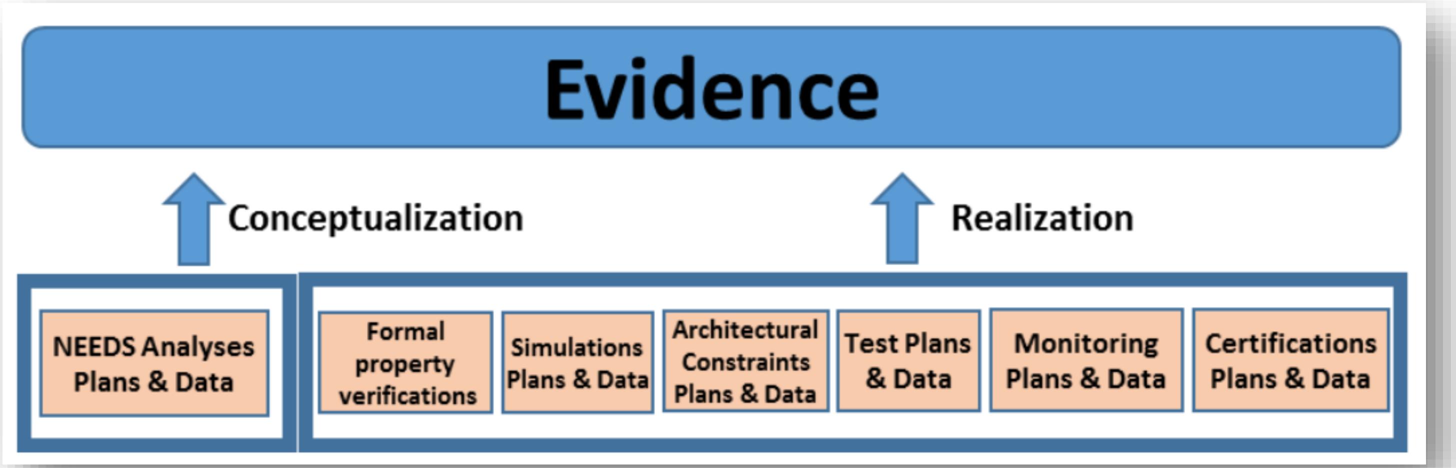
"The [Evidence] is sufficient to conclude that the [Claims] are true based on the [Argumentation] with this [Estimate of Confidence]."



https://www.nist.gov/publications/framework-cyber-physical-systems-volume-1-overview

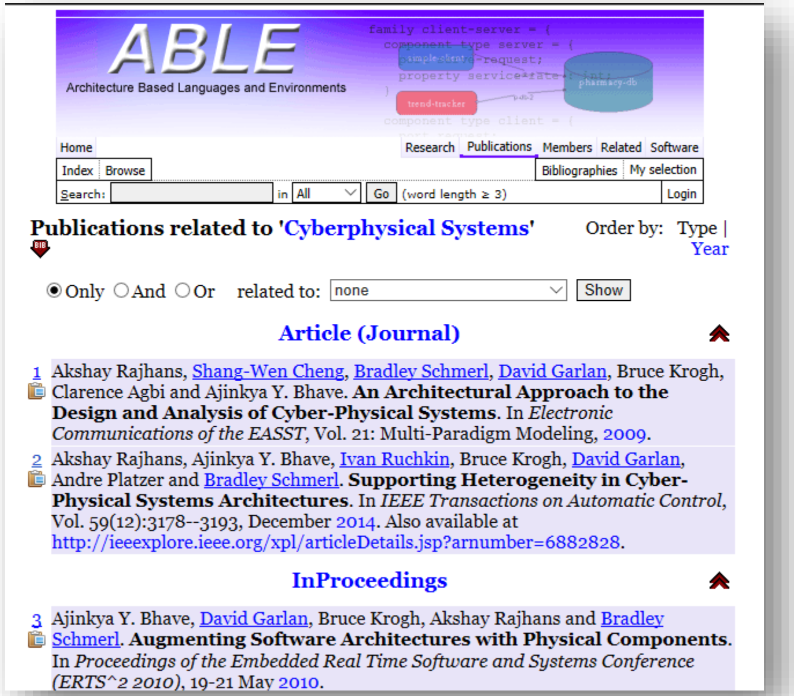


"The [Evidence] is sufficient to conclude that the [Claims] are true based on the [Argumentation] with this [Estimate of Confidence]."



https://www.nist.gov/publications/framework-cyber-physical-systems-volume-1-overview

Two new automated functionalities related to my own work



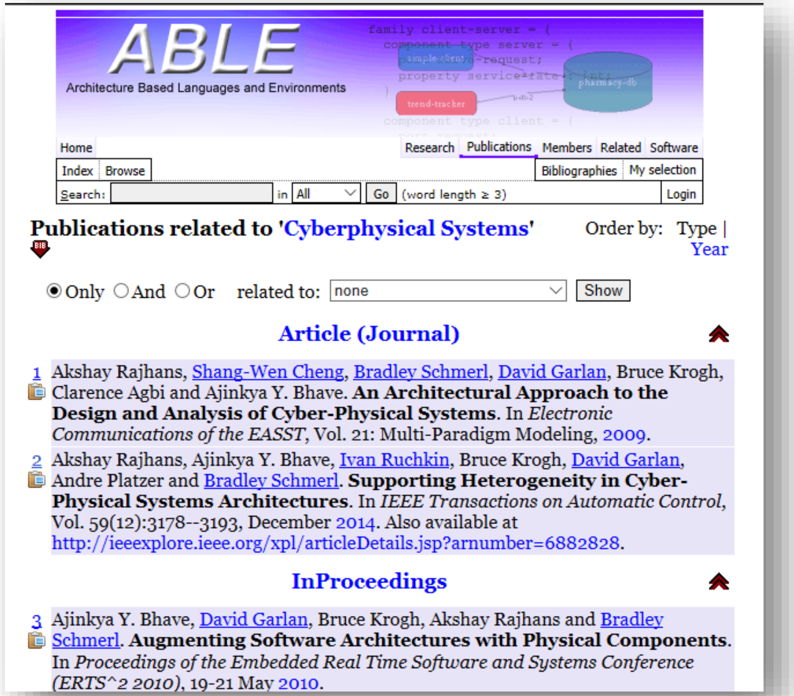
http://acme.able.cs.cmu.edu/pubs/show.php?keyword=Cyberphysical%20Systems

https://sites.google.com/berkeley.edu/mt-cps2019/

Architecture Modeling and Analysis

Formalizing Specifications

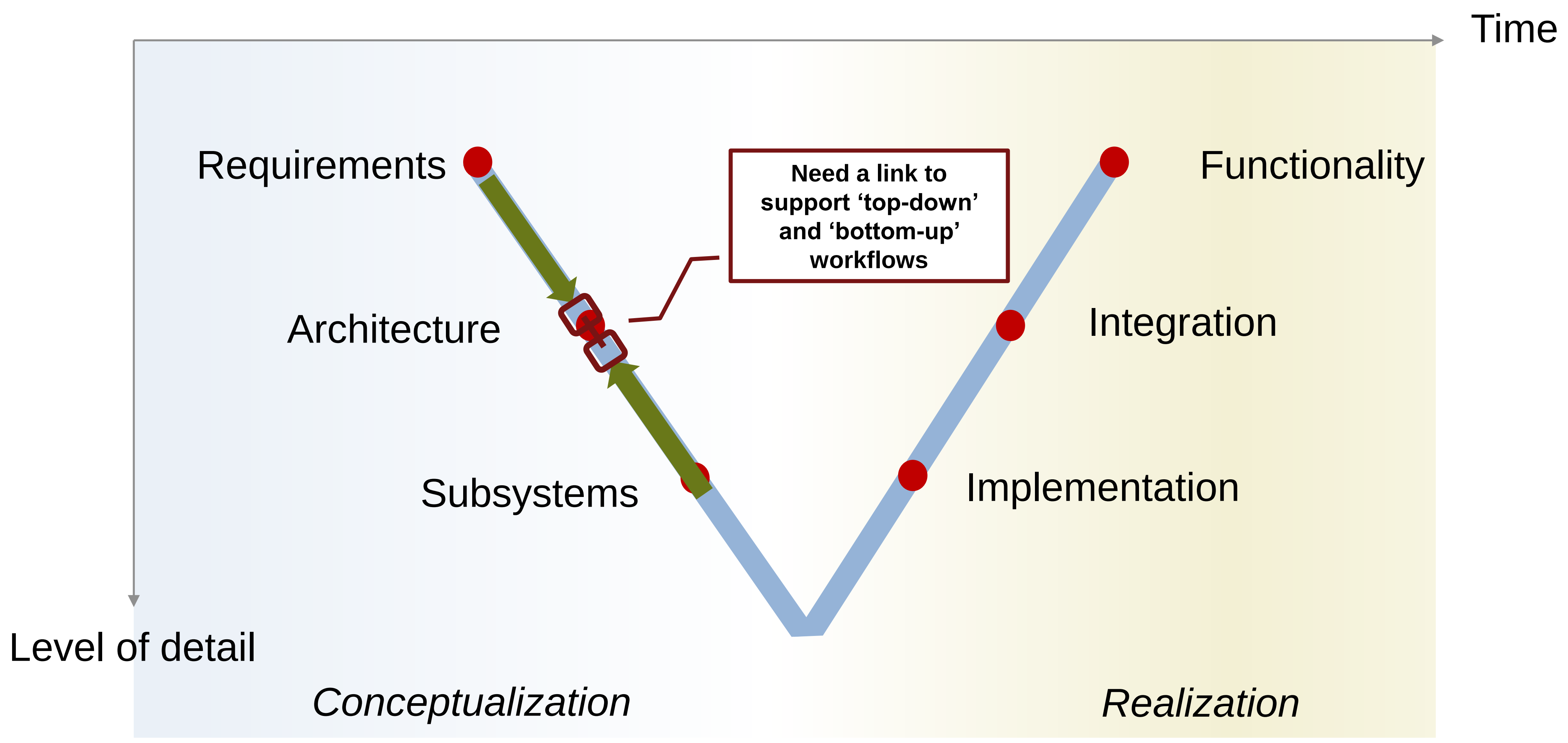
Architecture Modeling and Analysis



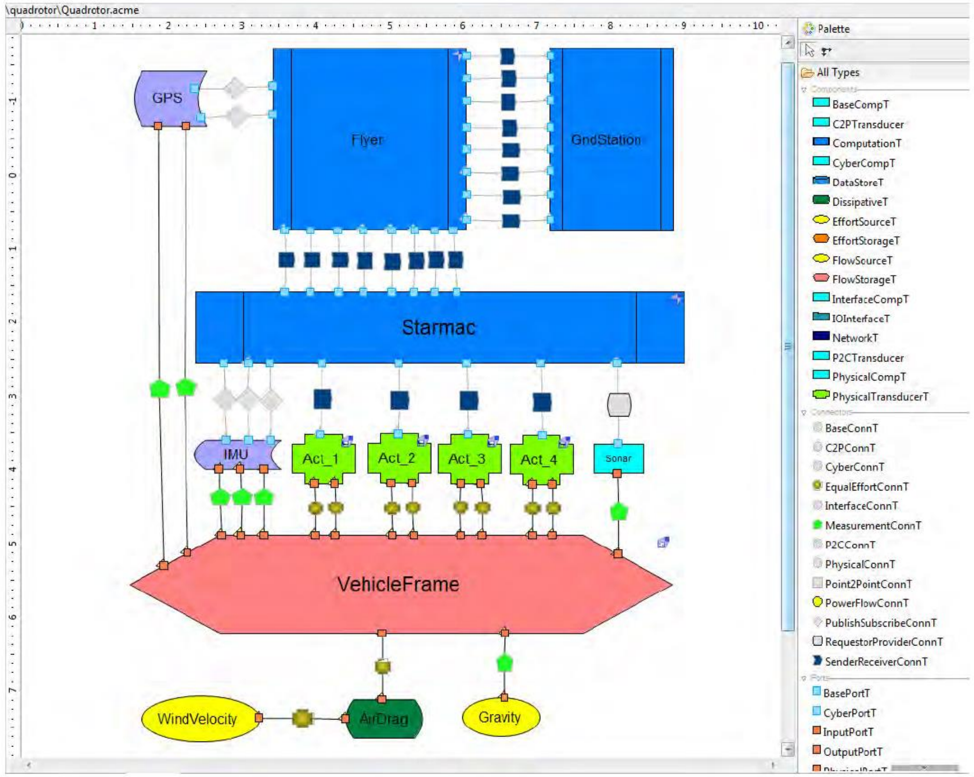
http://acme.able.cs.cmu.edu/pubs/show.php?keyword=Cyberphysical%20Systems

Architecture Modeling and Analysis

Model-based design ‘V’



Architecture modeling of the STARMAC quadrotor



Augmenting Software Architectures with Physical Components

Ajinkya Bhave[1], David Garlan[2], Bruce H. Krogh[1], Akshay Rajhans[1], Bradley Schmerl[2]

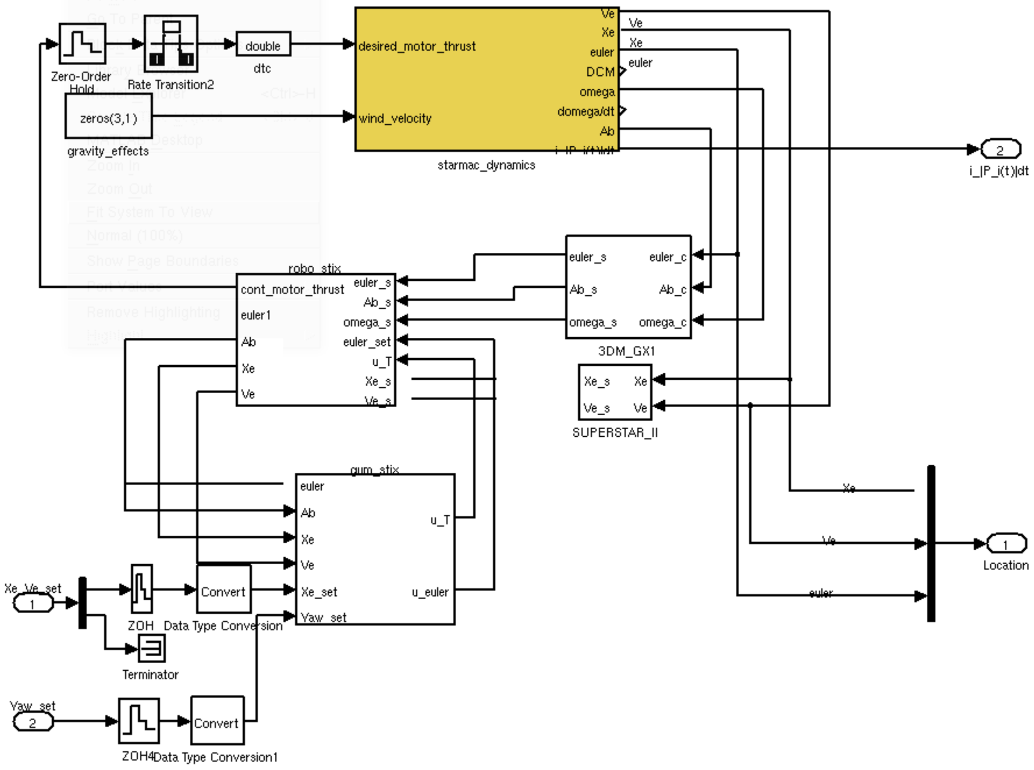
[1]Dept. of Electrical and Computer Engineering
[2]School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3890 USA
email: {ajinkya@ | garlan@cs.| krogh@ece.| arajhans@ece.| schmerl@cs.}cmu.edu

ERTS² '10

http://www.cs.cmu.edu/~acme/AcmeStudio/

Work done in the context of HCDDES MURI: https://ptolemy.berkeley.edu/projects/chess/hcddes/

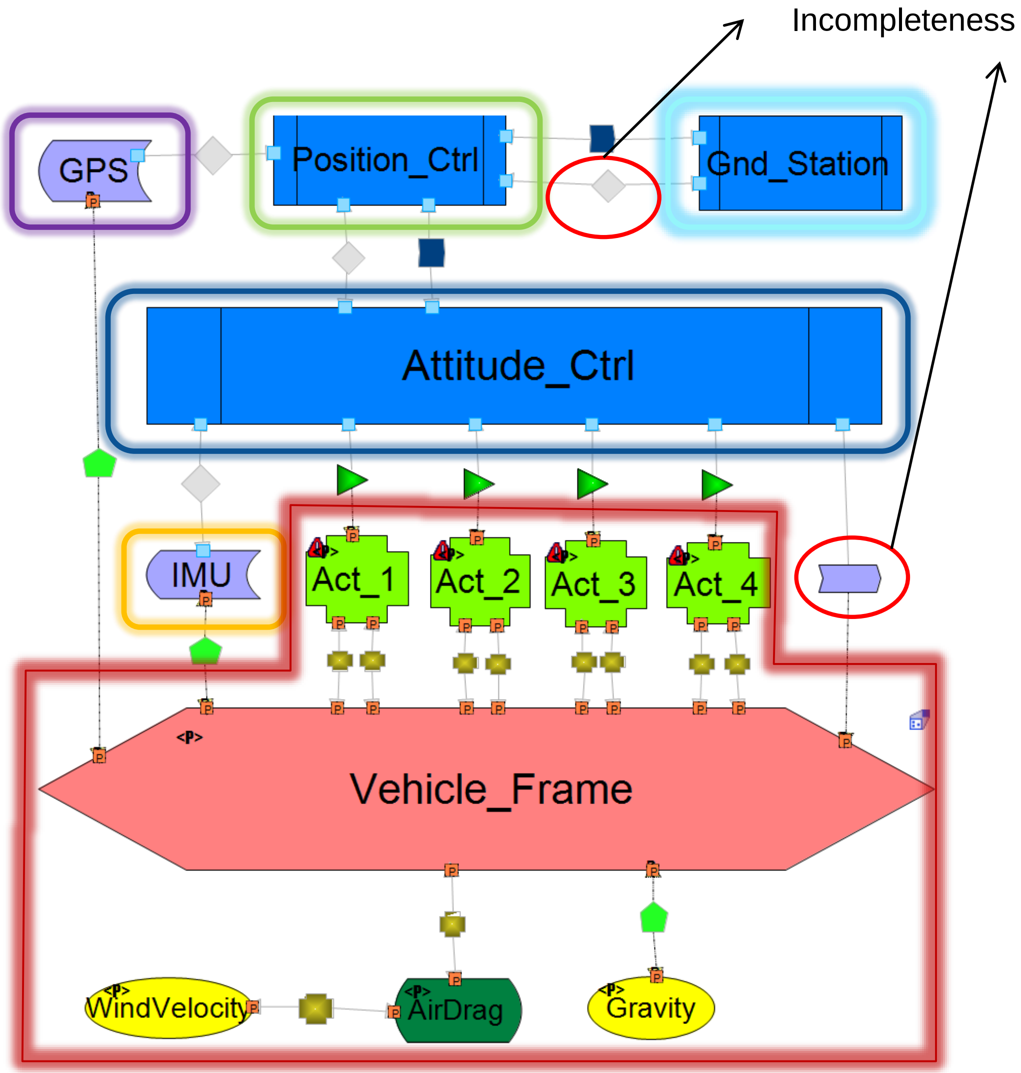
Simulink Architecture View



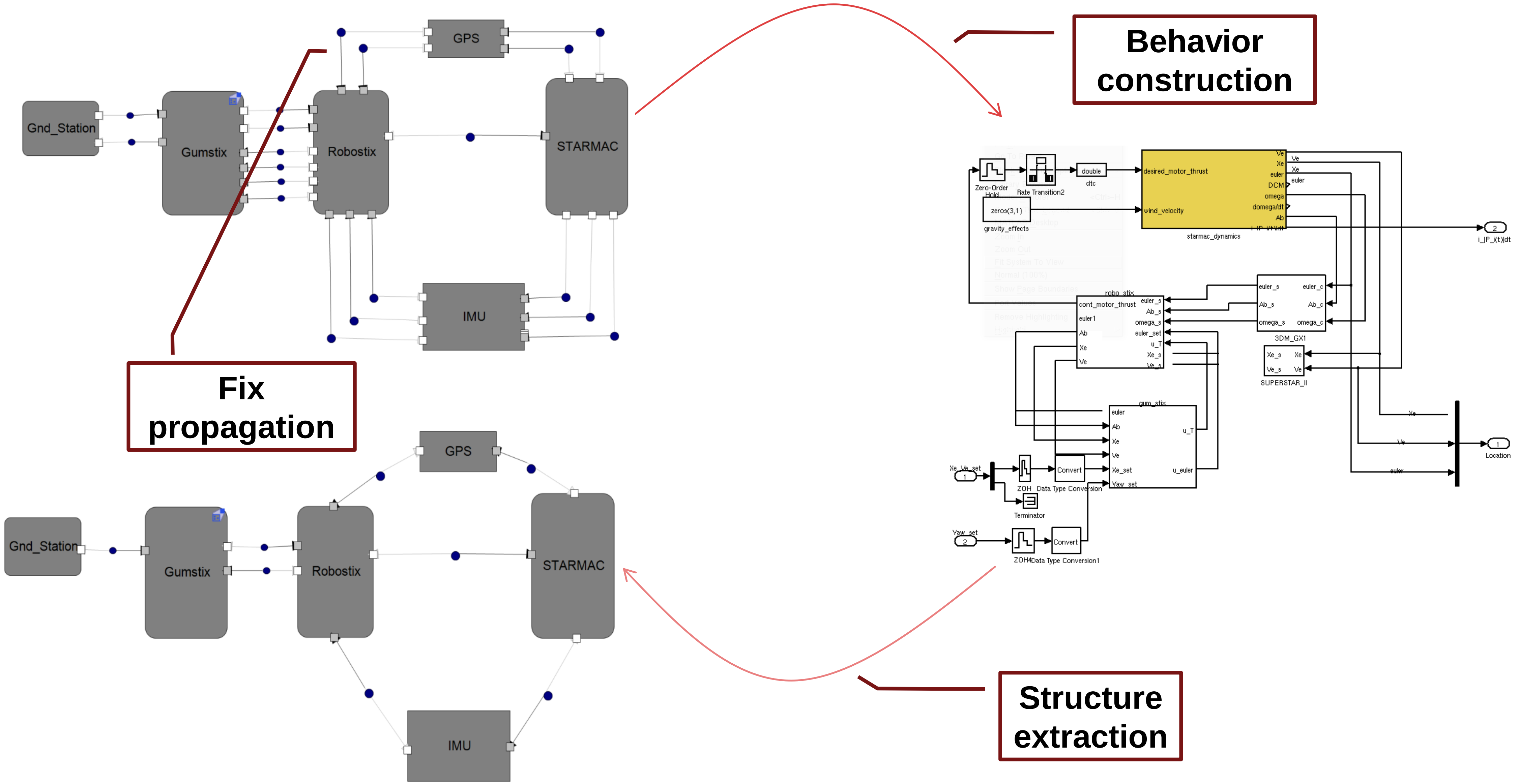
View Consistency in Architectures for Cyber-Physical Systems

ICCPS '11

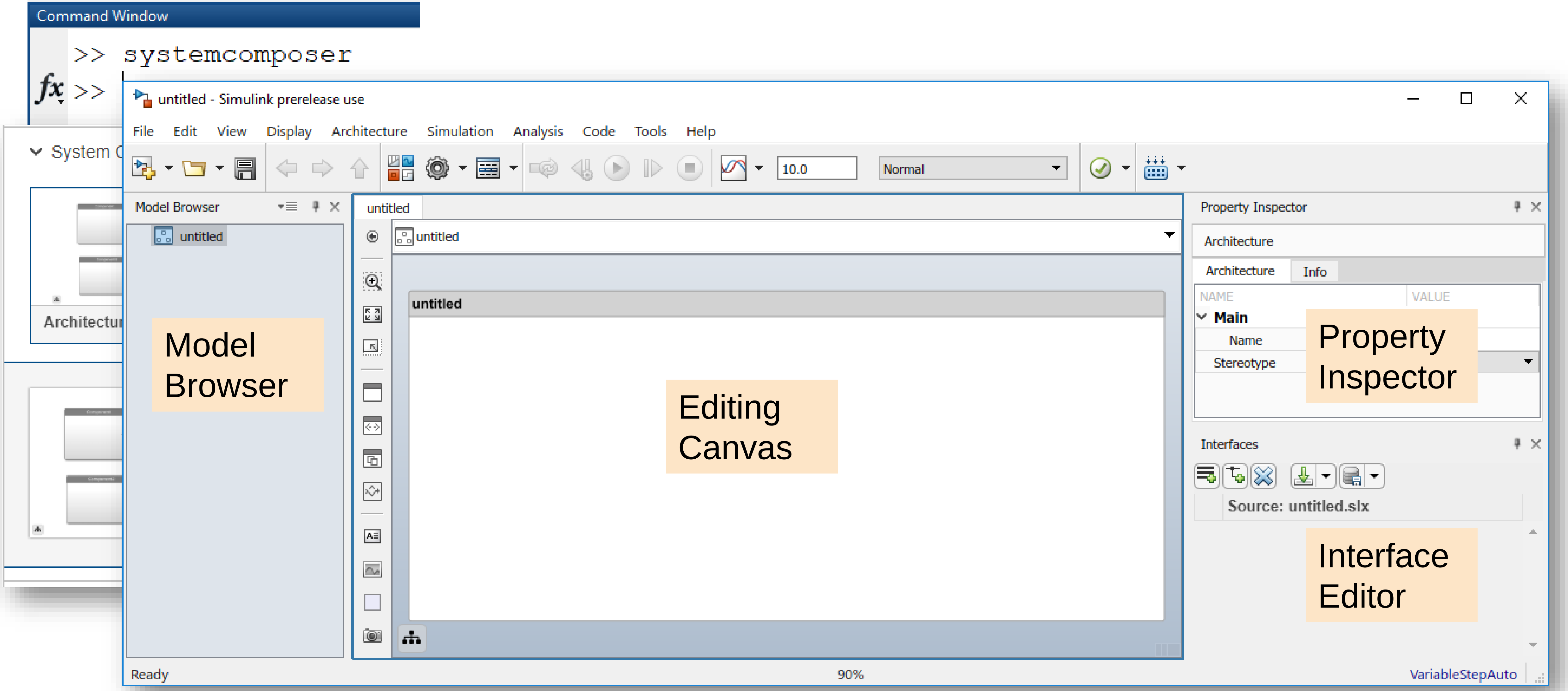
Ajinkya Bhave, Bruce H. Krogh David Garlan, Bradley Schmerl



Simulink Architecture ←→ Simulink Model: Manual Step

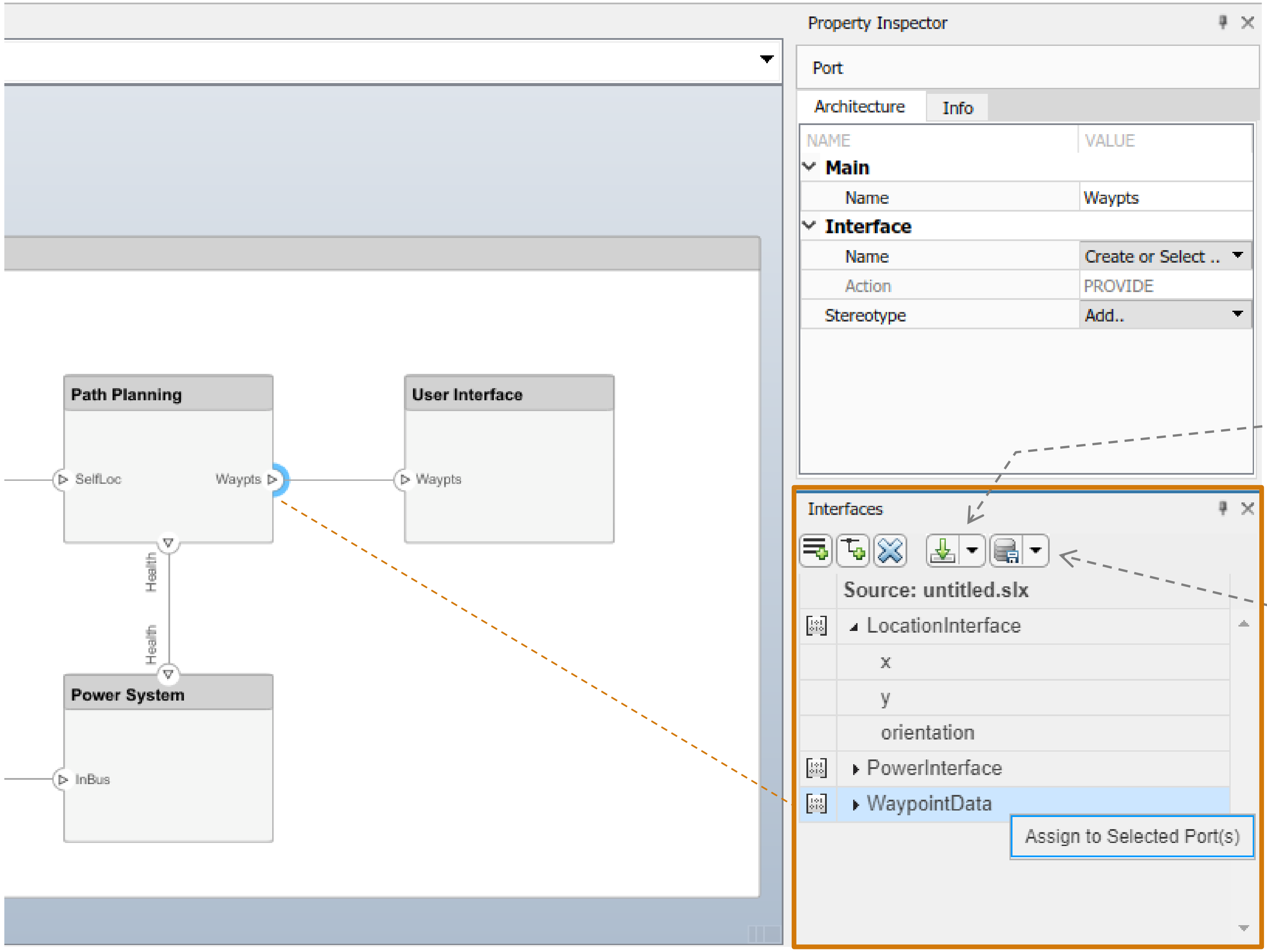


System Composer



https://www.mathworks.com/products/system-composer.html

Describe abstract component interfaces and allocate to ports



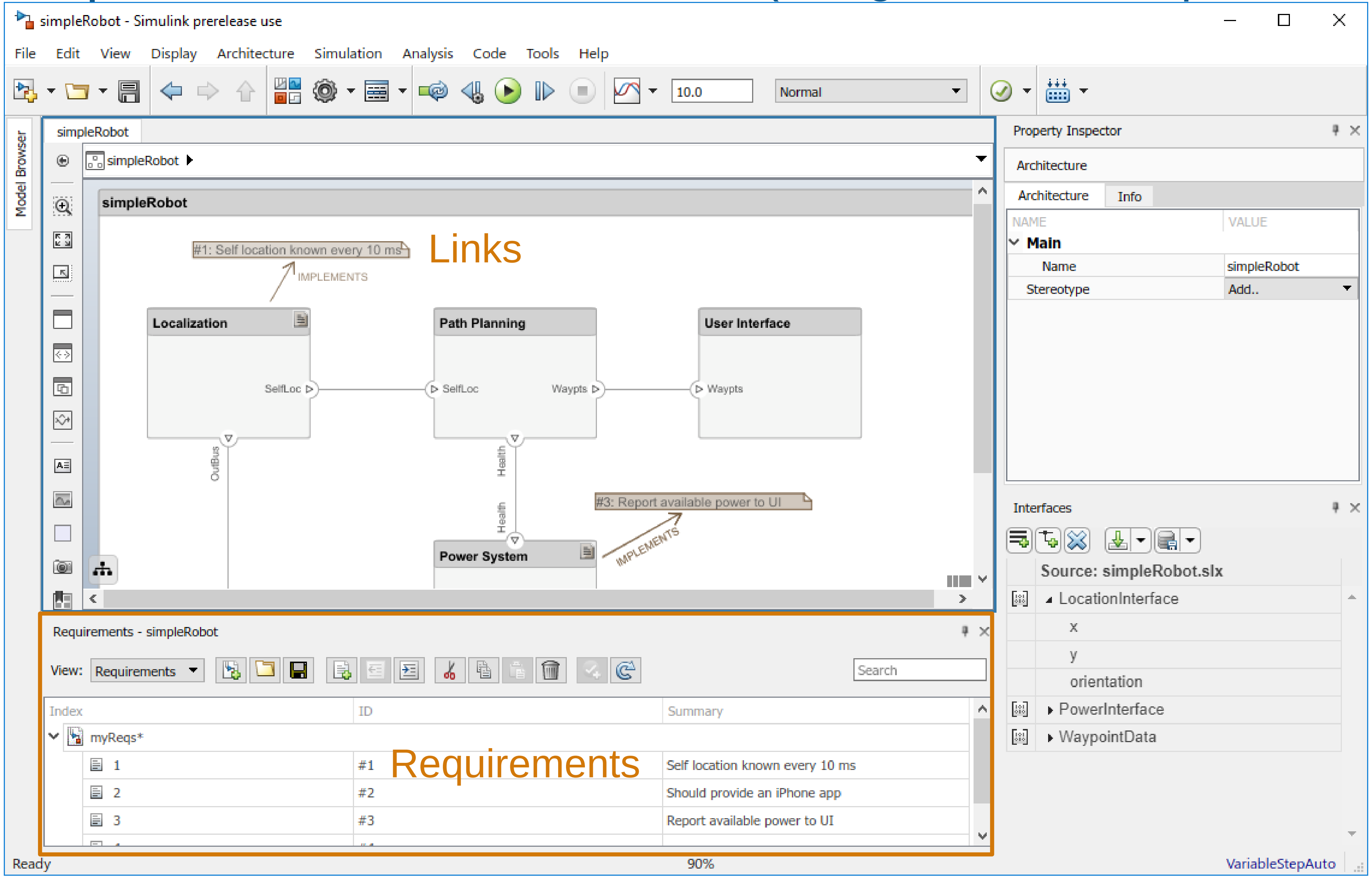
Store in the model when sketching

Import from workspace or file

Save/link to data dictionary

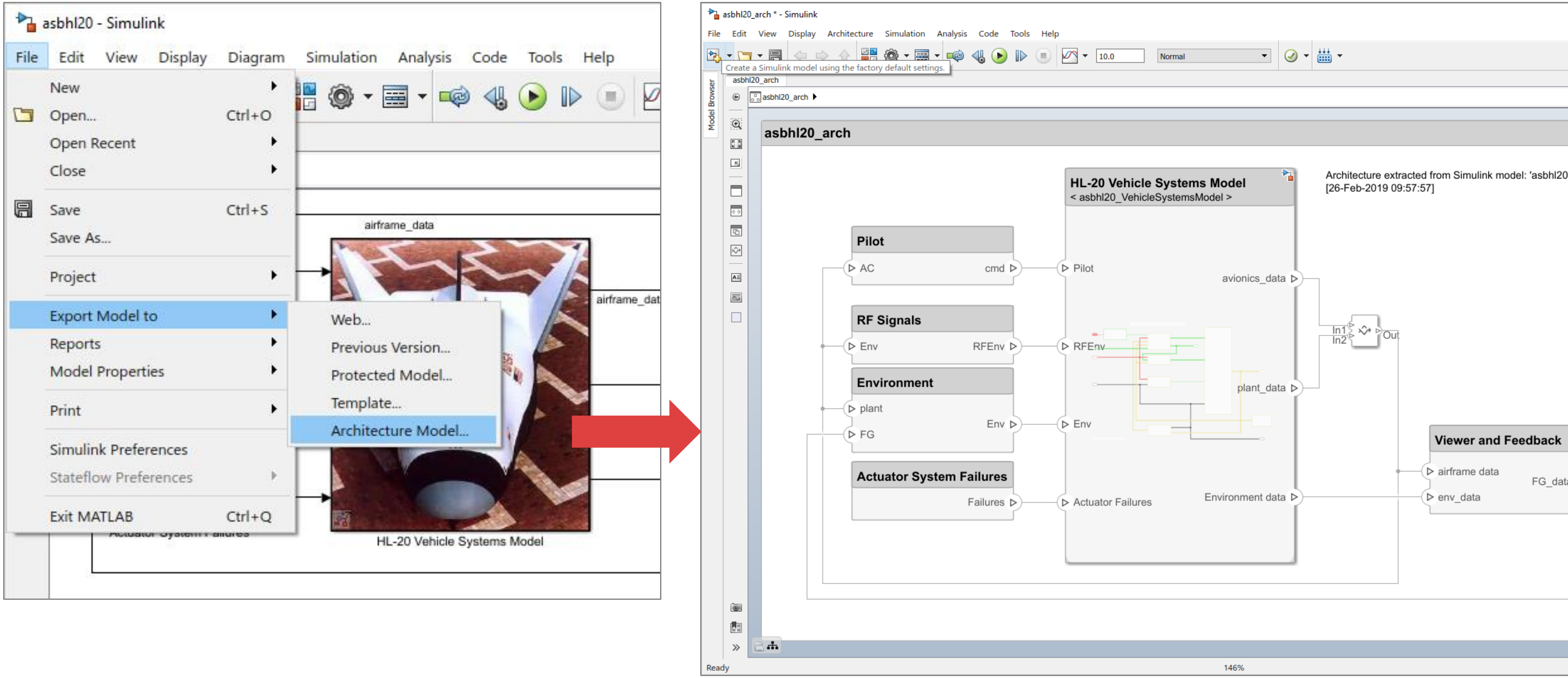
https://www.mathworks.com/products/system-composer.html

Allocate requirements to architectures (using Simulink Requirements)



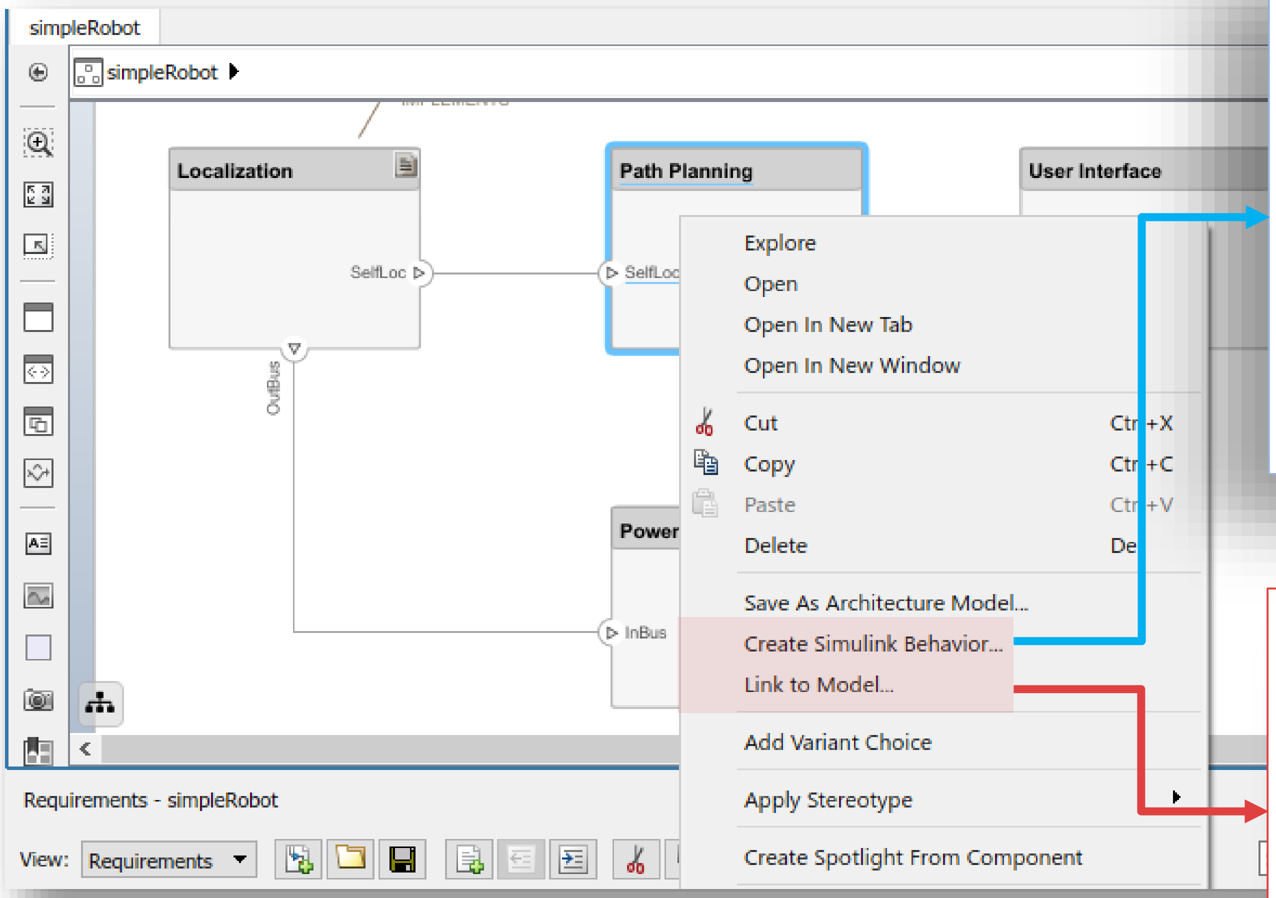
https://www.mathworks.com/videos/requirements-allocation-to-architectural-elements-1550678923400.html

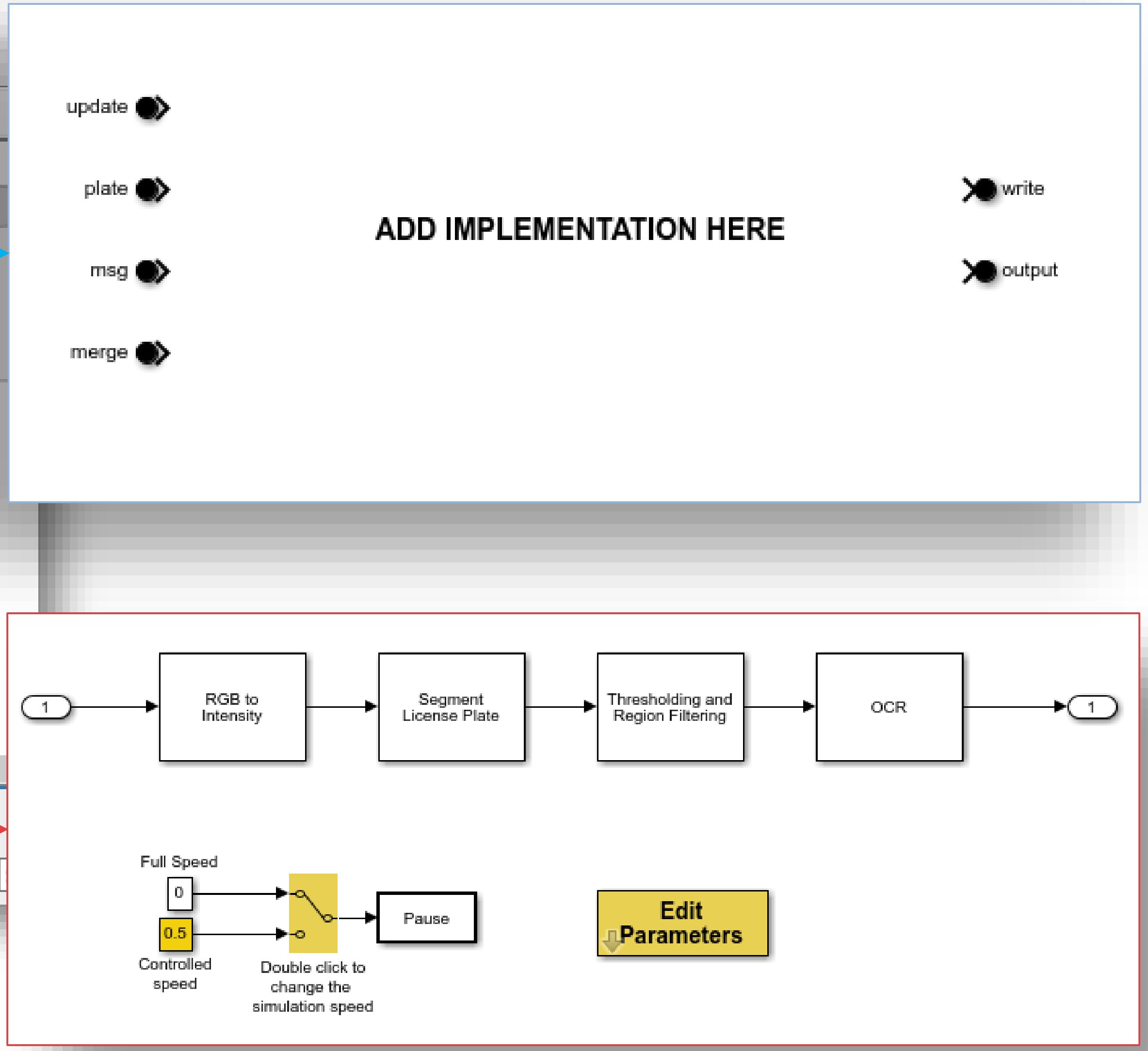
Simulink to Architecture Automation



https://www.mathworks.com/help/systemcomposer/ug/extract-architecture-from-simulink-model.html

Architecture to Simulink Automation

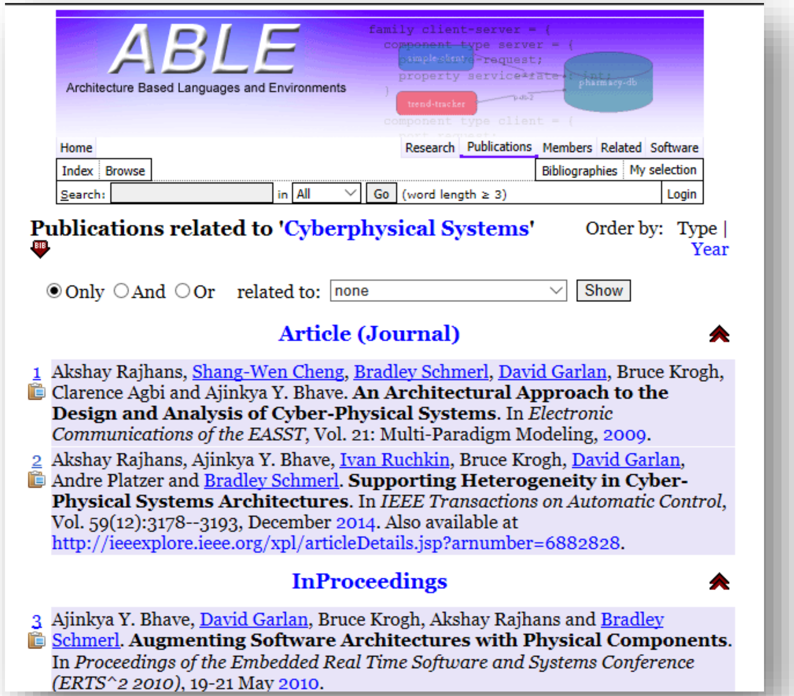




Interfaces shared across models via data dictionaries

https://www.mathworks.com/help/systemcomposer/ug/implement-components-in-simulink.html

Two new automated functionalities related to my own work



http://acme.able.cs.cmu.edu/pubs/show.php?keyword=Cyberphysical%20Systems

https://sites.google.com/berkeley.edu/mt-cps2019/

Architecture Modeling and Analysis

Formalizing Specifications

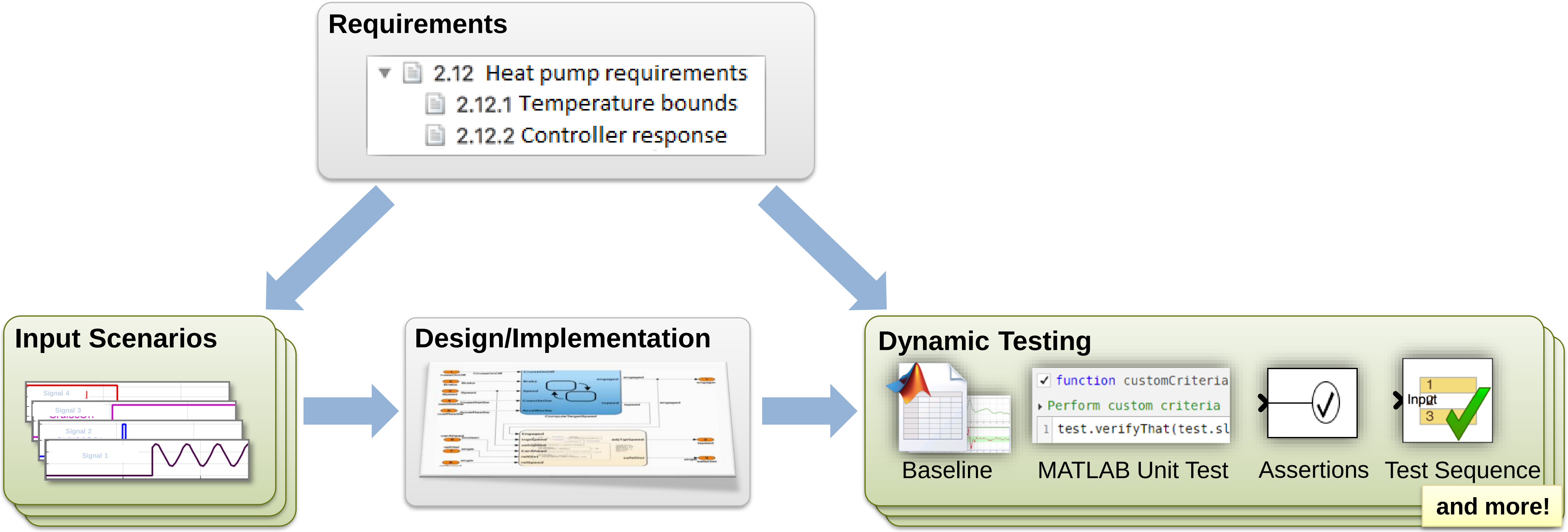
Formalizing Specifications

https://sites.google.com/berkeley.edu/mt-cps2019/

Formalizing Specifications

J.-F. Kempf, Khoo Y. P., and A. Rajhans, "*Specification and Assessment of Temporal Requirements using Simulink Test*", Fourth International Workshop on Monitoring and Testing of Cyber-Physical Systems (MT-CPS 2019), part of CPS-IoT Week 2019. [Abstract (PDF)]

Testing today



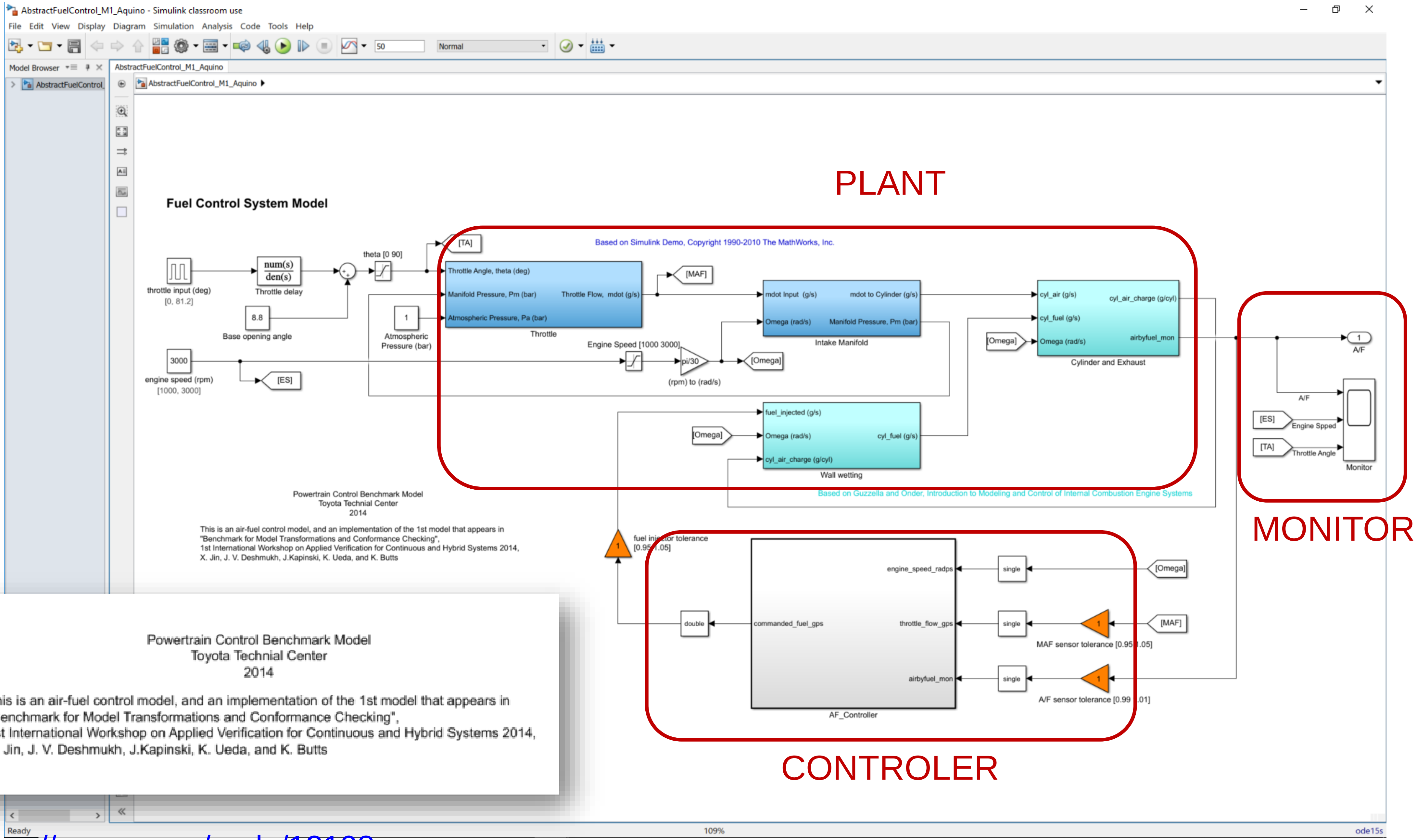
Toyota Air-Fuel Ratio Control Example

We define the normalized A/F ratio μ as $\frac{(\lambda - 14.7)}{14.7}$, where λ is the A/F ratio. As regulating μ to 1 is the control objective, we compare the models on the basis of this signal. We use control-theoretic properties of μ, such as the maximum overshoot, minimum undershoot, and settling time as criteria for comparison. We define a settling region of $\pm 1\%$ of the reference value for μ (which is 1.0) for the cases where the engine speed is $[1000, 1500]$ rpm. For higher speeds we use a settling region of $\pm 2\%$. We also measure the RMS error between the signal μ_c for the complex model, and the signal μ_s for the simplified model.

Powertrain Control Benchmark Model
Toyota Technial Center
2014

This is an air-fuel control model, and an implementation of the 1st model that appears in "Benchmark for Model Transformations and Conformance Checking", 1st International Workshop on Applied Verification for Continuous and Hybrid Systems 2014, X. Jin, J. V. Deshmukh, J.Kapinski, K. Ueda, and K. Butts

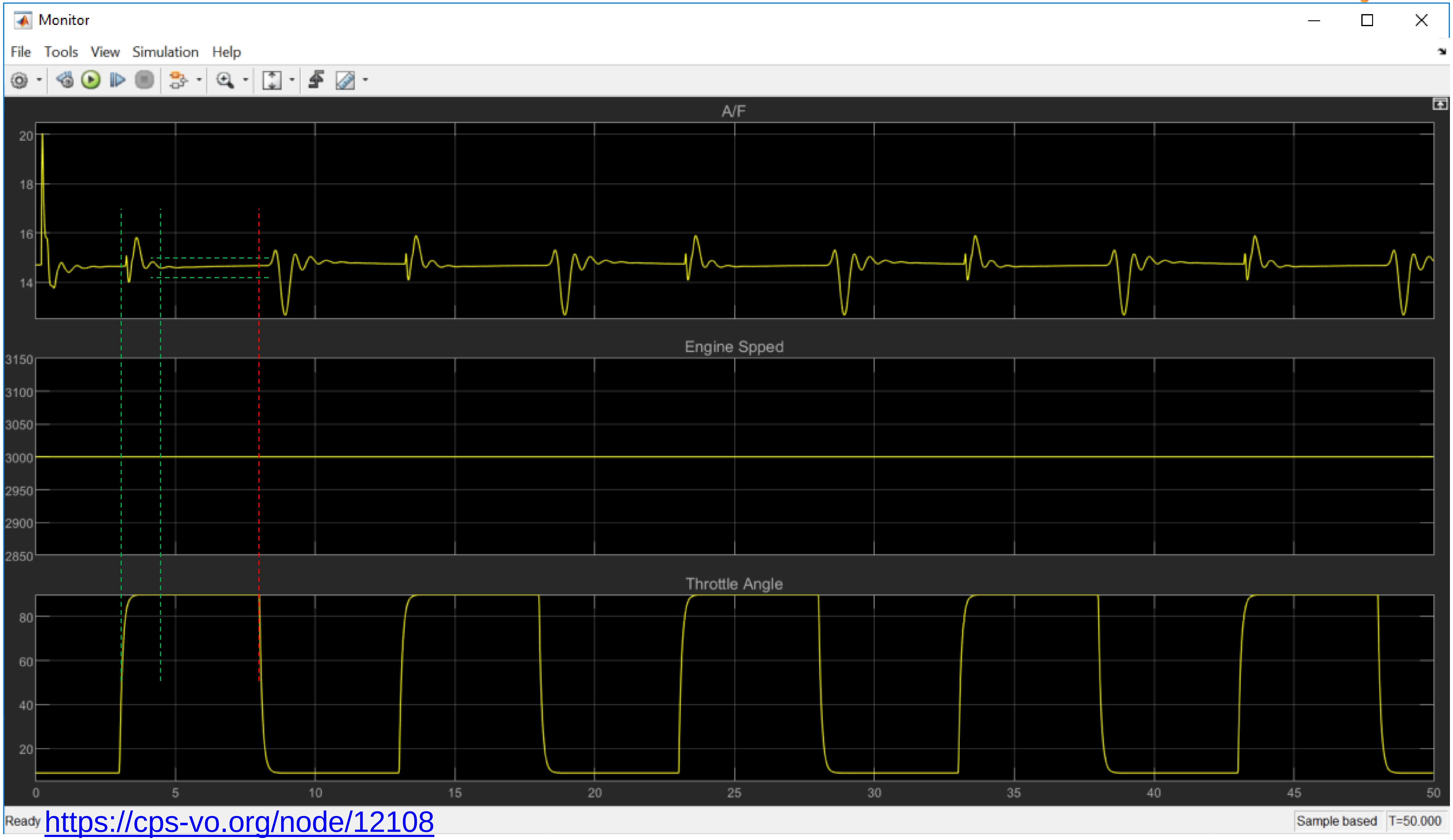
https://cps-vo.org/node/12108



Powertrain Control Benchmark Model
Toyota Technial Center
2014

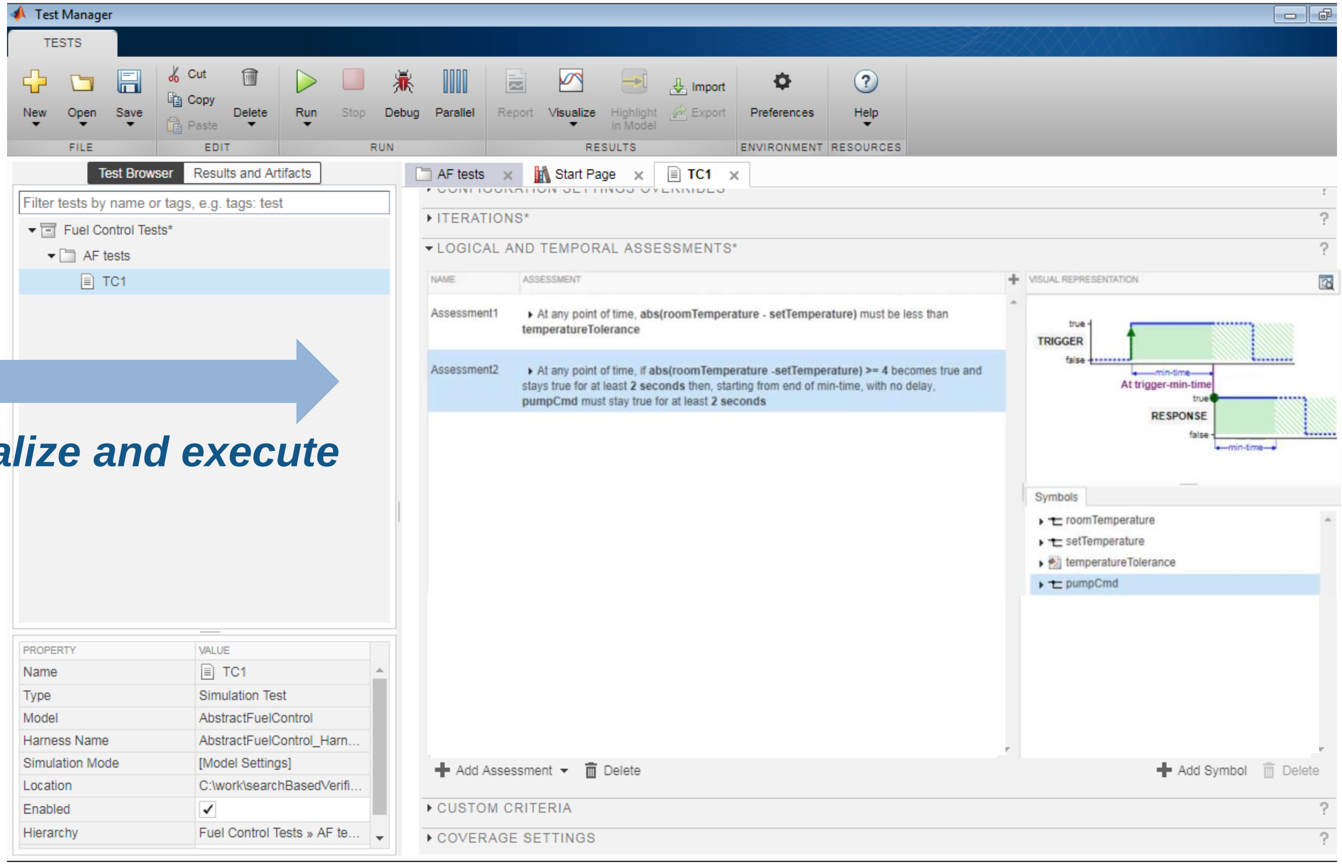
This is an air-fuel control model, and an implementation of the 1st model that appears in "Benchmark for Model Transformations and Conformance Checking", 1st International Workshop on Applied Verification for Continuous and Hybrid Systems 2014, X. Jin, J. V. Deshmukh, J.Kapinski, K. Ueda, and K. Butts

https://cps-vo.org/node/12108



https://cps-vo.org/node/12108

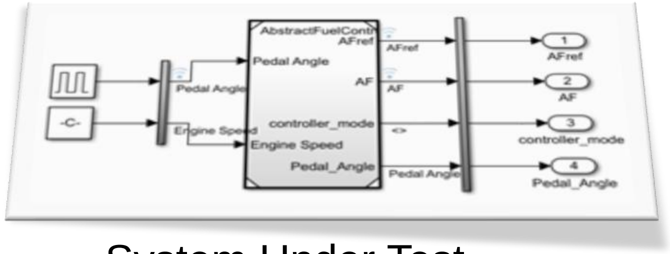
How could we formalize and execute requirements directly?



Requirements

1. The difference between the room temperature and the set temperature should never exceed 6 degrees.
2. If the temperature difference exceeds 4 degrees for more than 2 seconds, then the pump shall activate for at least 2 seconds

Formalize and execute



System Under Test

J.-F. Kempf, Khoo Y. P., and A. Rajhans, "*Specification and Assessment of Temporal Requirements using Simulink Test*", Fourth International Workshop on Monitoring and Testing of Cyber-Physical Systems (MT-CPS 2019), part of CPS-IoT Week 2019. [Abstract (PDF)]

Formal property specification for Simulink models

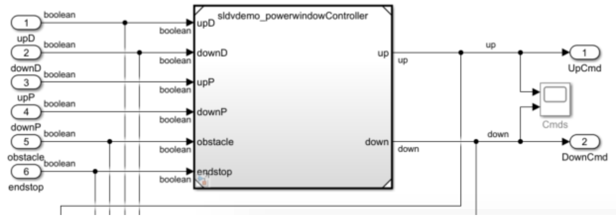
1. Using MATLAB scripts
   – Breach
   – Toyota ARCH'14 Benchmark

```
%% Writing a Simple STL Specification
% First we define a predicate stating that AF is above 1% of AFref
AF_not_ok = STL_Formula('AF_ok', 'abs(AF[t]- AFref[t]) < 0.01*14.7')
```

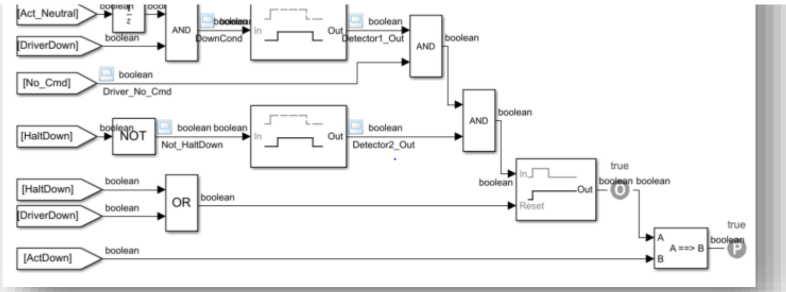
```
if (abs(y(i)-ref) < 0.02*ref) && (abs(y(i-1)-ref) > 0.02*ref)
    stime_iter = t(i)-start;
    stime = max(stime,stime_iter);
end
```

2. Using Simulink blocks
   – Simulink Design Verifier
   – Jens Oehlerking (Bosch internal tool)

Power Window Controller Temporal Property Specification

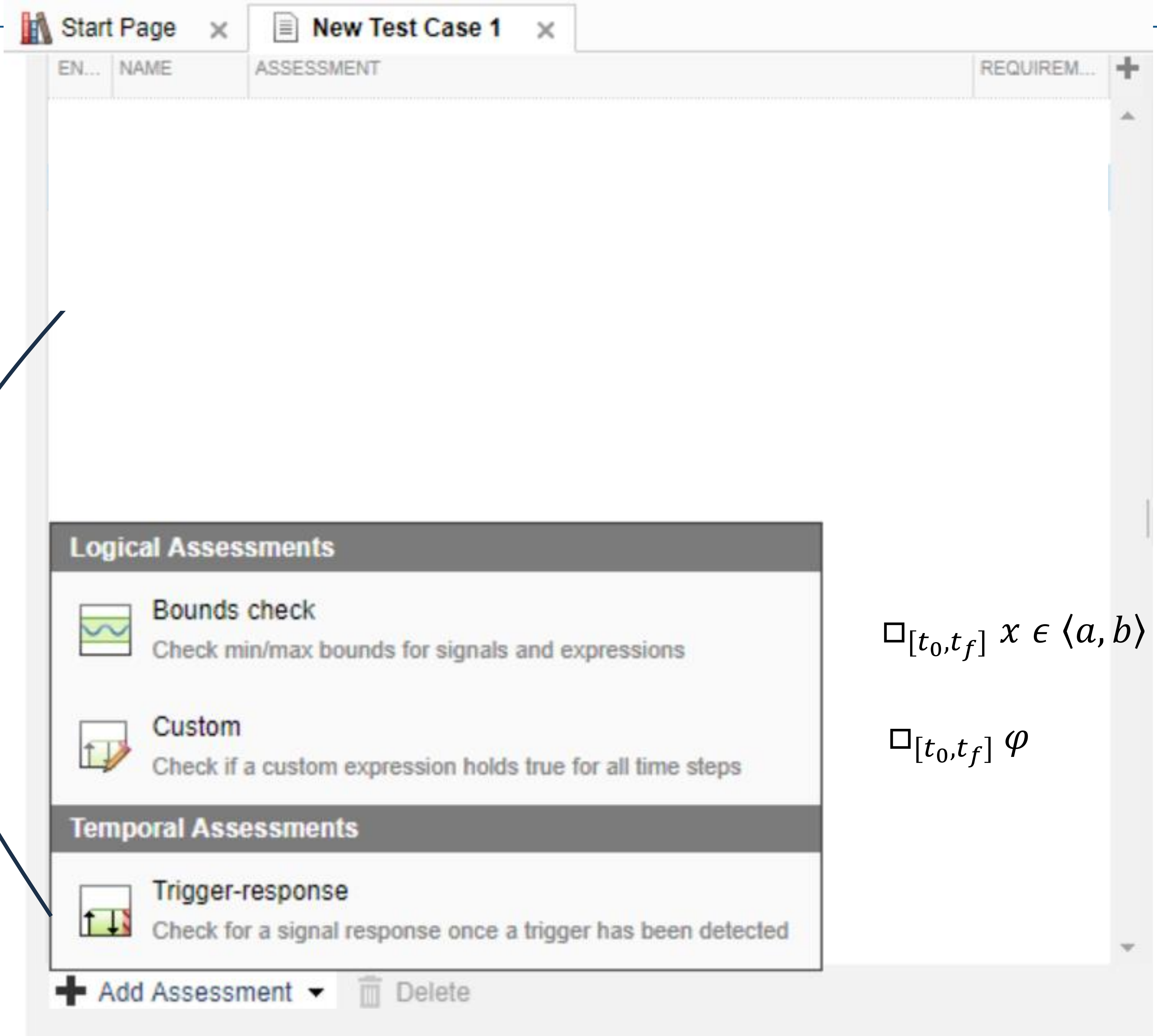


Requirement (Autodown)
If the driver presses the down button for less than 5 steps, then the controller gives the down command as long as end has not been reached or the driver presses the up button.



3. Using a dedicated Test Manager
   – Simulink Test (Logical and Temporal Assessments)

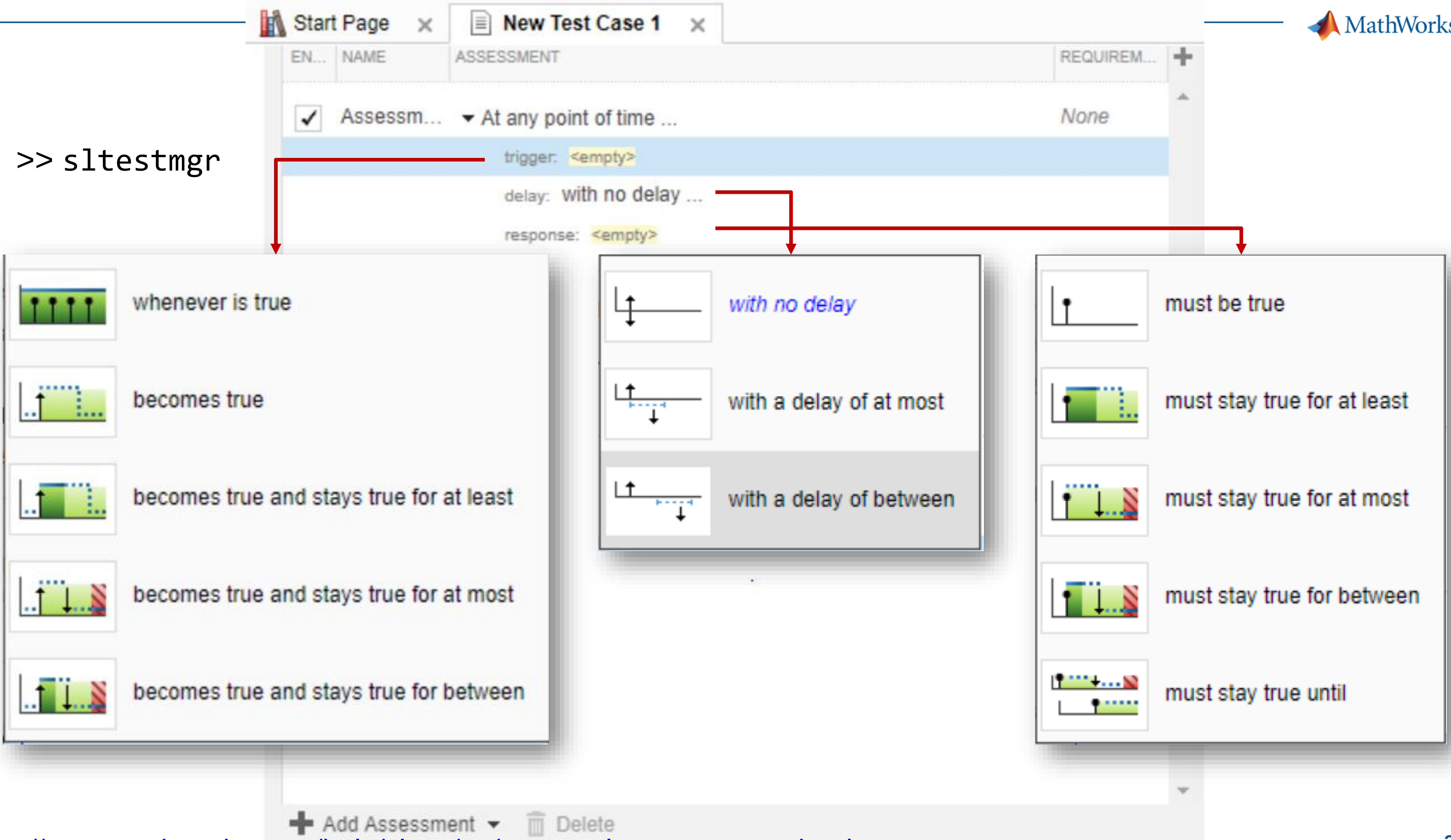
>> sltestmgr



$\square_{[t_0,t_f]}\, x \in \langle a, b \rangle$ where $\langle \;\; \in \{\leq, <\}$

$\square_{[t_0,t_f]}\, \varphi$

```
>> sltestmgr
```

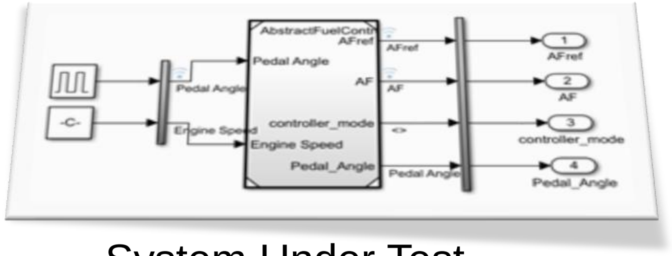


https://www.mathworks.com/help/sltest/ug/temporal-assessments.html

How could we formalize and execute requirements directly?

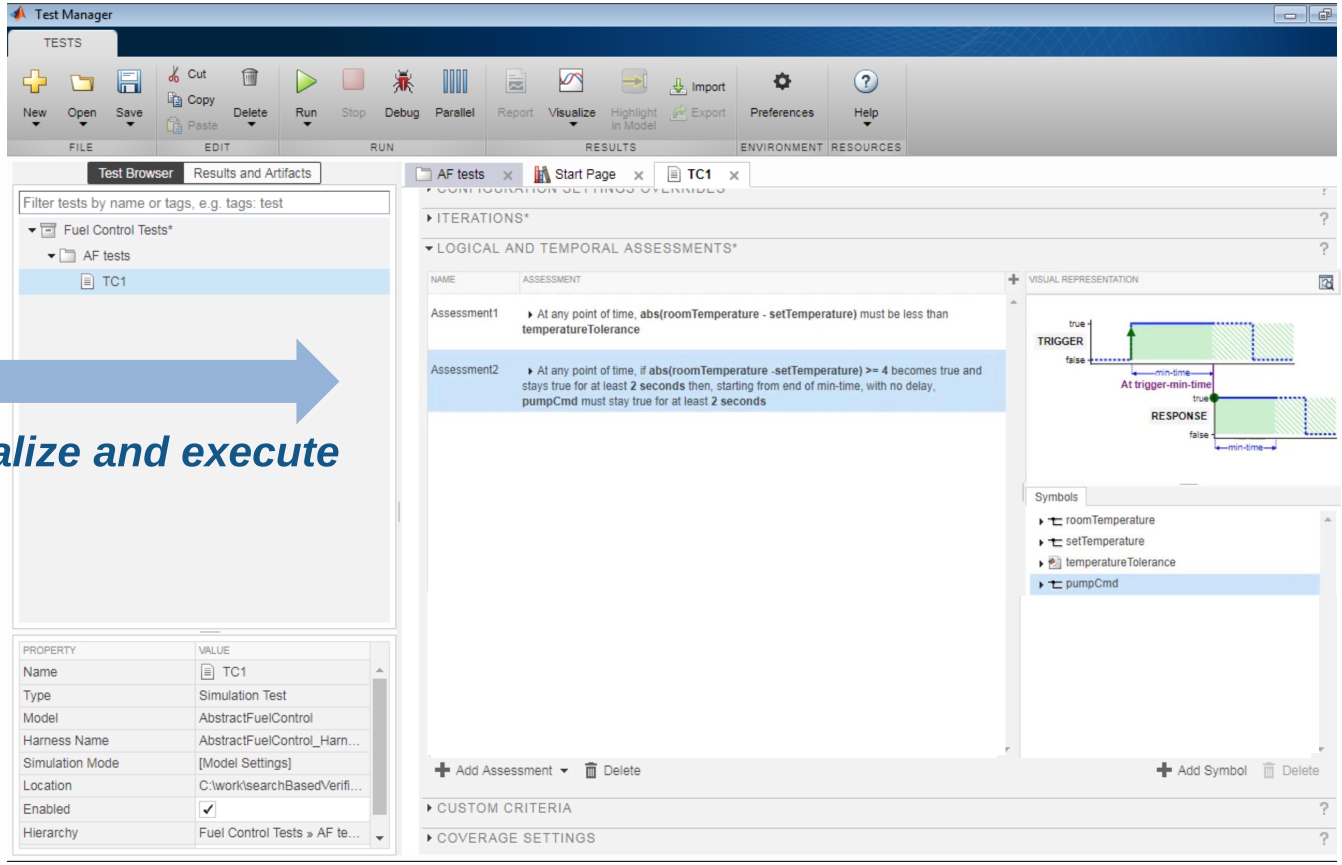
Requirements

1. The difference between the room temperature and the set temperature should never exceed 6 degrees.
2. If the temperature difference exceeds 4 degrees for more than 2 seconds, then the pump shall activate for at least 2 seconds



System Under Test

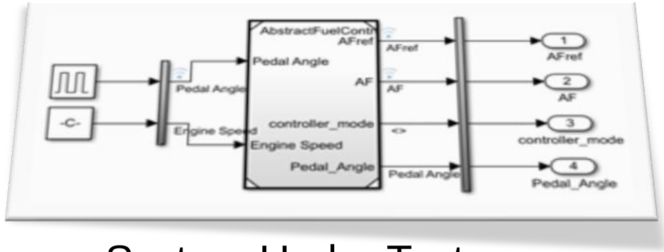
Formalize and execute



How could we formalize and execute requirements directly?

Requirements

1. The difference between the room temperature and the set temperature should never exceed 6 degrees.
2. If the temperature difference exceeds 4 degrees for more than 2 seconds, then the pump shall activate for at least 2 seconds



System Under Test

When <condition 1> **is true,**
Then <condition 2> **must be true for some time**

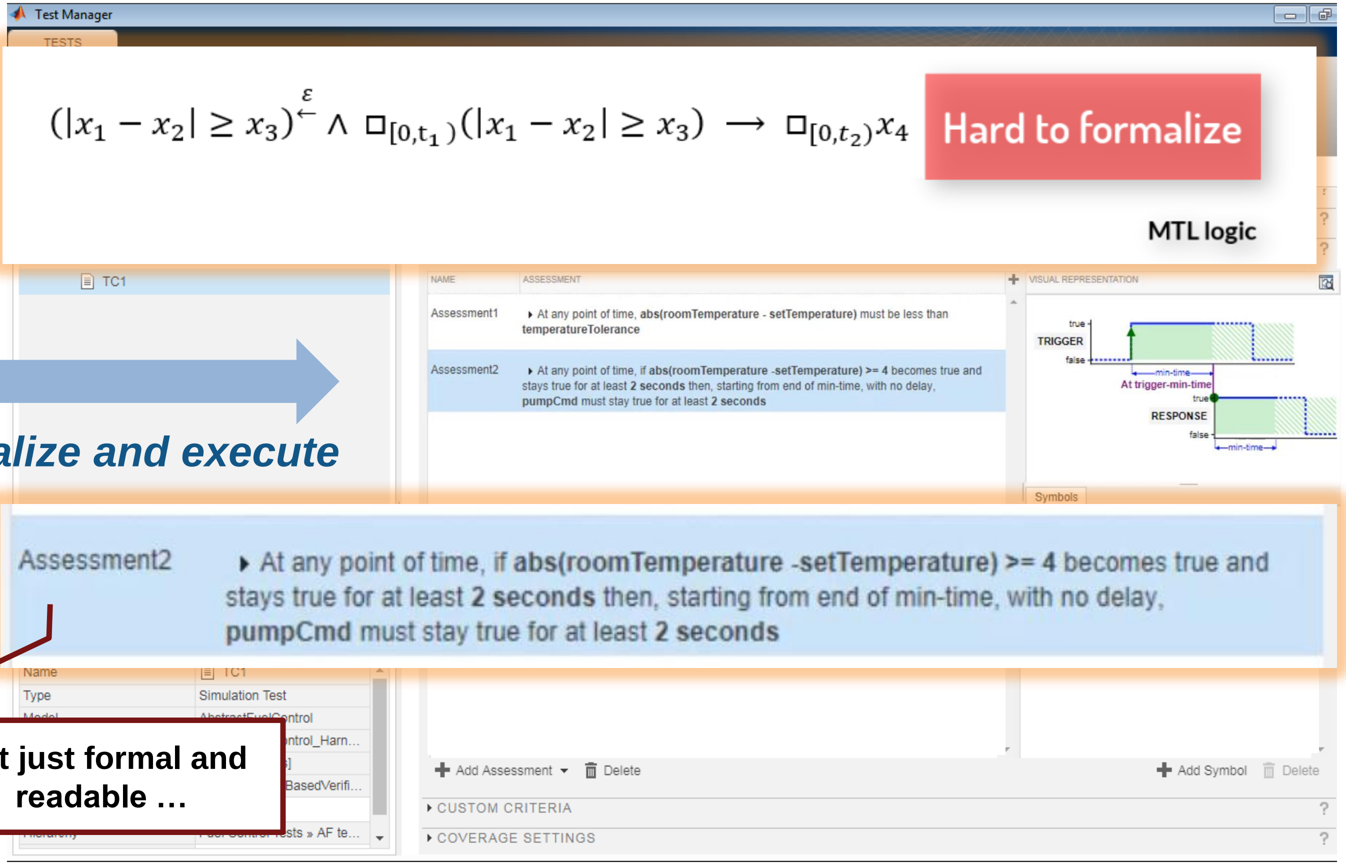
Simple concept

$$(|x_1 - x_2| \geq x_3)^{\overleftarrow{\varepsilon}} \wedge \Box_{[0,t_1)}(|x_1 - x_2| \geq x_3) \longrightarrow \Box_{[0,t_2)} x_4$$

Hard to formalize

MTL logic

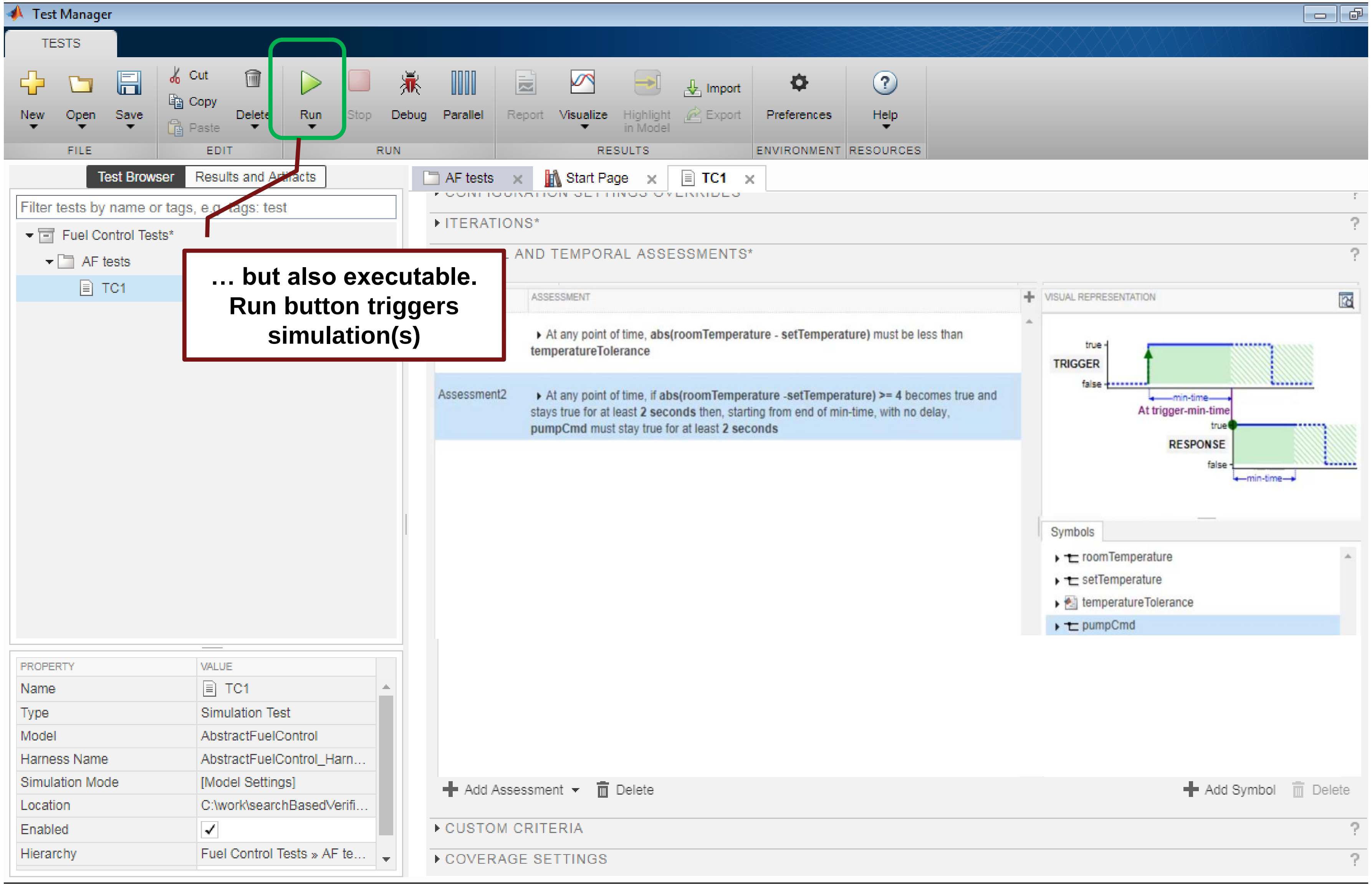
Formalize and execute

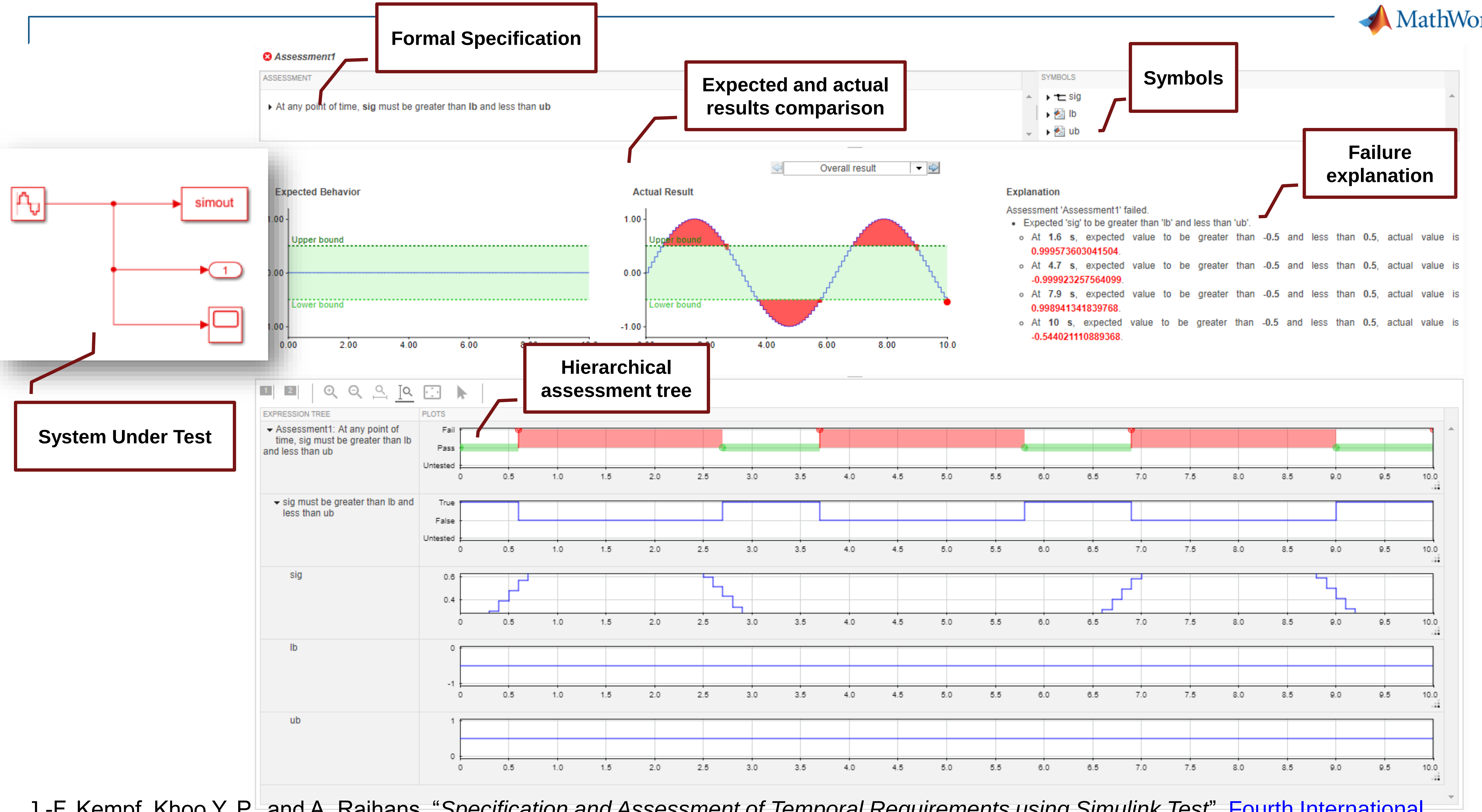


Assessment2 ▸ At any point of time, if **abs(roomTemperature -setTemperature) >= 4** becomes true and stays true for at least **2 seconds** then, starting from end of min-time, with no delay, **pumpCmd** must stay true for at least **2 seconds**

Not just formal and readable …

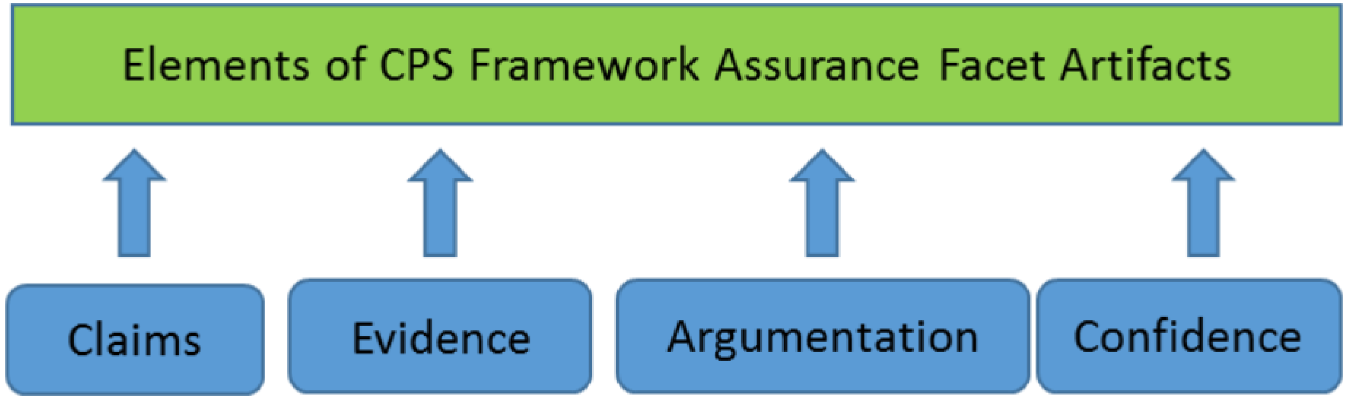
How could we formalize and execute requirements directly?



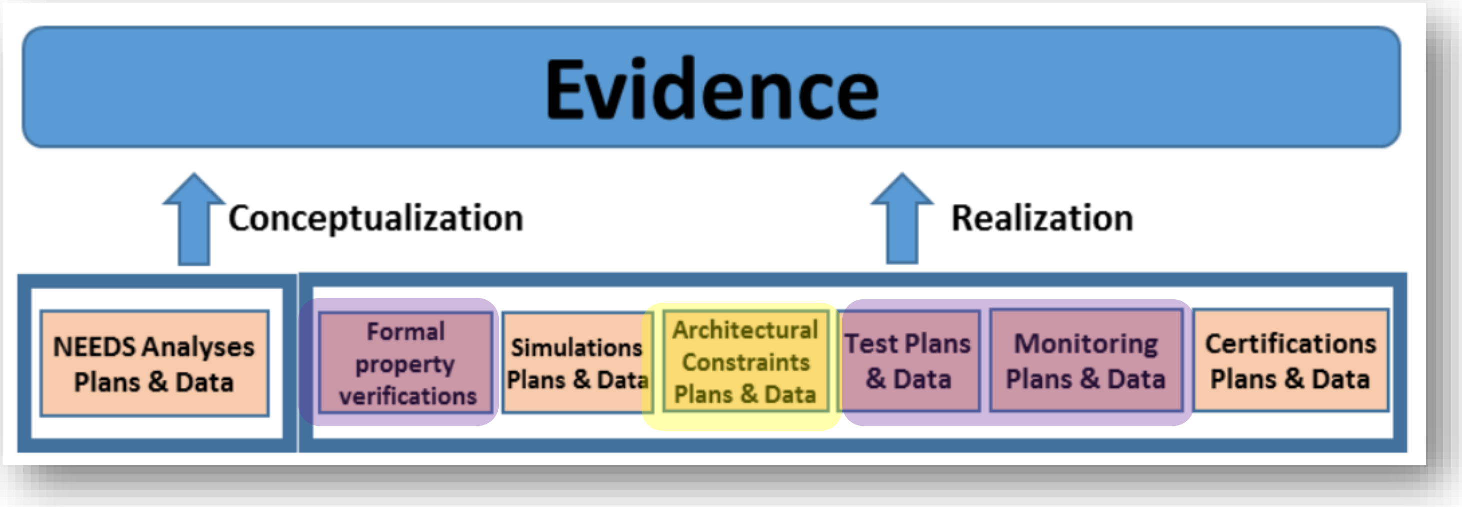


J.-F. Kempf, Khoo Y. P., and A. Rajhans, “*Specification and Assessment of Temporal Requirements using Simulink Test*”, Fourth International Workshop on Monitoring and Testing of Cyber-Physical Systems (MT-CPS 2019), part of CPS-IoT Week 2019. [Abstract (PDF)]

Conclusion



“The [Evidence] is sufficient to conclude that the [Claims] are true based on the [Argumentation] with this [Estimate of Confidence].”



System Composer

Simulink Test

https://www.nist.gov/publications/framework-cyber-physical-systems-volume-1-overview

Acknowledgments

- Architecture Modeling and Analysis
  - Ajinkya Bhave, Bruce Krogh, David Garlan, Ivan Ruchkin, Bradley Schmerl, … (research colleagues)
  - Several MathWorks colleagues

- Formalizing Specifications
  - Jyo Deshmukh, Alexandre Donze, Gerogios Fainekos, Bruce Krogh, Dejan Nickovic, Jens Oehlerking, Andre Platzer, … (research colleagues)
  - Several MathWorks colleagues

- References: https://arajhans.github.io

MathWorks®

Accelerating the pace of engineering and science