

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

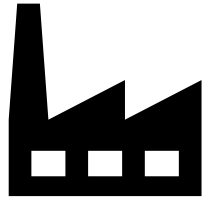
Akshay Rajhans, PhD
Principal Research Scientist
arajhans@mathworks.com
<https://arajhans.github.io>

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



Carnegie Mellon

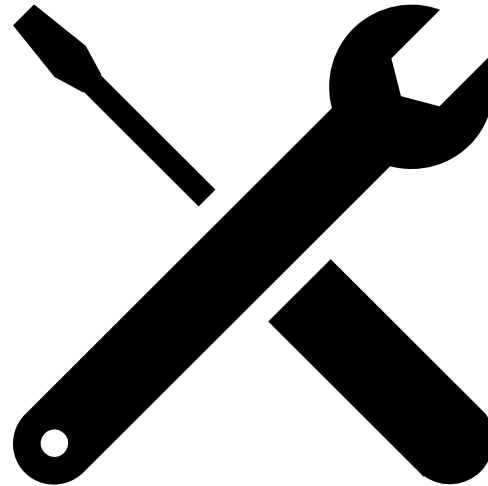


Perspective shaped by my personal career trajectory

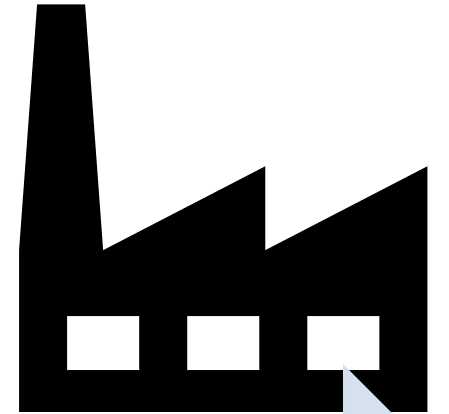
Academic
Researcher



Tool
Developer



Industry
Practitioner



Interests span this tradeoff

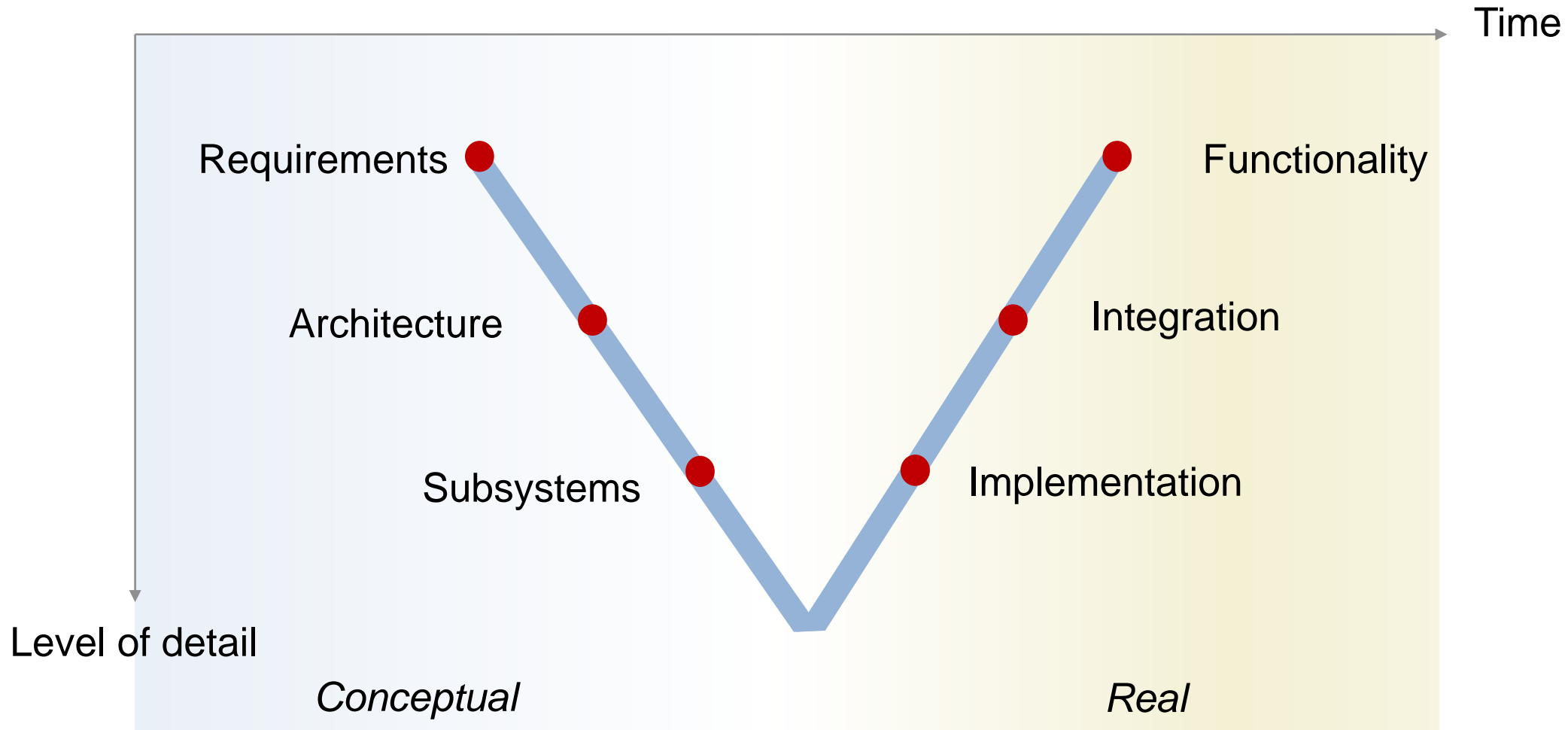
Outline

- Introduction
 - Model-Based Design context for Software and System Development

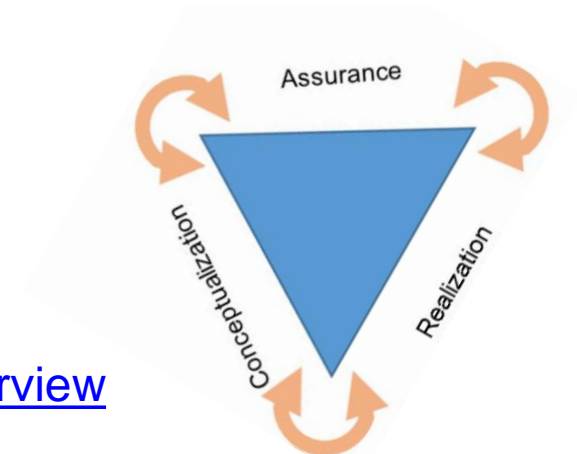
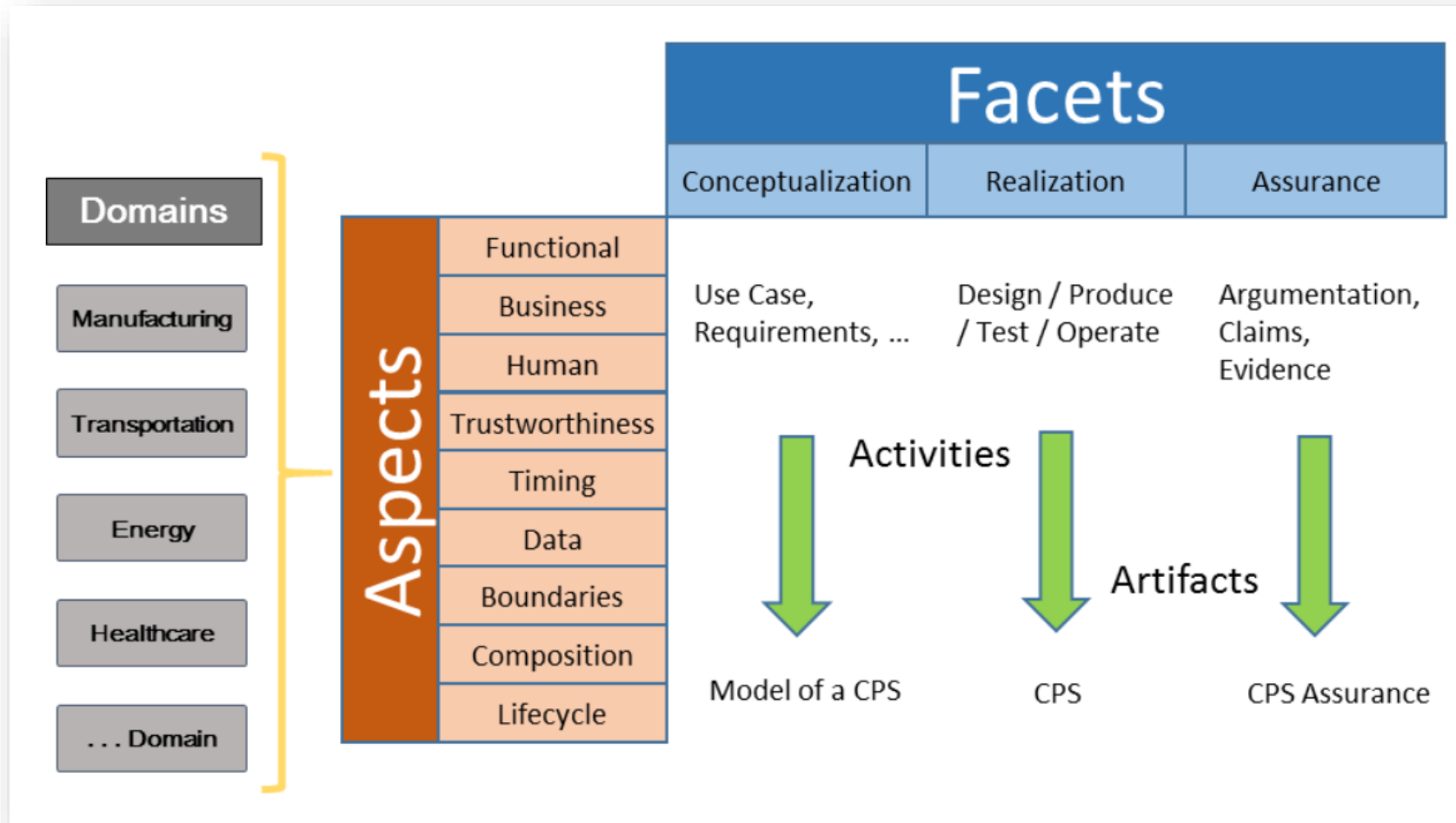
- Two new recently-released products/features related to my research
 - Architecture Design and Analysis
 - Formalizing Specifications

- Conclusion
 - Link back to the Model-Based Design context

Model-based design 'V'

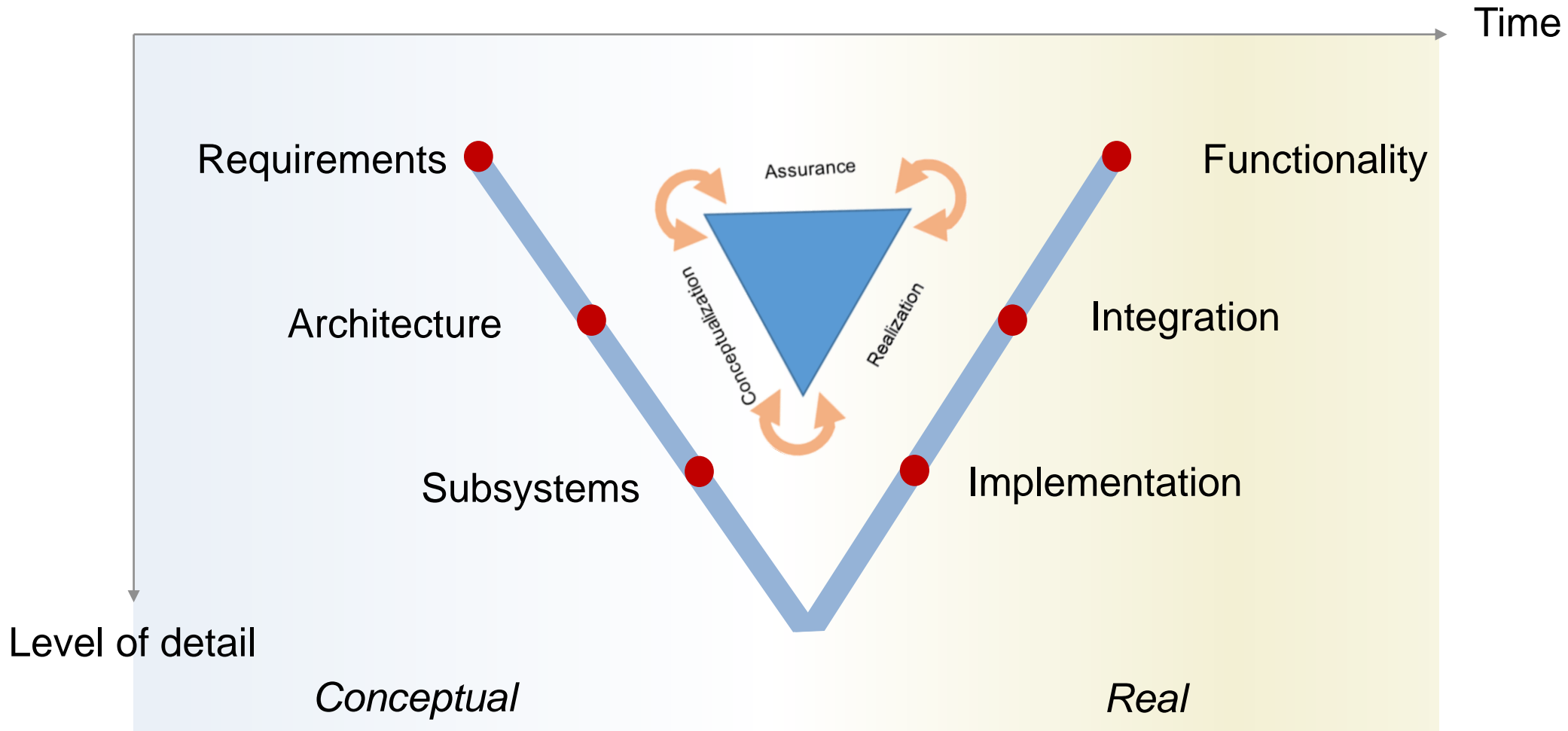


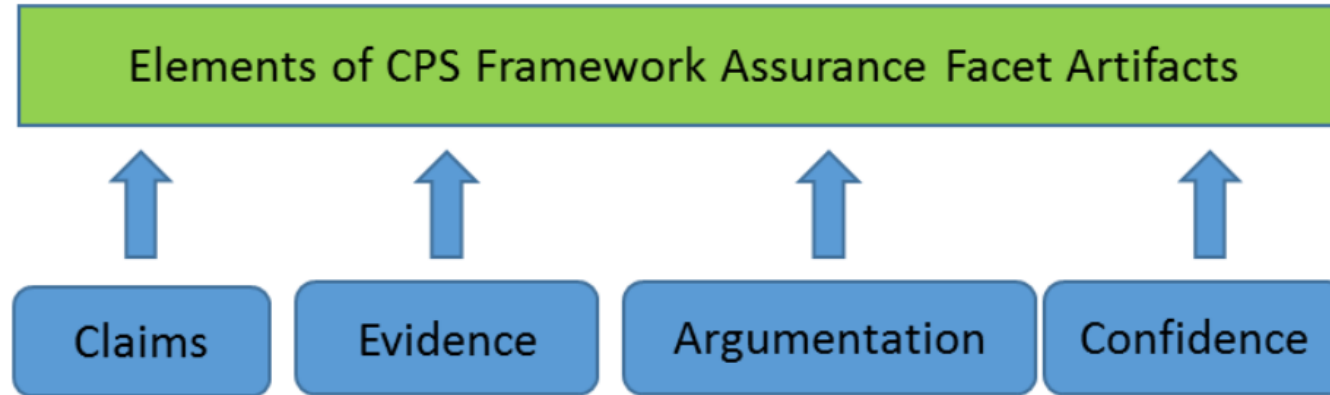
NIST CPS Framework



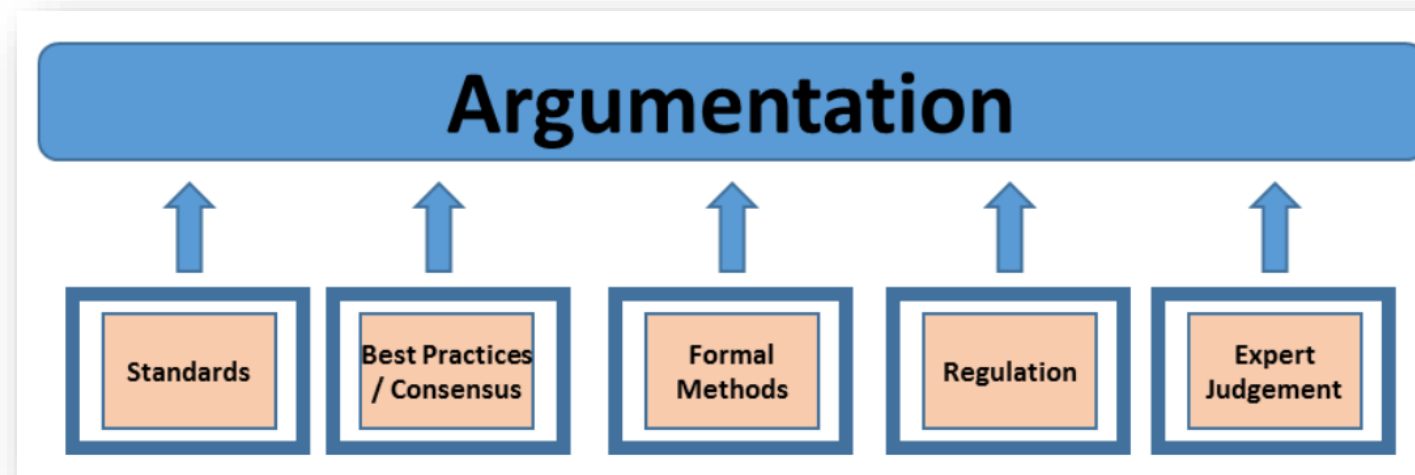
<https://www.nist.gov/publications/framework-cyber-physical-systems-volume-1-overview>

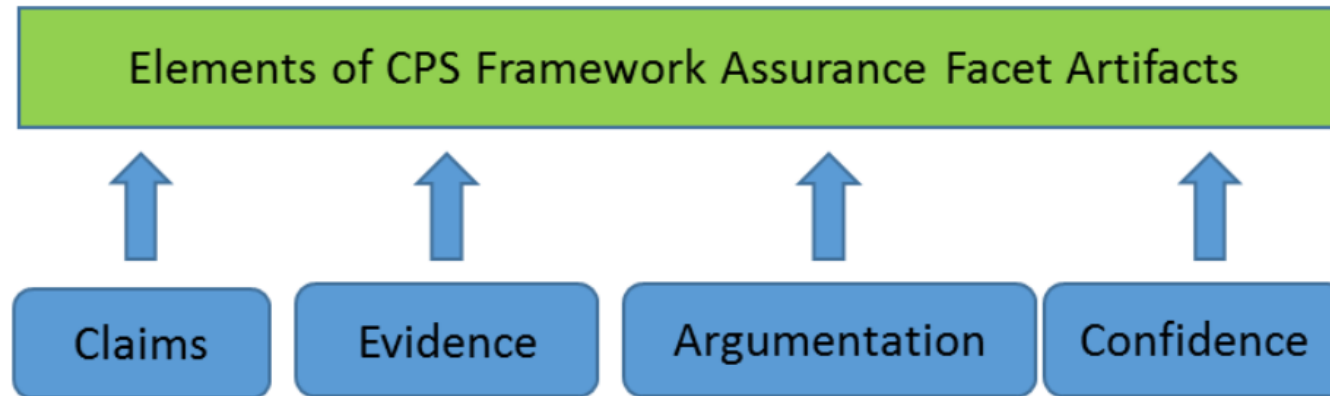
Model-based design 'V'



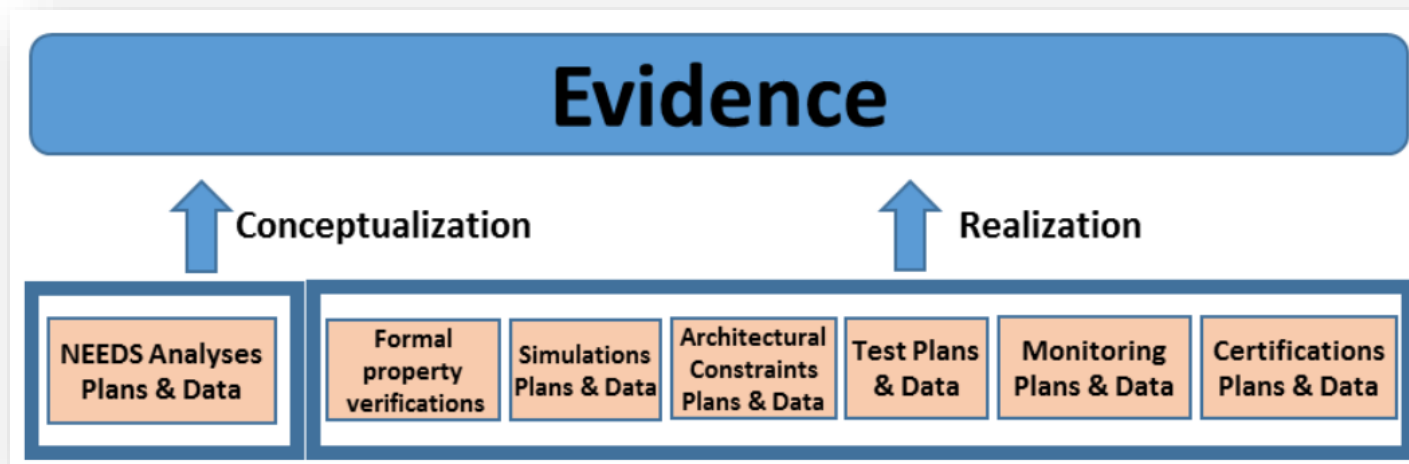


“The [Evidence] is sufficient to conclude that the [Claims] are true based on the [Argumentation] with this [Estimate of Confidence].”





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



Two new automated functionalities related to my own work

The screenshot shows the ABLÉ website interface. At the top, there's a navigation menu with 'Home', 'Research', 'Publications', 'Members', 'Related', and 'Software'. Below that is a search bar with a dropdown menu set to 'All' and a 'Go' button. The main content area displays 'Publications related to 'Cyberphysical Systems'' with options to order by 'Type' or 'Year'. There are radio buttons for 'Only', 'And', and 'Or' related to a dropdown menu set to 'none'. The results are categorized into 'Article (Journal)' and 'InProceedings'. The first article is by Akshay Rajhans, Shang-Wen Cheng, Bradley Schmerl, David Garlan, Bruce Krogh, Clarence Agbi, and Ajinkya Y. Bhavé, titled 'An Architectural Approach to the Design and Analysis of Cyber-Physical Systems'. The second article is by Akshay Rajhans, Ajinkya Y. Bhavé, Ivan Ruchkin, Bruce Krogh, David Garlan, Andre Platzer, and Bradley Schmerl, titled 'Supporting Heterogeneity in Cyber-Physical Systems Architectures'. The third article is by Ajinkya Y. Bhavé, David Garlan, Bruce Krogh, Akshay Rajhans, and Bradley Schmerl, titled 'Augmenting Software Architectures with Physical Components'.

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

Architecture Modeling and Analysis

The banner features a cityscape background with the text '4TH WORKSHOP ON MONITORING AND TESTING OF CYBER-PHYSICAL SYSTEMS'. Below this, it says 'Part of CPS-IoT Week 2019' and 'April 15, 2019 - Montreal, Canada'. At the bottom right, the word 'COMMITTEES' is written in large letters. Below 'COMMITTEES', the text 'Program Chairs' is followed by a list of names and affiliations: Tommaso Dreossi, University of California, Berkeley; and Akshay Rajhans, MathWorks.

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

Formalizing Specifications

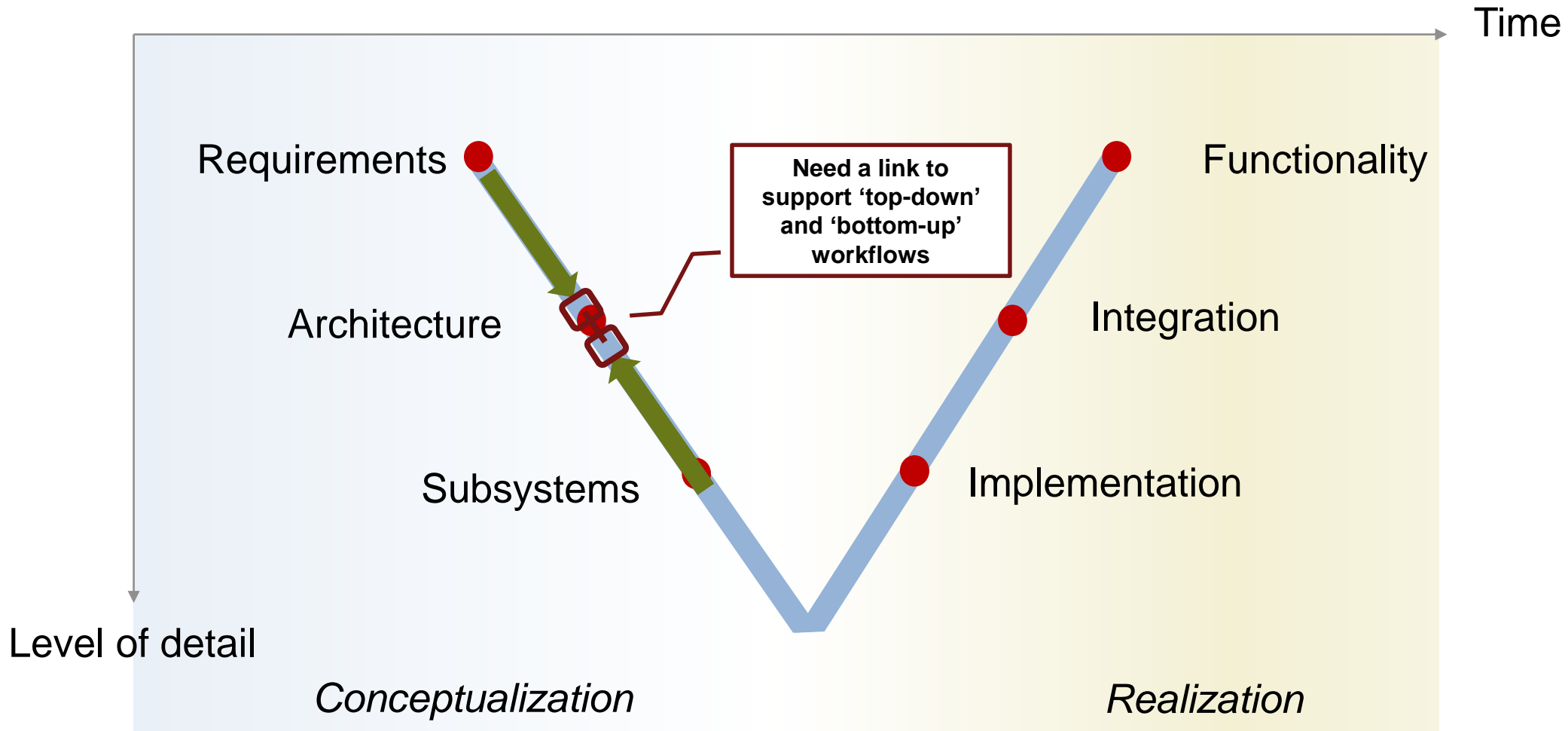
Architecture Modeling and Analysis

The screenshot displays the ABL website interface. At the top, the logo 'ABLE' is prominent, with the tagline 'Architecture Based Languages and Environments'. Below the logo, there is a navigation menu with options: Home, Research, Publications, Members, Related, and Software. A search bar is located below the navigation, with a search box, a dropdown menu set to 'All', a 'Go' button, and a note '(word length ≥ 3)'. To the right of the search bar is a 'Login' button. The main content area is titled 'Publications related to 'Cyberphysical Systems'' and includes an 'Order by:' dropdown menu with options 'Type' and 'Year'. Below this, there are radio buttons for 'Only', 'And', and 'Or', followed by a 'related to:' dropdown menu set to 'none' and a 'Show' button. The results are categorized into 'Article (Journal)' and 'InProceedings'. Under 'Article (Journal)', there are three entries: 1. Akshay Rajhans, Shang-Wen Cheng, Bradley Schmerl, David Garlan, Bruce Krogh, Clarence Agbi and Ajinkya Y. Bhav. 'An Architectural Approach to the Design and Analysis of Cyber-Physical Systems'. In *Electronic Communications of the EASST*, Vol. 21: Multi-Paradigm Modeling, 2009. 2. Akshay Rajhans, Ajinkya Y. Bhav, Ivan Ruchkin, Bruce Krogh, David Garlan, Andre Platzer and Bradley Schmerl. 'Supporting Heterogeneity in Cyber-Physical Systems Architectures'. In *IEEE Transactions on Automatic Control*, Vol. 59(12):3178--3193, December 2014. Also available at <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6882828>. Under 'InProceedings', there is one entry: 3. Ajinkya Y. Bhav, David Garlan, Bruce Krogh, Akshay Rajhans and Bradley Schmerl. 'Augmenting Software Architectures with Physical Components'. In *Proceedings of the Embedded Real Time Software and Systems Conference (ERTS^2 2010)*, 19-21 May 2010.

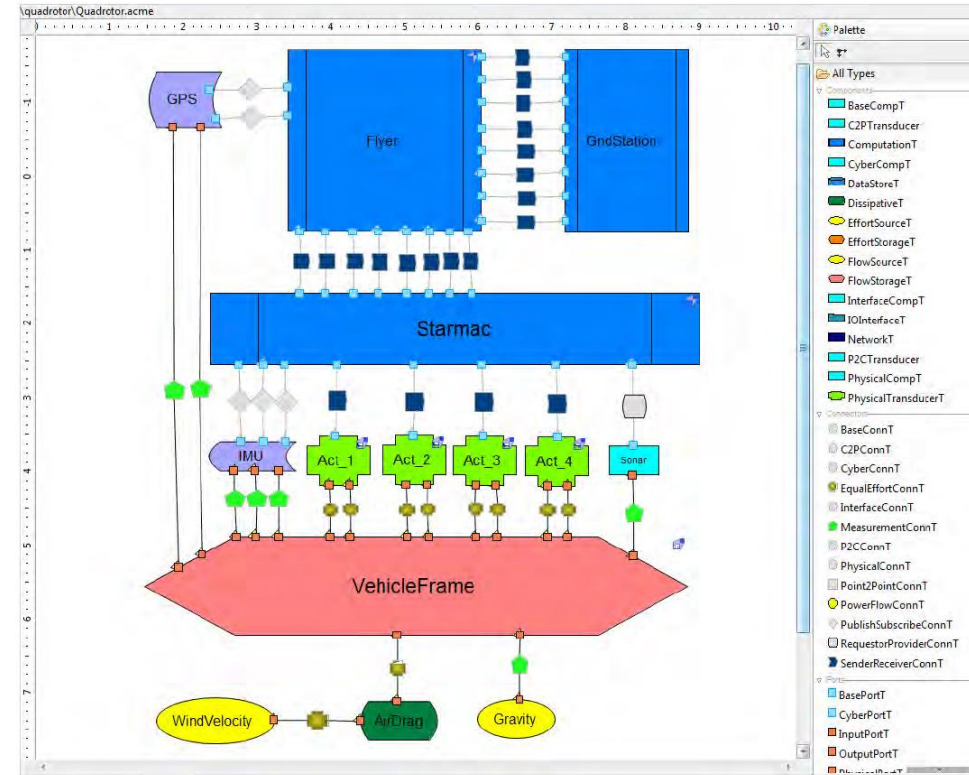
<http://acme.able.cs.cmu.edu/pubs/show.php?keyword=Cyberphysical%20Systems>

Architecture Modeling and Analysis

Model-based design 'V'



Architecture modeling of the STARMAC quadrotor



Augmenting Software Architectures with Physical Components

Ajinkya Bhave¹, David Garlan², Bruce H. Krogh¹, Akshay Rajhans¹, Bradley Schmerl²

ERTS² '10

¹Dept. of Electrical and Computer Engineering

²School of Computer Science

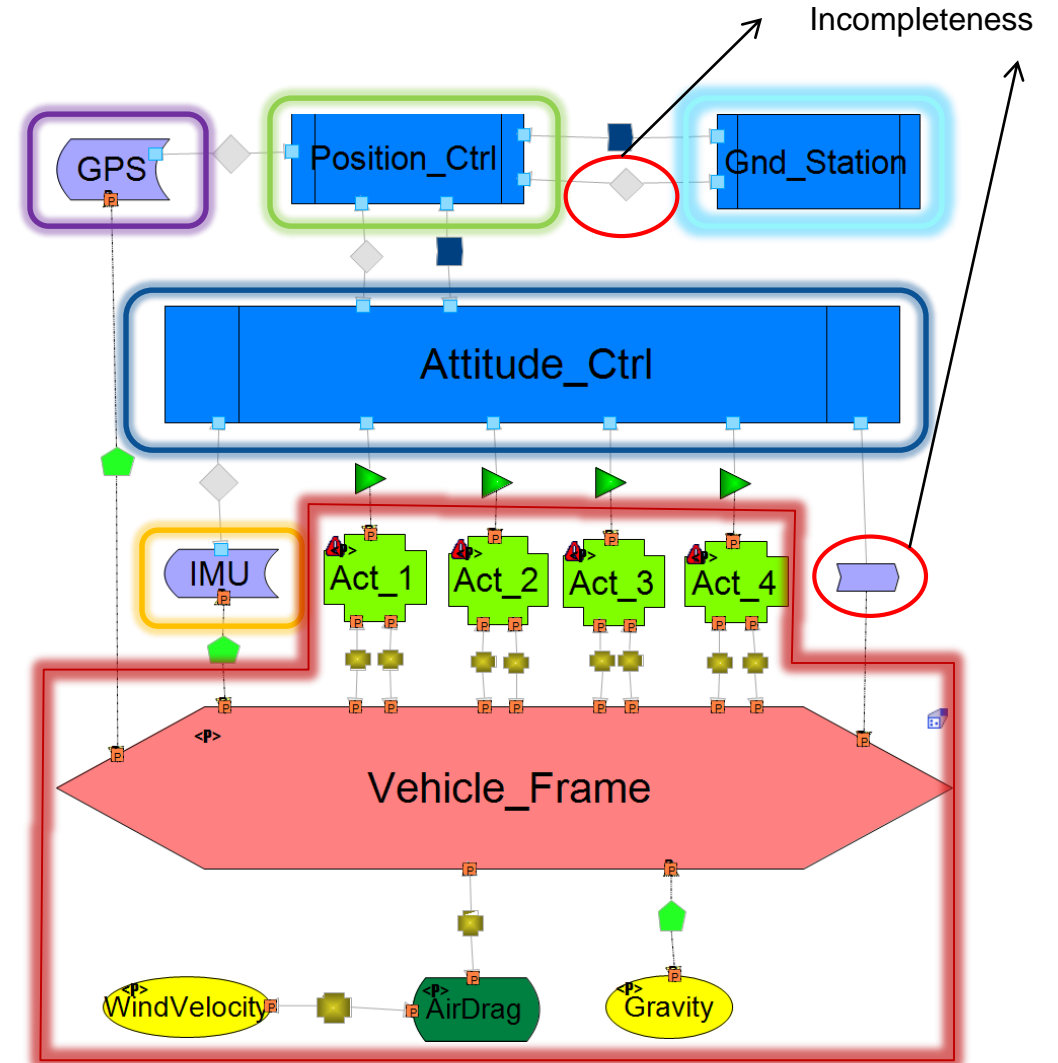
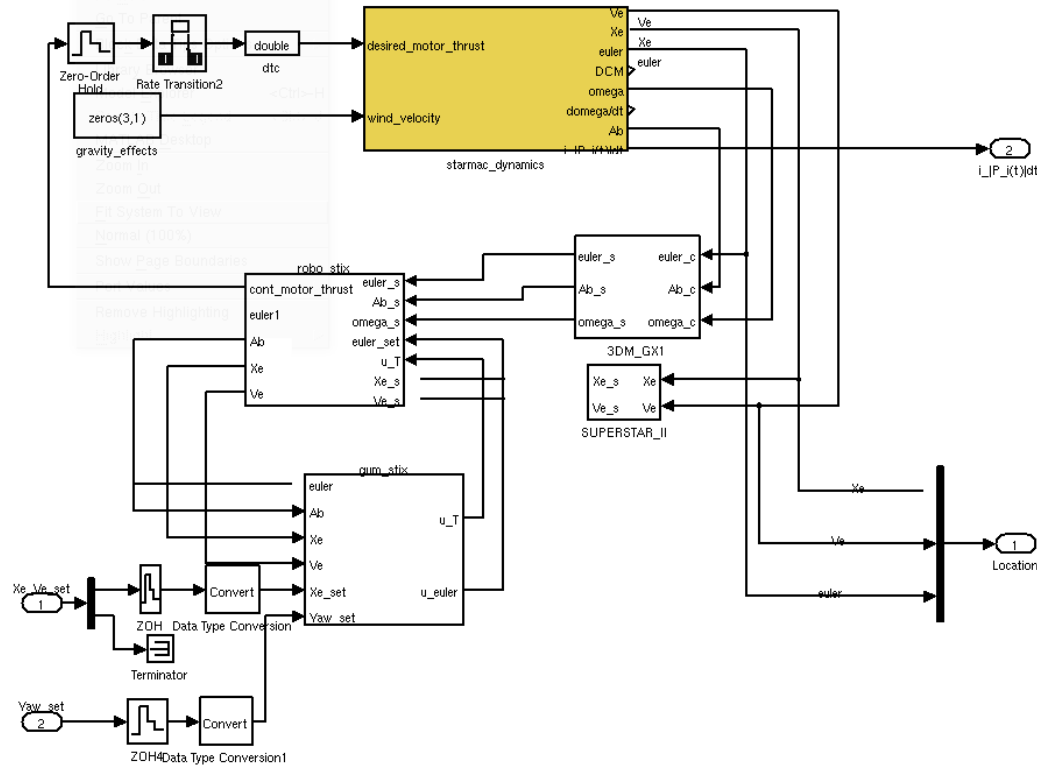
Carnegie Mellon University

Pittsburgh, PA 15213-3890 USA

email: {ajinkya@ | garlan@cs. | krogh@ece. | arajhans@ece. | schmerl@cs.}cmu.edu

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

Simulink Architecture View



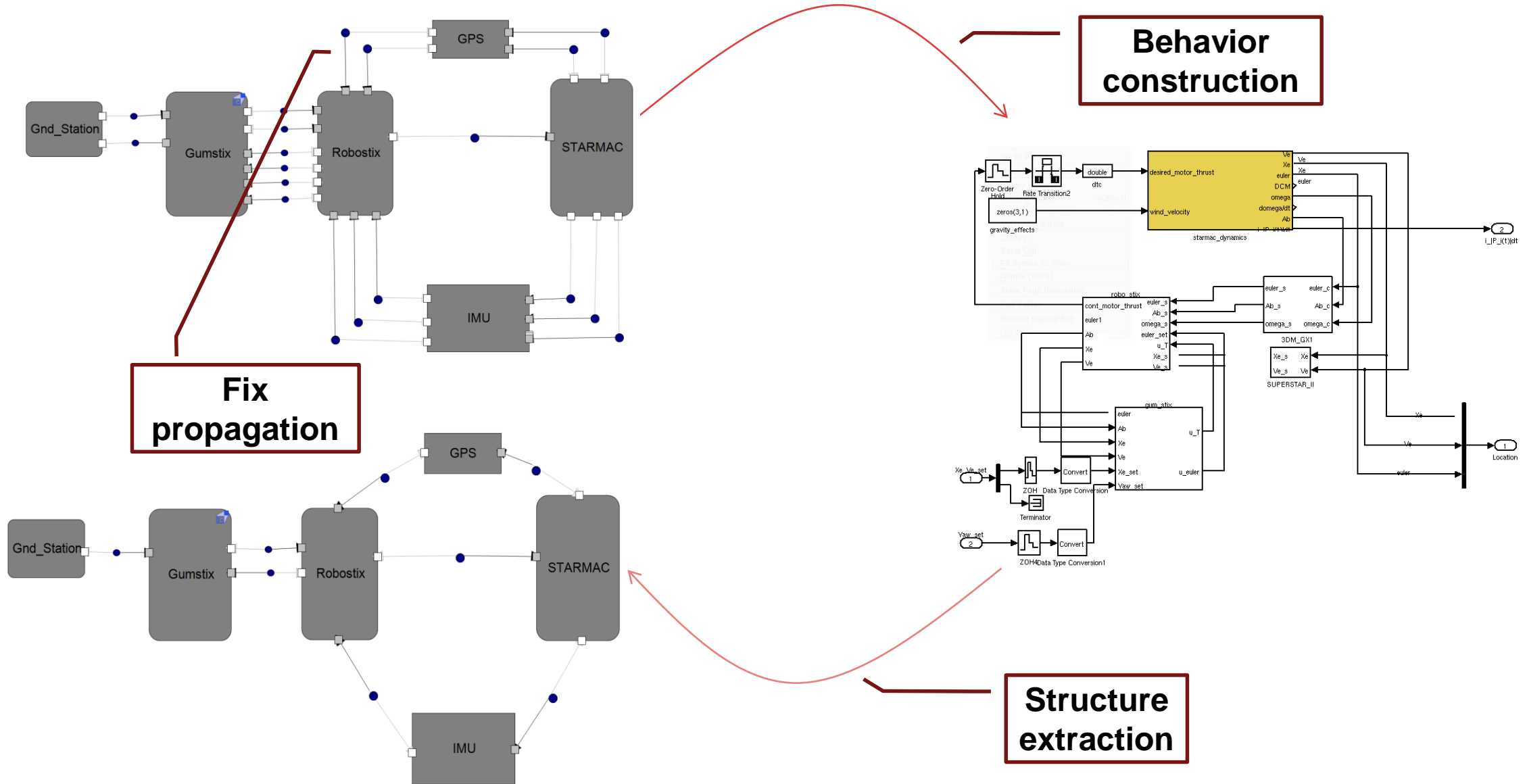
View Consistency in Architectures for Cyber-Physical Systems

ICCPs '11

Ajinkya Bhawe, Bruce H. Krogh

David Garlan, Bradley Schmerl

Simulink Architecture \leftrightarrow Simulink Model: Manual Step



System Composer

The screenshot displays the System Composer application window. At the top left, a Command Window shows the command `>> systemcomposer` and the `fx` prompt. The main window has a menu bar (File, Edit, View, Display, Architecture, Simulation, Analysis, Code, Tools, Help) and a toolbar with various icons. On the left, the Model Browser shows a tree view with 'untitled' highlighted. The central area is the Editing Canvas, which is currently blank. On the right, the Property Inspector is open, showing the 'Architecture' tab with a table of properties for the 'Main' component. Below it, the Interface Editor shows the source file 'untitled.slx'. The status bar at the bottom indicates 'Ready' and '90%' zoom.

Command Window:

```
>> systemcomposer
fx >>
```

Model Browser:

- untitled

Editing Canvas:

Property Inspector:

NAME	VALUE
Main	
Name	Property Inspector
Stereotype	

Interface Editor:

Source: untitled.slx

Describe abstract component interfaces and allocate to ports

The screenshot shows the System Composer environment. On the left, a block diagram features three main components: 'Path Planning', 'User Interface', and 'Power System'. 'Path Planning' has a 'SelfLoc' input port and a 'Waypts' output port. 'User Interface' has a 'Waypts' input port. 'Power System' has an 'InBus' input port and a 'Health' output port. A dashed orange line connects the 'Waypts' port of 'Path Planning' to the 'Waypts' port of 'User Interface'. Another dashed orange line connects the 'Waypts' port of 'Path Planning' to the 'WaypointData' interface in the 'Interfaces' inspector.

The 'Property Inspector' on the right shows the 'Port' configuration for the selected 'Waypts' port. It has two tabs: 'Architecture' and 'Info'. The 'Info' tab is active, displaying a table with the following data:

NAME	VALUE
Main	
Name	Waypts
Interface	
Name	Create or Select ..
Action	PROVIDE
Stereotype	Add..

Below the 'Property Inspector' is the 'Interfaces' inspector, which is highlighted with an orange border. It shows a list of interfaces under the source 'untitled.slx':

- LocationInterface
 - x
 - y
 - orientation
- PowerInterface
- WaypointData

The 'WaypointData' interface is selected, and a tooltip 'Assign to Selected Port(s)' is visible. A dashed arrow points from the 'Import from workspace or file' text to the 'Import' icon in the 'Interfaces' inspector. Another dashed arrow points from the 'Save/link to data dictionary' text to the 'Save' icon in the 'Interfaces' inspector.

Store in the model when sketching

Import from workspace or file

Save/link to data dictionary

Allocate requirements to architectures (using Simulink Requirements)

The screenshot displays the Simulink Requirements tool interface for a model named 'simpleRobot'. The main workspace shows an architectural diagram with four components: Localization, Path Planning, User Interface, and Power System. Localization and Path Planning are connected via a 'SelfLoc' port. Path Planning and User Interface are connected via a 'Waypts' port. Localization and Path Planning are connected to an 'OutBus' port. Path Planning and Power System are connected via a 'Health' port. Requirements are allocated as follows:

- Requirement #1: 'Self location known every 10 ms' is implemented by the Localization component.
- Requirement #3: 'Report available power to UI' is implemented by the Power System component.

The 'Requirements - simpleRobot' window is open, showing a table of requirements:

Index	ID	Summary
1	#1	Self location known every 10 ms
2	#2	Should provide an iPhone app
3	#3	Report available power to UI

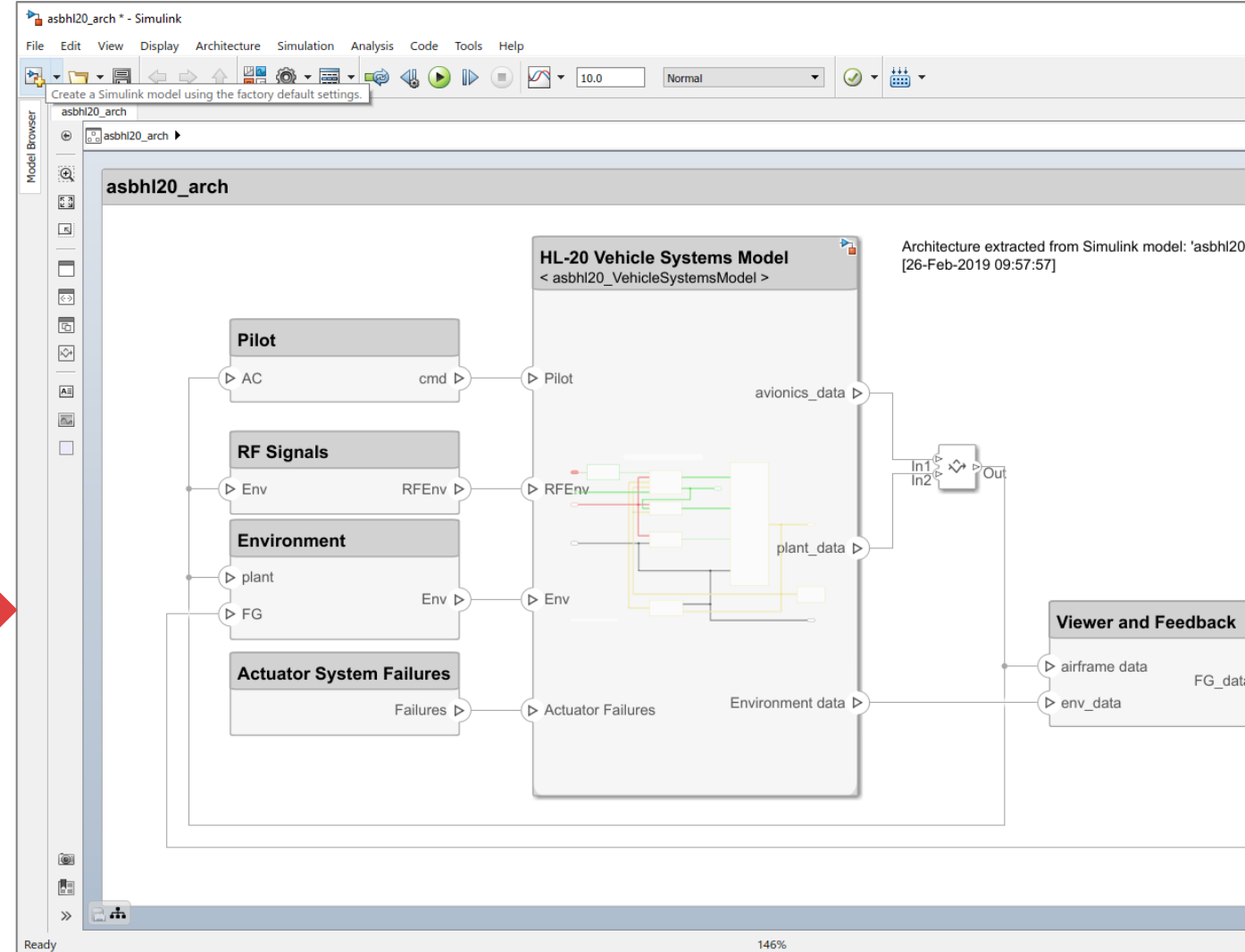
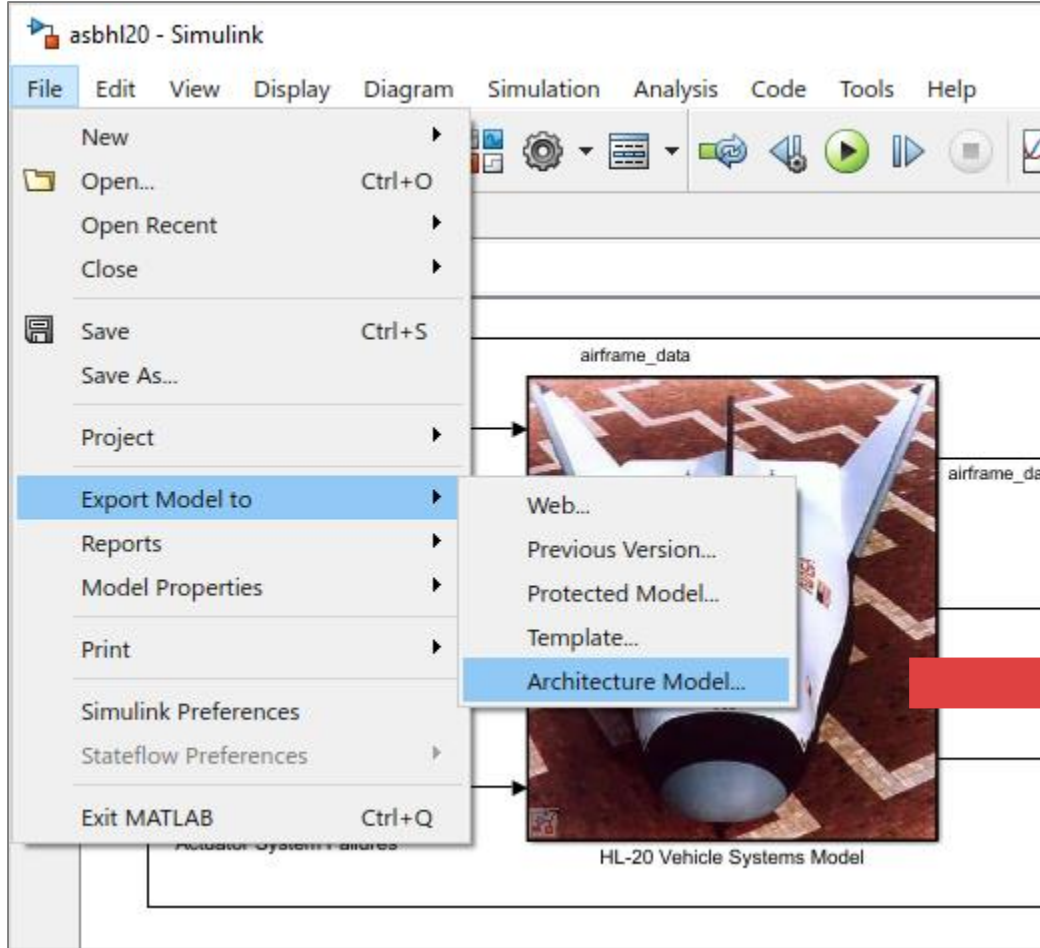
The 'Property Inspector' on the right shows the 'Architecture' tab with the following information:

NAME	VALUE
Name	simpleRobot
Stereotype	Add..

The 'Interfaces' window on the right shows the source 'simpleRobot.slx' and the following interfaces:

- LocationInterface
 - x
 - y
 - orientation
- PowerInterface
- WaypointData

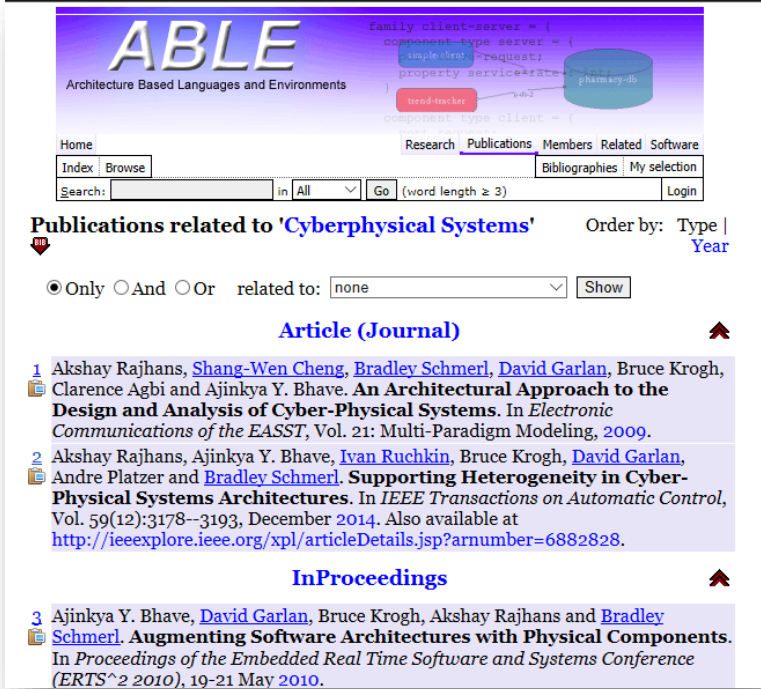
Simulink to Architecture Automation



Architecture to Simulink Automation

Interfaces shared across models via data dictionaries

Two new automated functionalities related to my own work



The screenshot shows the ABLE (Architecture Based Languages and Environments) website. At the top, there is a navigation menu with 'Home', 'Research', 'Publications', 'Members', 'Related', and 'Software'. Below the menu is a search bar with a dropdown menu set to 'All' and a 'Go' button. The main content area displays 'Publications related to 'Cyberphysical Systems'' with an 'Order by:' dropdown set to 'Type'. There are radio buttons for 'Only', 'And', and 'Or' related to a dropdown menu set to 'none'. The results are categorized into 'Article (Journal)' and 'InProceedings'. The first article is by Akshay Rajhans, Shang-Wen Cheng, Bradley Schmerl, David Garlan, Bruce Krogh, Clarence Agbi and Ajinkya Y. Bhawe, titled 'An Architectural Approach to the Design and Analysis of Cyber-Physical Systems'. The second article is by Akshay Rajhans, Ajinkya Y. Bhawe, Ivan Ruchkin, Bruce Krogh, David Garlan, Andre Platzer and Bradley Schmerl, titled 'Supporting Heterogeneity in Cyber-Physical Systems Architectures'. The third article is by Ajinkya Y. Bhawe, David Garlan, Bruce Krogh, Akshay Rajhans and Bradley Schmerl, titled 'Augmenting Software Architectures with Physical Components'.

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

Architecture Modeling and Analysis



The screenshot shows the website for the 4th Workshop on Monitoring and Testing of Cyber-Physical Systems. The background is a cityscape. The text on the page reads: '4TH WORKSHOP ON MONITORING AND TESTING OF CYBER-PHYSICAL SYSTEMS', 'Part of CPS-IoT Week 2019', and 'April 15, 2019 - Montreal, Canada'. Below this, the word 'COMMITTEES' is displayed in large letters. Underneath, the 'Program Chairs' are listed: Tommaso Dreossi, University of California, Berkeley, and Akshay Rajhans, MathWorks.

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

Formalizing Specifications

Formalizing Specifications



4TH WORKSHOP ON MONITORING AND TESTING OF
CYBER-PHYSICAL SYSTEMS

Part of CPS-IoT Week 2019
April 15, 2019 - Montreal, Canada

COMMITTEES

Program Chairs

- Tommaso Dreossi, University of California, Berkeley
- Akshay Rajhans, MathWorks

<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

Requirements

- ▼ 2.12 Heat pump requirements
 - 2.12.1 Temperature bounds
 - 2.12.2 Controller response



Input Scenarios



Design/Implementation



Dynamic Testing

1	Inpgt
3	

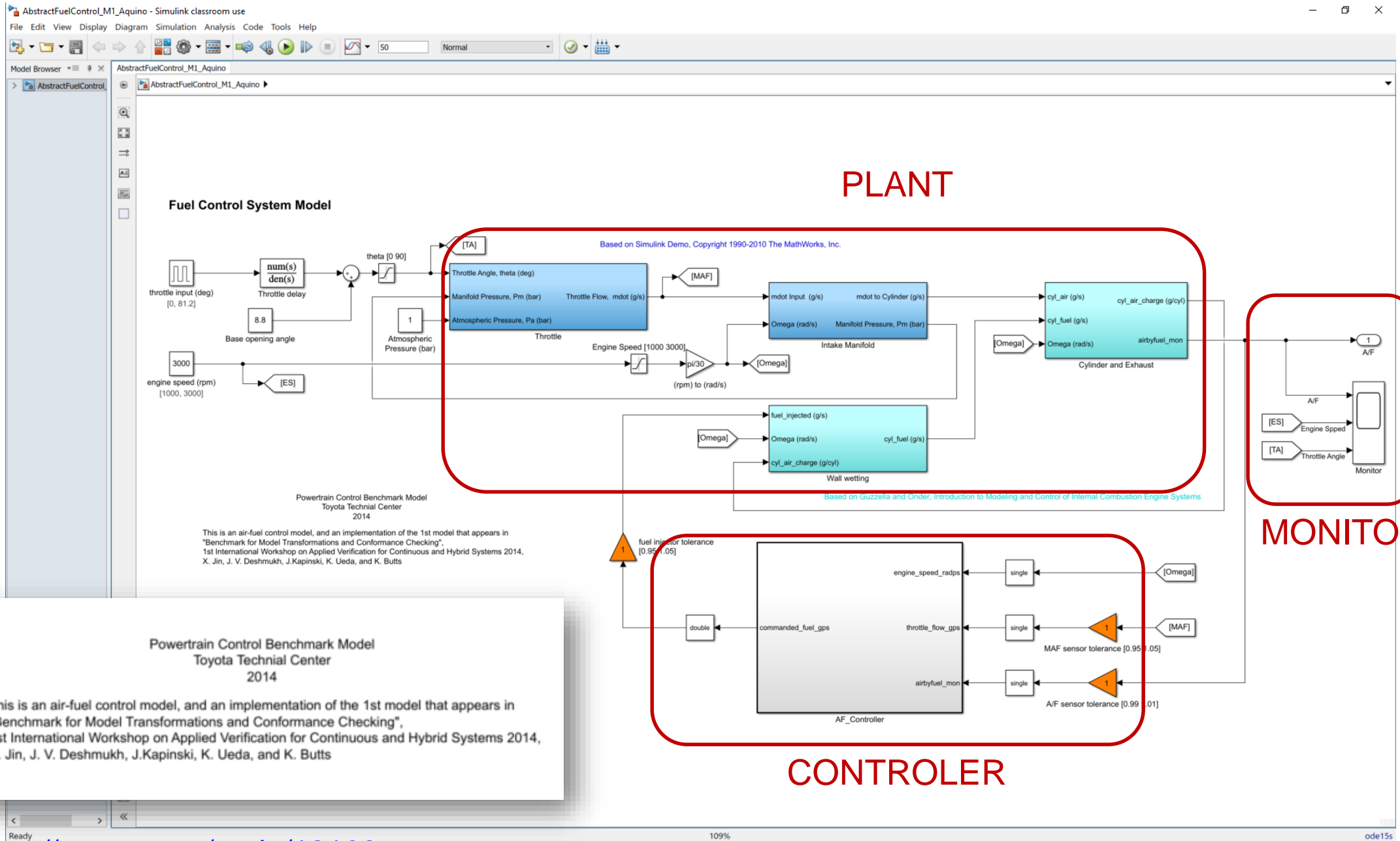
and more!

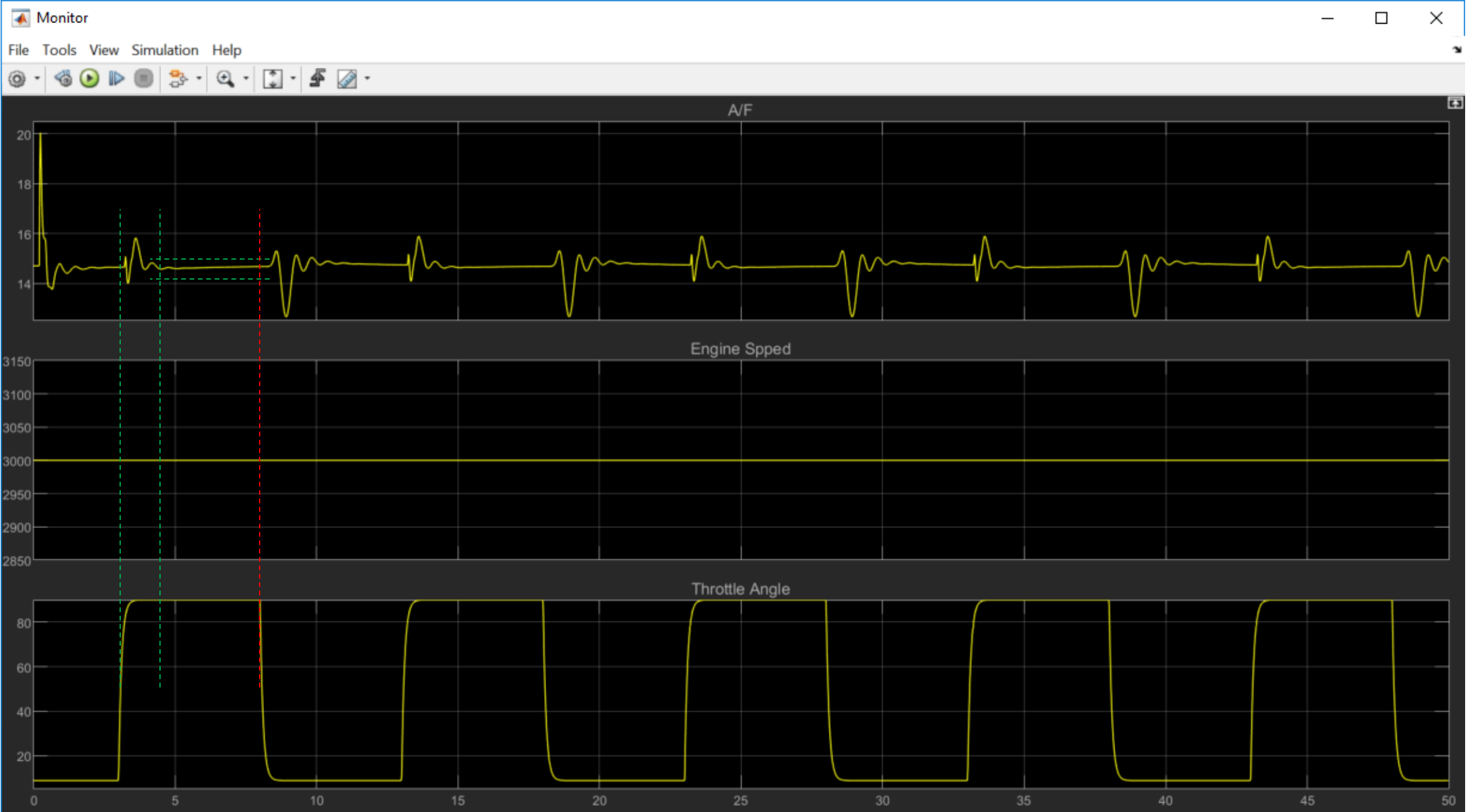
Toyota Air-Fuel Ratio Control Example

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 Technical 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

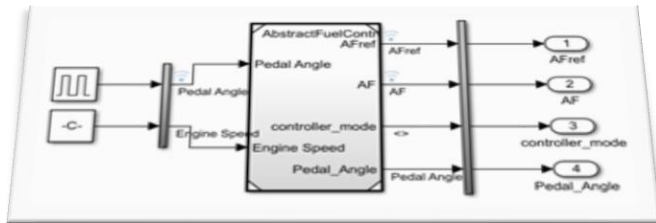




How could we formalize and execute requirements directly?

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

PROPERTY	VALUE
Name	TC1
Type	Simulation Test
Model	AbstractFuelControl
Harness Name	AbstractFuelControl_Harn...
Simulation Mode	[Model Settings]
Location	C:\work\searchBasedVerifi...
Enabled	<input checked="" type="checkbox"/>
Hierarchy	Fuel Control Tests > AF te...

Formal property specification for Simulink models

1. Using MATLAB scripts

- Breach
- Toyota ARCH'14 Benchmark

%% Writing a Simple STL Specification

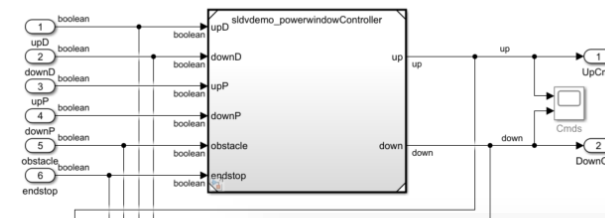
% First we define a predicate stating that AF is above 1% of ARef
 AF_not_ok = STL_Formula('AF_ok', 'abs(AF[t]- ARef[t]) < 0.01*14.7')

```
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

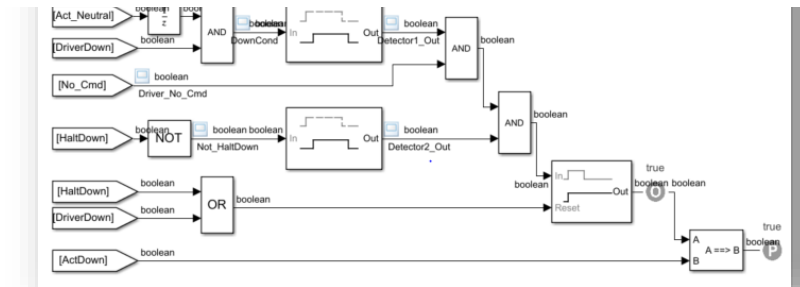


Requirement (Autodown)




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)



Start Page x New Test Case 1 x

EN...	NAME	ASSESSMENT	REQUIREM...	+
Logical Assessments				
	Bounds check	Check min/max bounds for signals and expressions		
	Custom	Check if a custom expression holds true for all time steps		
Temporal Assessments				
	Trigger-response	Check for a signal response once a trigger has been detected		

+ Add Assessment ▾ Delete

>> sltestmgr

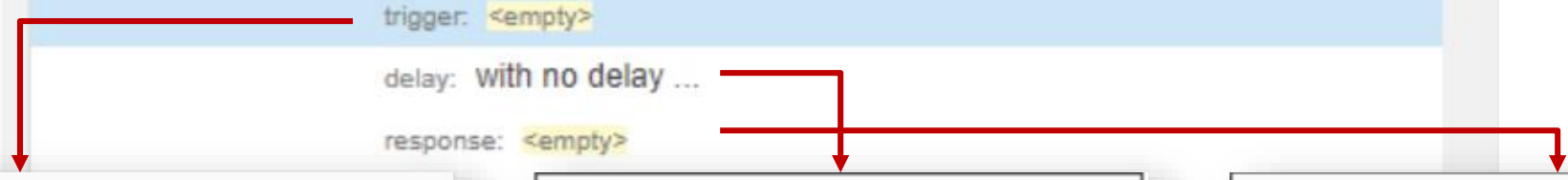
$\square_{[t_0, t_f]} x \in \langle a, b \rangle$ where $\langle \in \{\leq, <\}$

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

Start Page x New Test Case 1 x

EN...	NAME	ASSESSMENT	REQUIREM...
<input checked="" type="checkbox"/>	Assessm...	At any point of time ...	None
		trigger: <empty>	
		delay: with no delay ...	
		response: <empty>	

>> sltestmgr



- whenever is true
- becomes true
- becomes true and stays true for at least
- becomes true and stays true for at most
- becomes true and stays true for between

- with no delay
- with a delay of at most
- with a delay of between

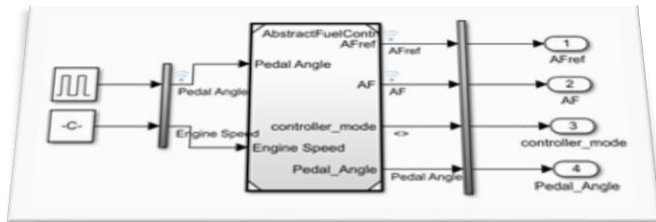
- must be true
- must stay true for at least
- must stay true for at most
- must stay true for between
- must stay true until

+ Add Assessment x Delete

How could we formalize and execute requirements directly?

Requirements

- The difference between the room temperature and the set temperature should never exceed 6 degrees.
- 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

PROPERTY	VALUE
Name	TC1
Type	Simulation Test
Model	AbstractFuelControl
Harness Name	AbstractFuelControl_Harn...
Simulation Mode	[Model Settings]
Location	C:\work\searchBasedVerifi...
Enabled	<input checked="" type="checkbox"/>
Hierarchy	Fuel Control Tests > AF te...

How could we formalize and execute requirements directly?

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

Simple concept

$$(|x_1 - x_2| \geq x_3)^{\epsilon} \wedge \square_{[0,t_1]}(|x_1 - x_2| \geq x_3) \rightarrow \square_{[0,t_2]}x_4$$

Hard to formalize

MTL logic

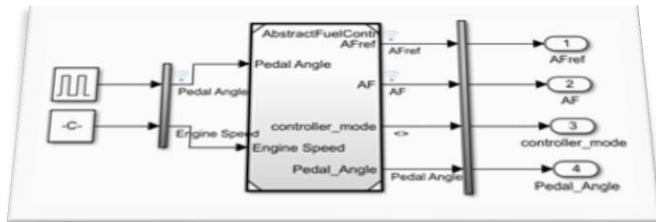
Requirements

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

The screenshot shows the Test Manager interface with a table of assessments and a visual representation of a timing diagram. The table lists two assessments: Assessment1 and Assessment2. Assessment2 is highlighted with a blue box and contains the text: "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". The visual representation shows a timing diagram with a TRIGGER signal that goes true for a duration of min-time, followed by a RESPONSE signal that goes true for a duration of min-time.

Not just formal and readable ...



System Under Test

How could we formalize and execute requirements directly?

The screenshot shows the Test Manager interface with the 'Run' button highlighted in green. A callout box points to the 'Run' button with the text: **... but also executable. Run button triggers simulation(s)**.

The interface displays a test hierarchy in the 'Test Browser' on the left, including 'Fuel Control Tests*' and 'AF tests' containing 'TC1'. The main area shows 'ASSESSMENT' details for 'TC1', including a table of assessments and a 'VISUAL REPRESENTATION' section with a timing diagram.

PROPERTY	VALUE
Name	TC1
Type	Simulation Test
Model	AbstractFuelControl
Harness Name	AbstractFuelControl_Harn...
Simulation Mode	[Model Settings]
Location	C:\work\searchBasedVerifi...
Enabled	<input checked="" type="checkbox"/>
Hierarchy	Fuel Control Tests » AF te...

ASSESSMENT

- At any point of time, $\text{abs}(\text{roomTemperature} - \text{setTemperature})$ must be less than $\text{temperatureTolerance}$
- Assessment2: At any point of time, if $\text{abs}(\text{roomTemperature} - \text{setTemperature}) \geq 4$ 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

VISUAL REPRESENTATION

The timing diagram shows a 'TRIGGER' signal that transitions from false to true and then back to false. A 'RESPONSE' signal transitions from false to true and then back to false. The duration of the true state for both signals is labeled as 'min-time'. A note indicates 'At trigger-min-time'.

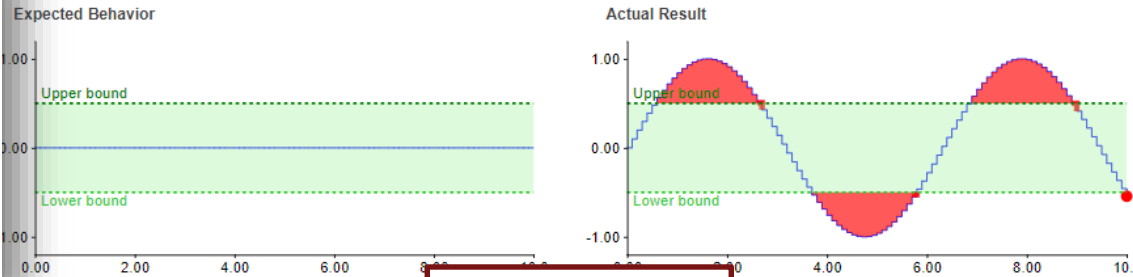
Symbols

- roomTemperature
- setTemperature
- temperatureTolerance
- pumpCmd

Formal Specification

Assessment1
 ASSESSMENT
 At any point of time, sig must be greater than lb and less than ub

Expected and actual results comparison



Symbols

SYMBOLS
 sig
 lb
 ub

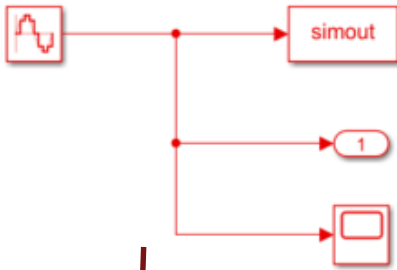
Failure explanation

Explanation
 Assessment 'Assessment1' failed.
 • Expected 'sig' to be greater than 'lb' and less than 'ub'.
 • At 1.6 s, expected value to be greater than -0.5 and less than 0.5, actual value is 0.999573603041504.
 • At 4.7 s, expected value to be greater than -0.5 and less than 0.5, actual value is -0.999923257564099.
 • At 7.9 s, expected value to be greater than -0.5 and less than 0.5, actual value is 0.998941341839768.
 • At 10 s, expected value to be greater than -0.5 and less than 0.5, actual value is -0.544021110889368.

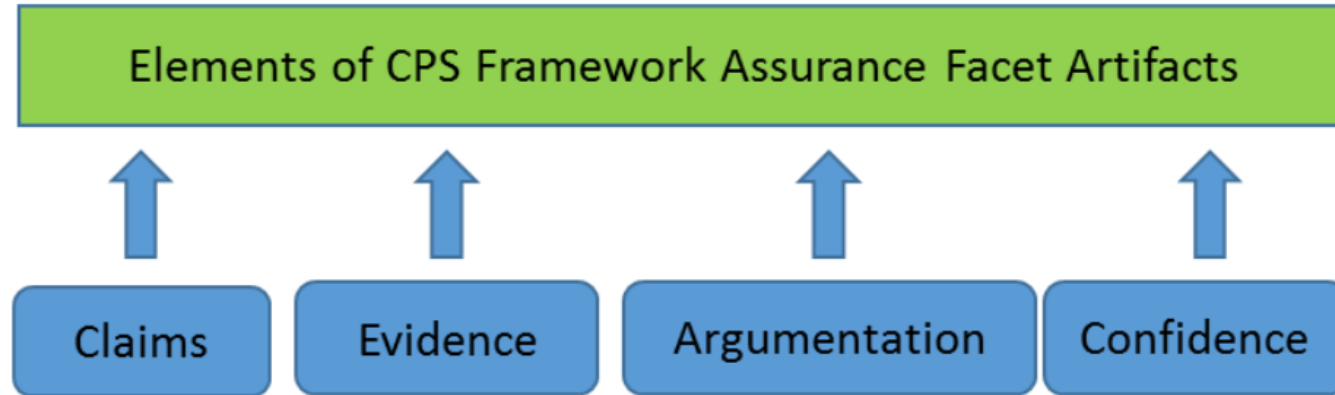
Hierarchical assessment tree



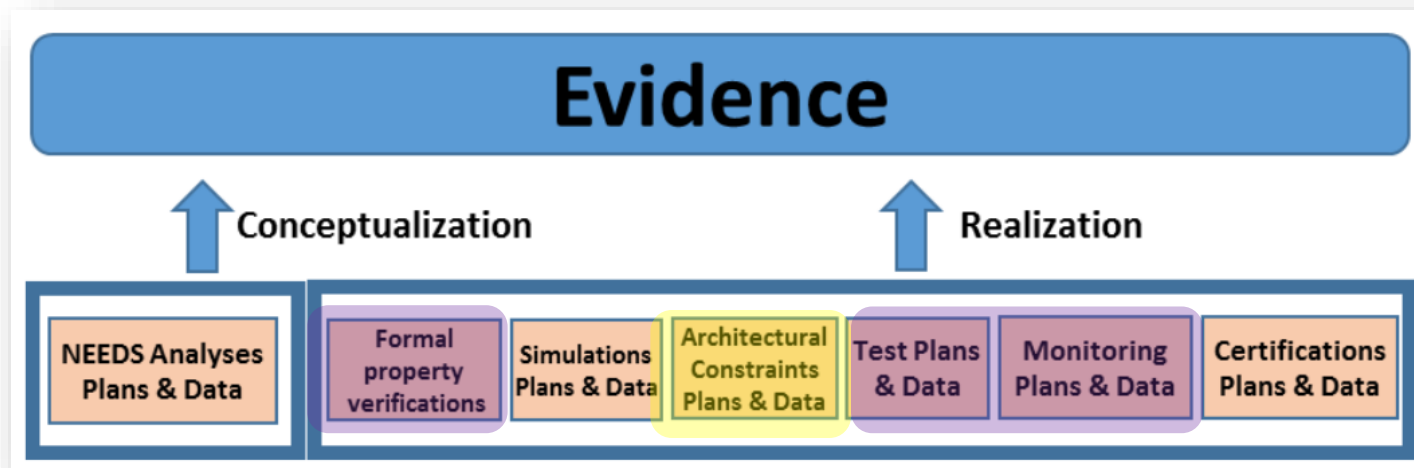
System Under Test



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

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>

