

# Integrating Viewpoints in the Development of CPS

**Martin Törngren**

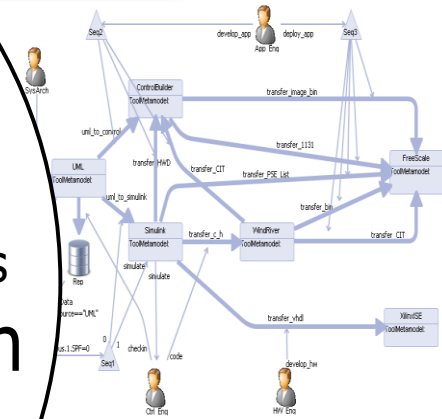
Professor in Embedded Control Systems, ICES director  
Division of Mechatronics, KTH – Royal Institute of Technology

Joint work with Matthias Biehl, Jad Elkhoury, Frederic Loiret, Magnus Persson and Ahsan Qamar, KTH, and Stavros Tripakis, Patricia Derler and Edward Lee, UC Berkeley



# Research areas in the group

ECS Group:  
Autonomy  
System Architecture  
Design and Optimization  
Innovative Product Concepts  
Model Based Design  
Multiview Modeling  
Tool Integration  
Safety



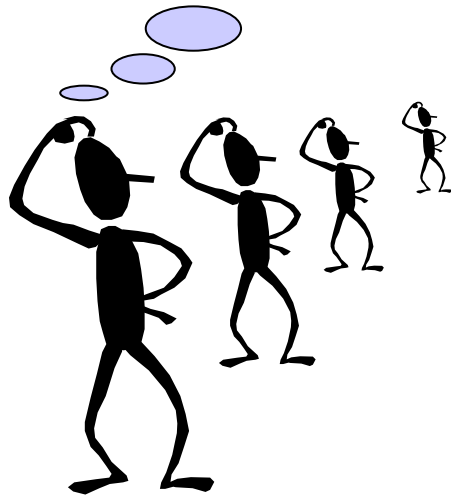
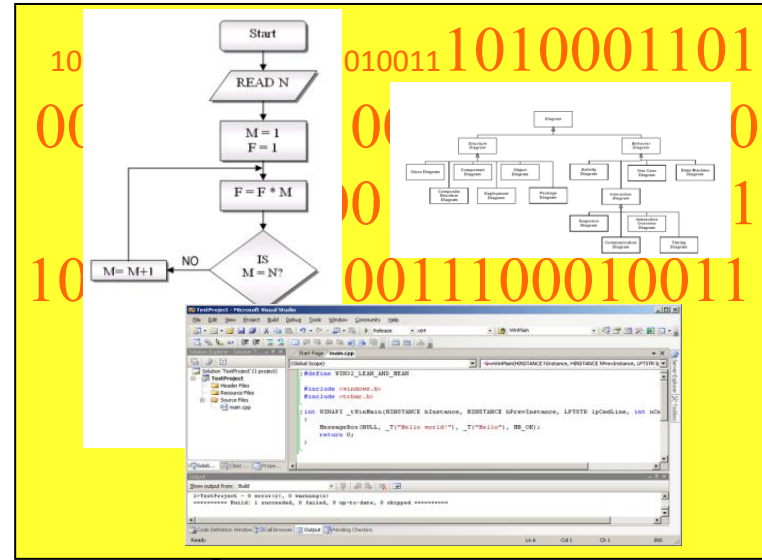
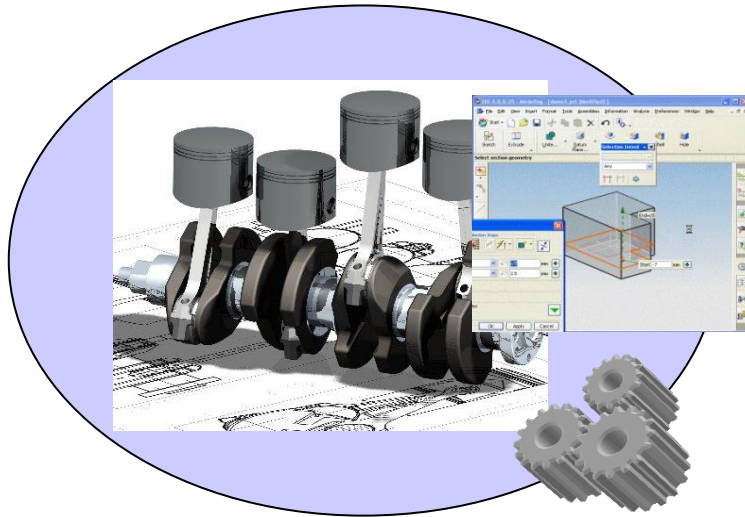
# CPS like demo



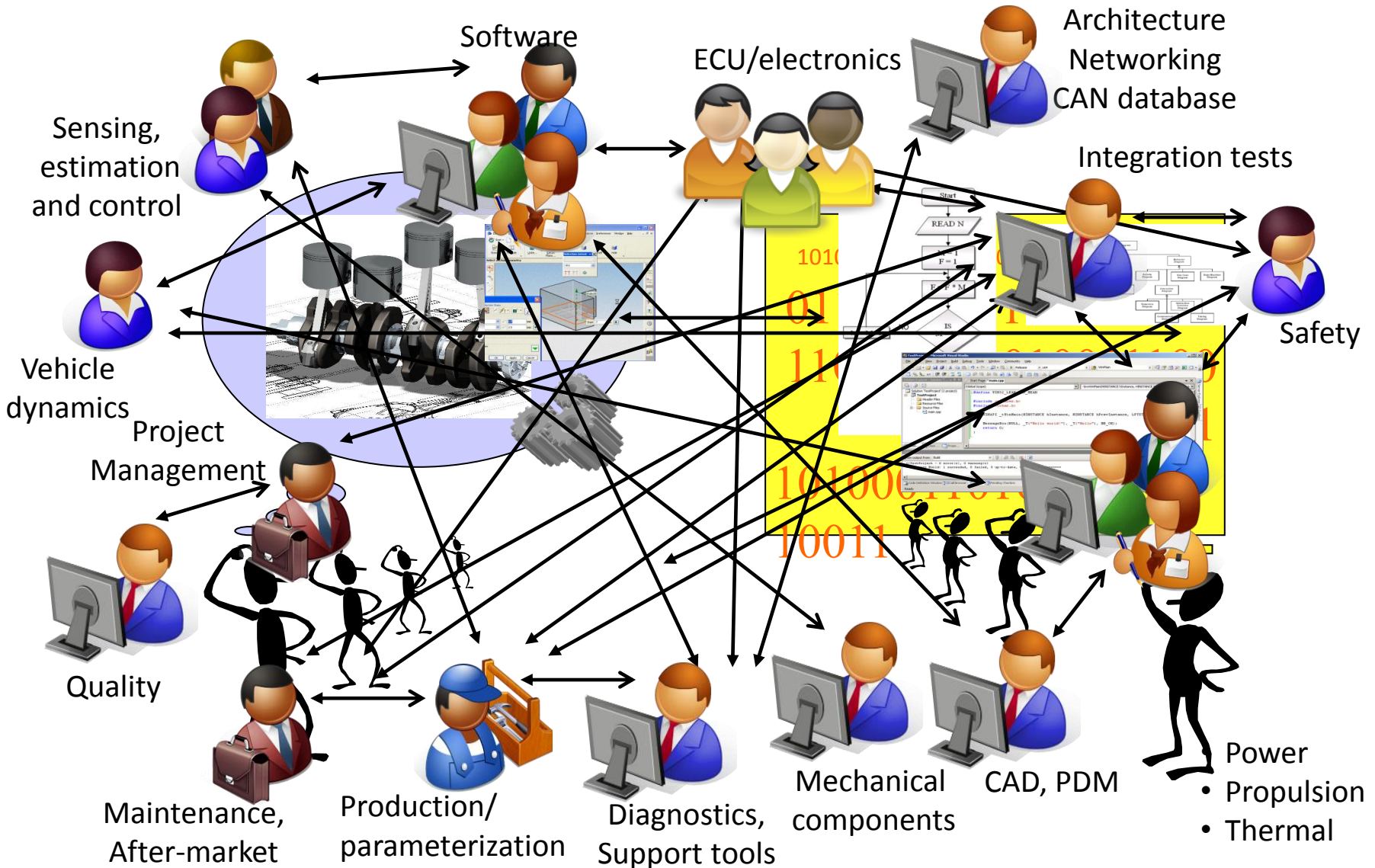
Manifested by, and courtesy of, the Berkeley juggling club



# Different viewpoints and concerns



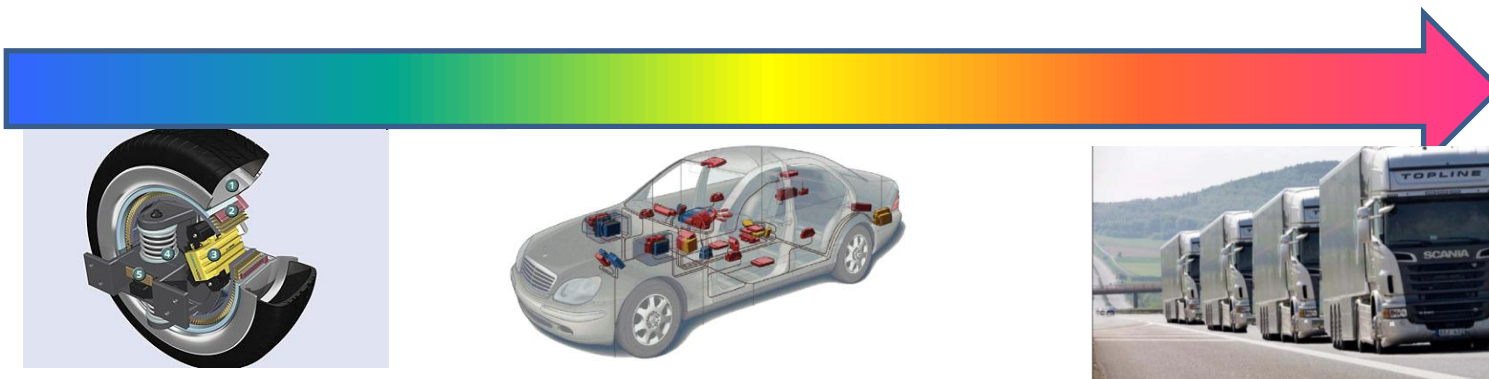
# Viewpoints and relationships



# Main messages

CPS come with growing functionality, extent and concerns

- More advanced engineering environments:
  - **viewpoints** and **relations** between them
- We need to engineer and deal with **viewpoint systems**
  - **Viewpoint contracts**
  - **Dependency modeling**
  - **Systematic and efficient tool integration**



Beyond  
traditional  
stakeholders

# Outline

- CPS Characterization
- Engineering Environments
  - Problem analysis
  - Multiview modeling
- Integrating viewpoints
  - Contracts, Dependency modeling, Tool integration
  - Discussion
- Wrap-up

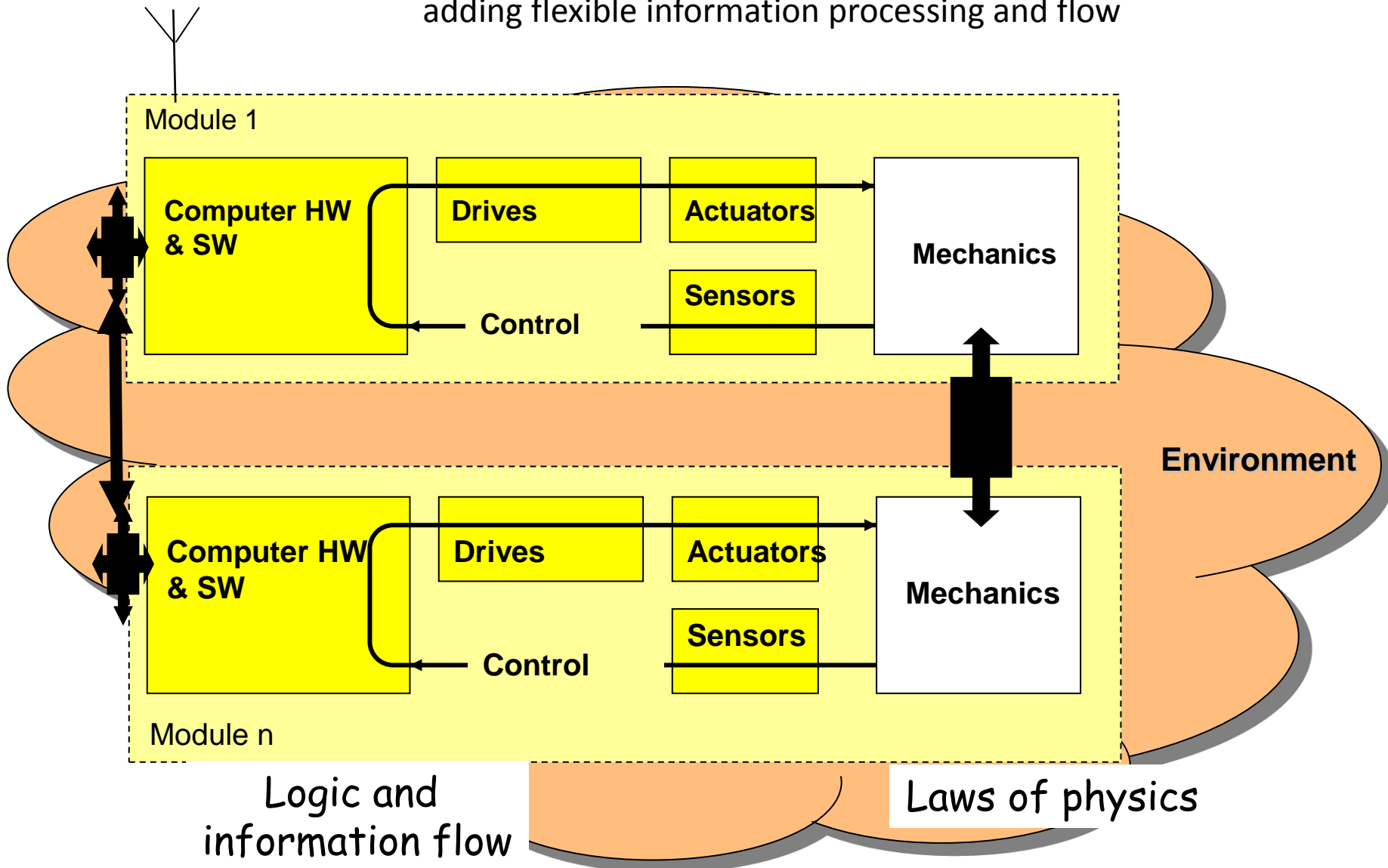
# CPS characteristics

- Stringent requirements
- Heterogeneity
- Extent and scale (compared to ES)
  - Autonomy
- Non-technical challenges
  - Socio-technical systems



# Mechanics vs. Mechatronics;

adding flexible information processing and flow



# "Purely" mechanical vehicle

	Sus/C	Brake	Steer	Wheel	Diff	Trans	Clutch	Eng	Driver
Susp				X					X
Brake				X					X
Steer				X					X
Wheel	X	X	X		X				
Diff				X		X			
Trans					X		X		
Clutch						X		X	
Eng							X		
Driver		X	X				X		

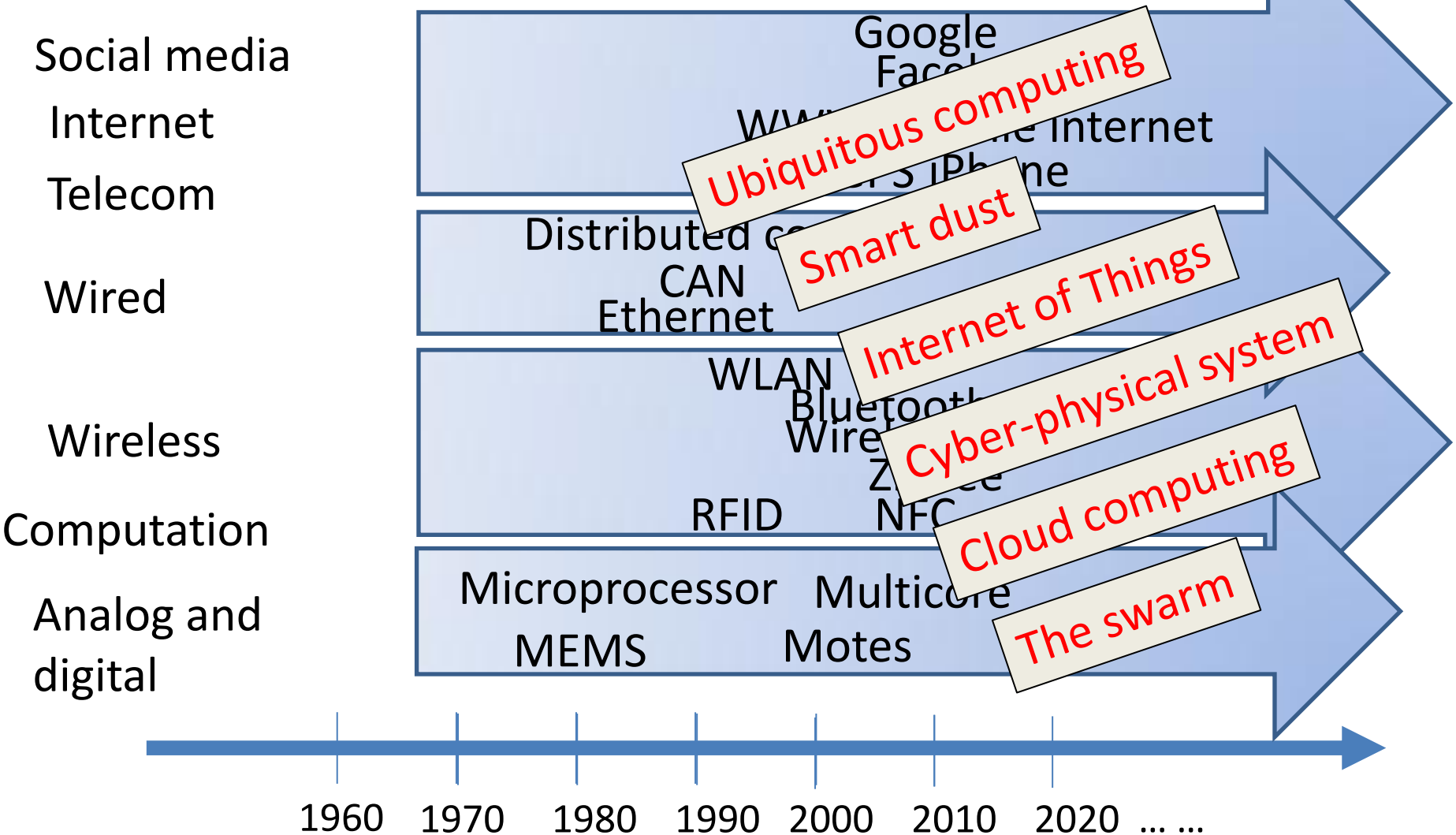
X - Mechanical relations

# Fully programmable vehicle!

	Sus/C	Brake	Steer	Wheel	Diff	Trans	Clutch	Eng	Driver
Susp		P	P	X+P	P	P	P	P	X+P
Brake	P		P	X+P	P	P	P	P	X+P
Steer	P	P		X+P	P	P	P	P	X+P
Wheel	X	X	X+P		X				
Diff	P	P	P	X+P		X+P	P	P	
Trans	P	P	P	P	X+P		X+P	P	P
Clutch		P	P		P	X+P		X+P	P
Eng	P	P	P	P	P	P	X+P		P
Driver	P	X+P	X+P		P	P	X+P	P	

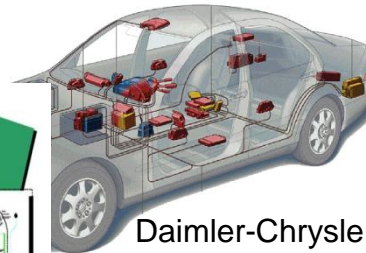
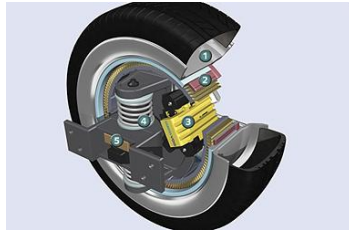
P - Programmable relations  
X - Possible change

# Technology evolution



# Autonomous vehicles

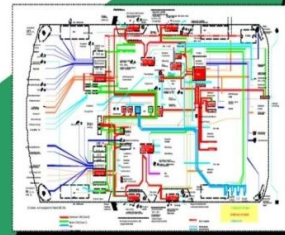
E-Corner, Siemens



Daimler-Chrysler

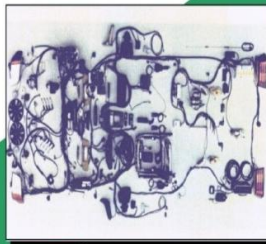


Platooning competition (Scania and KTH)



Wiring Harness 1999 S-Class

- 3 Data Bus Systems
- 60 ECUs
- 110 Electric Motors



Wiring Harness 1990 S-Class

- Length ~ 3 km
- Wires ~ 1900
- Contact Points ~ 3800
- Weight ~ 39 kg

Local, global communication:  
- V2V, V2I, Internet, positioning  
New Opportunities and  
Challenges!

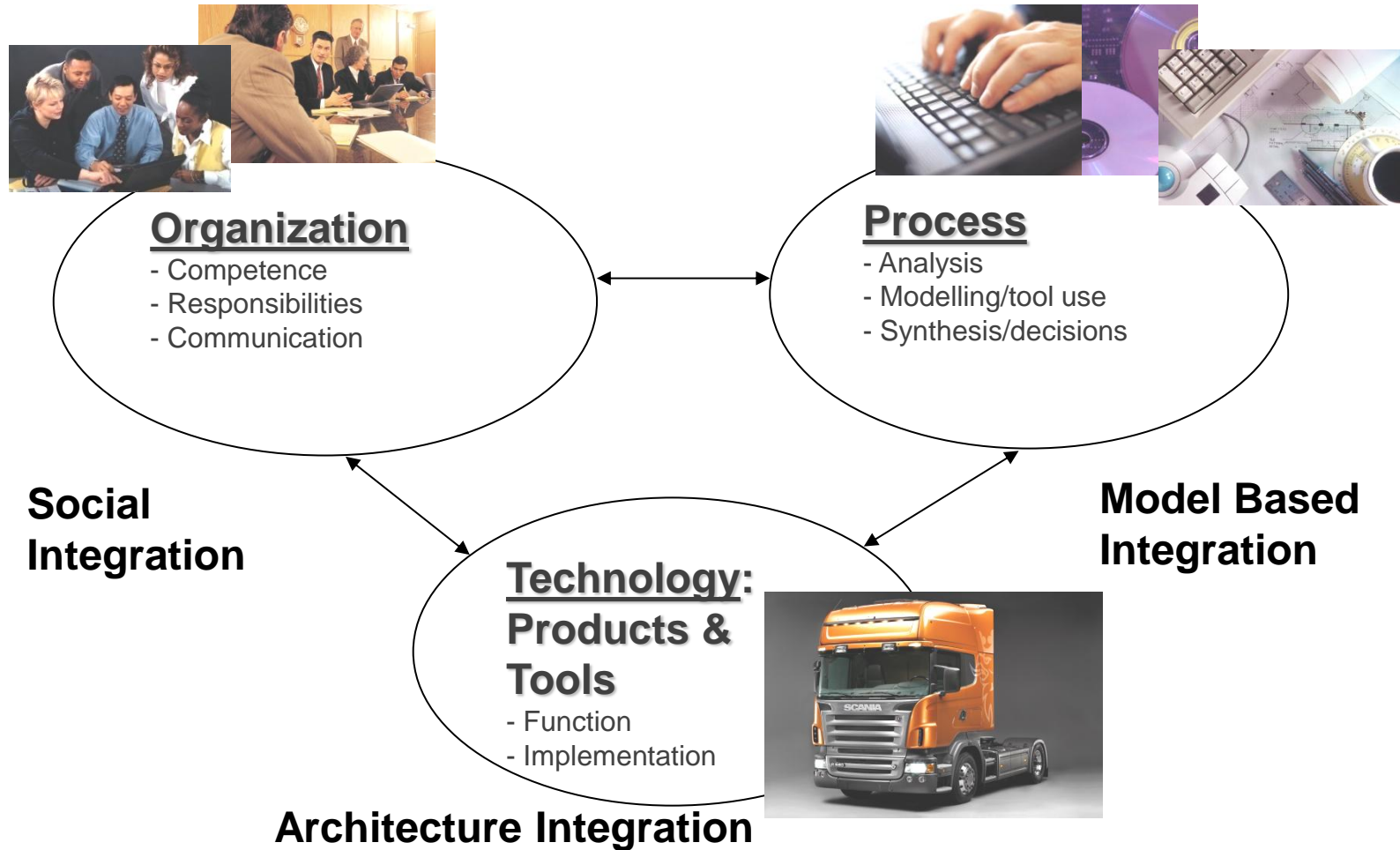
Beyond  
traditional  
stakeholders

Scaling problems

Components – Devices – Subsystems – Systems – SoS - Society



# People – Processes - Technology



# Complexity

Here taken in relation to human abilities to deal with different aspects and their relations

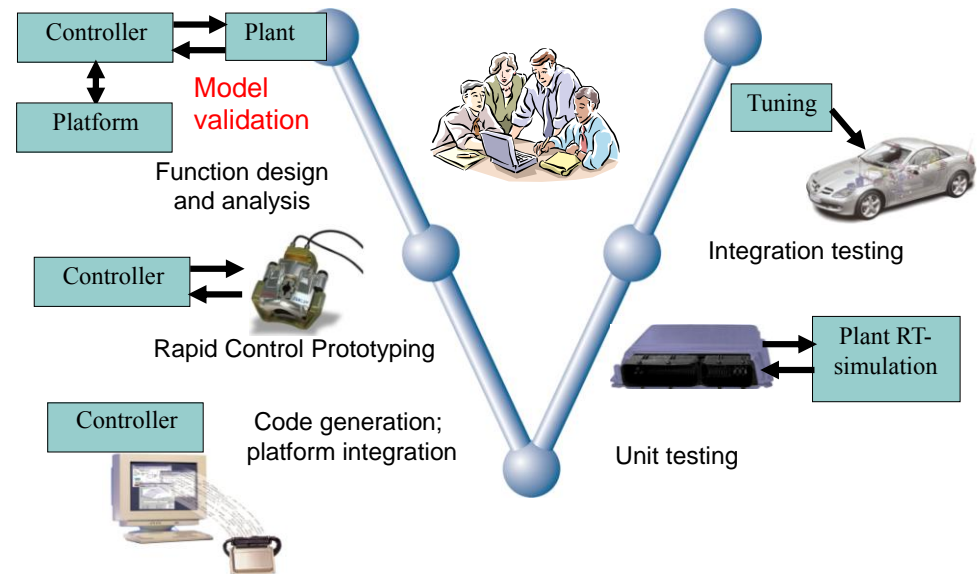
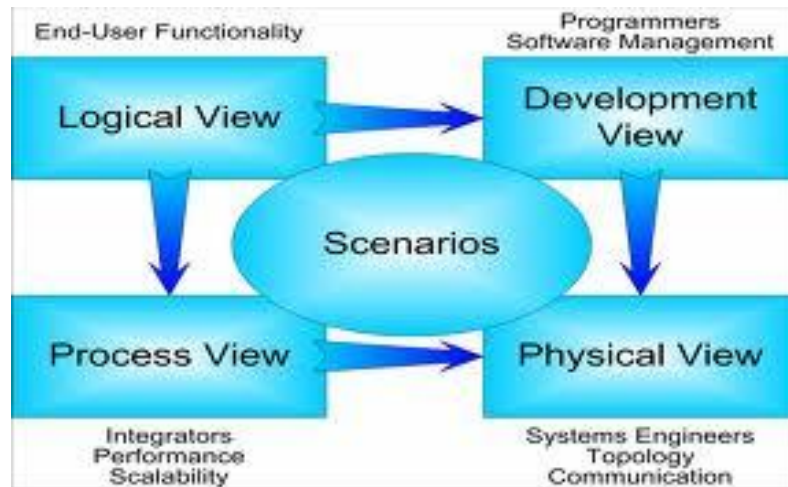
- Requirements, components and networks
- Behaviors and interactions
  - Digital state space
  - Hybrid systems
  - Product variants
- Life-cycle usages
- Potential faults, errors and failures

# How do we manage complexity when developing technical systems?

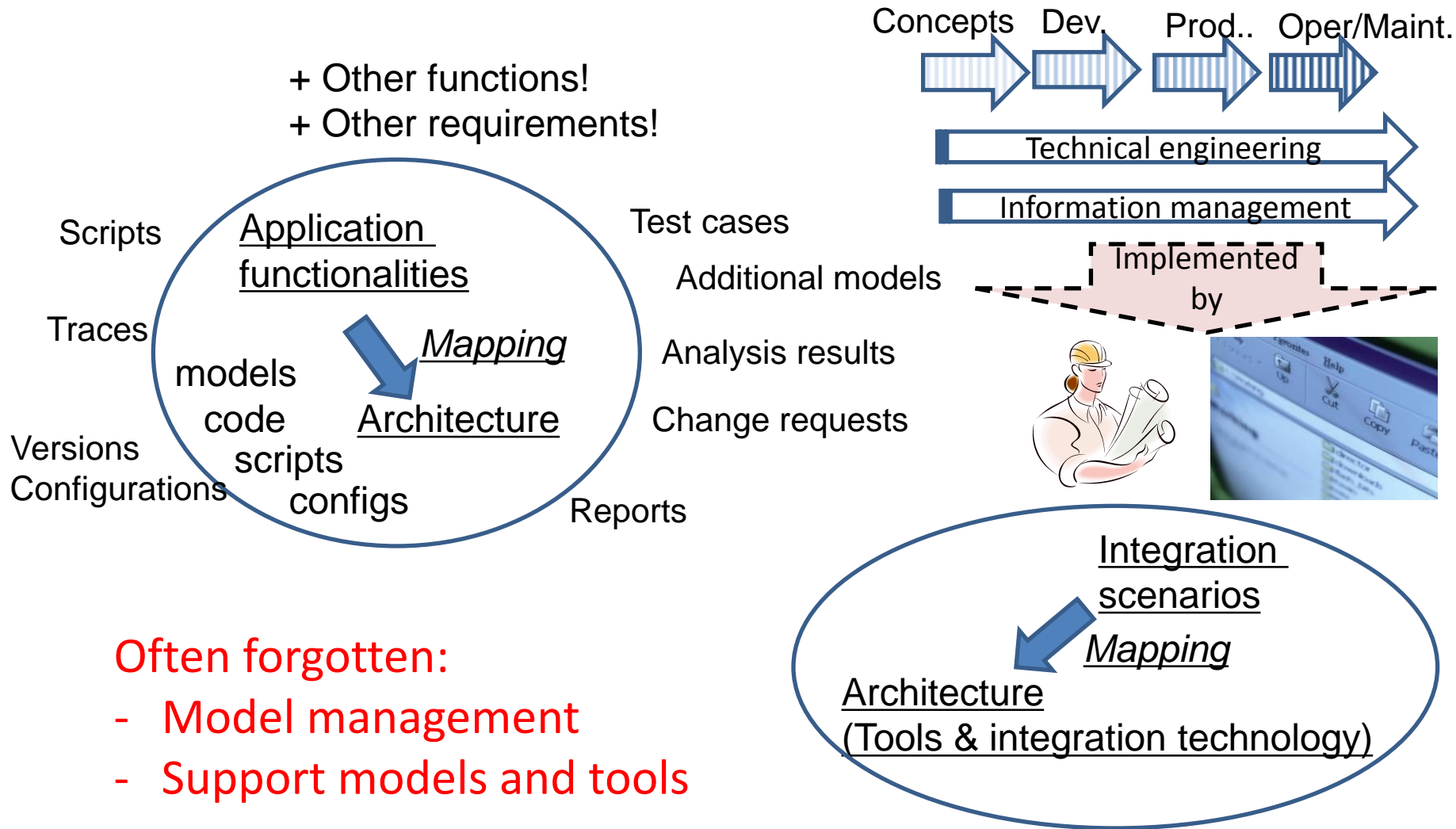
- Divide and conquer
  - Applied to products and systems (supporting artefacts, organizations)
- Abstraction
  - Modeling, at the "right" level of abstraction using the "right" formalism
- Ensuring integration / composability
  - Interfaces, interrelationships, assumptions
- By use of automation ("tools")

# Abstraction

- Modeling and simulation
- Multiple abstraction layers; Multiple views
- Platform abstractions: API's, services



# Aspects of product & support tools



Often forgotten:

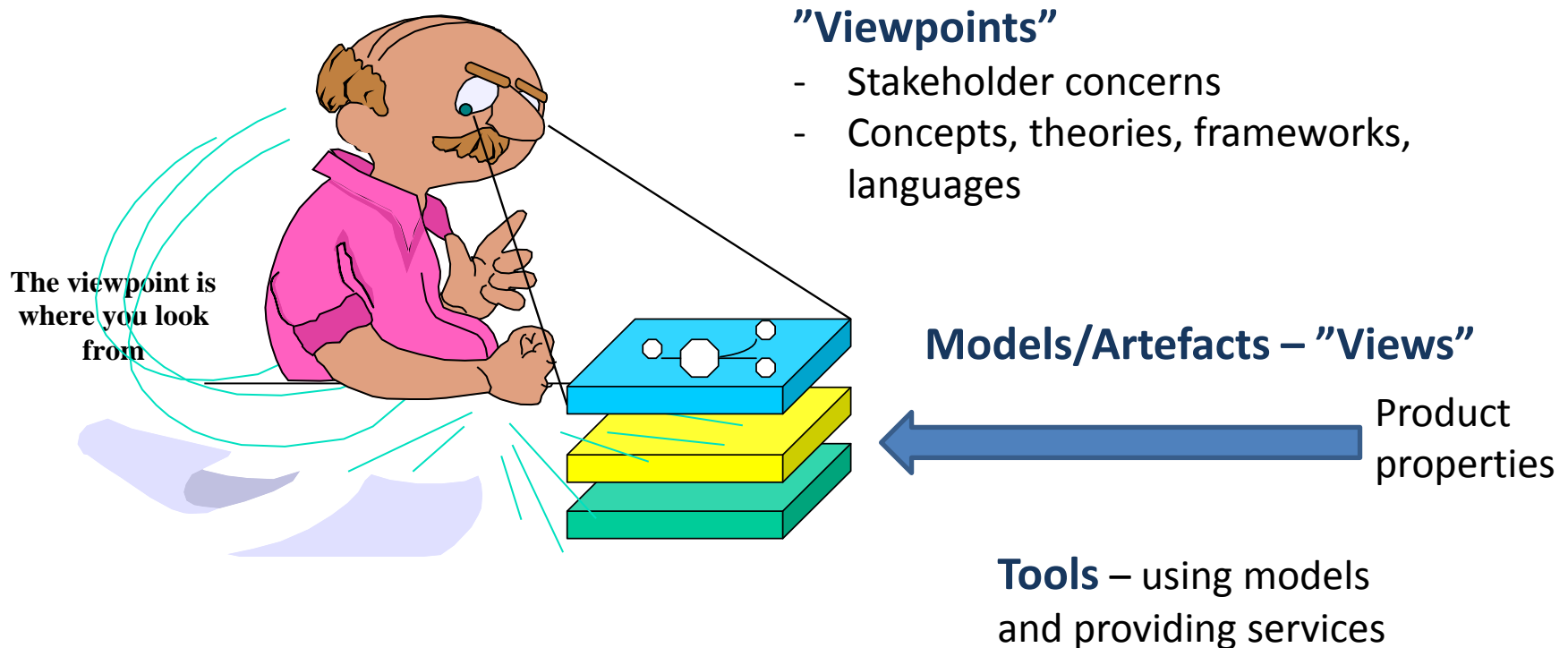
- Model management
- Support models and tools



# Outline

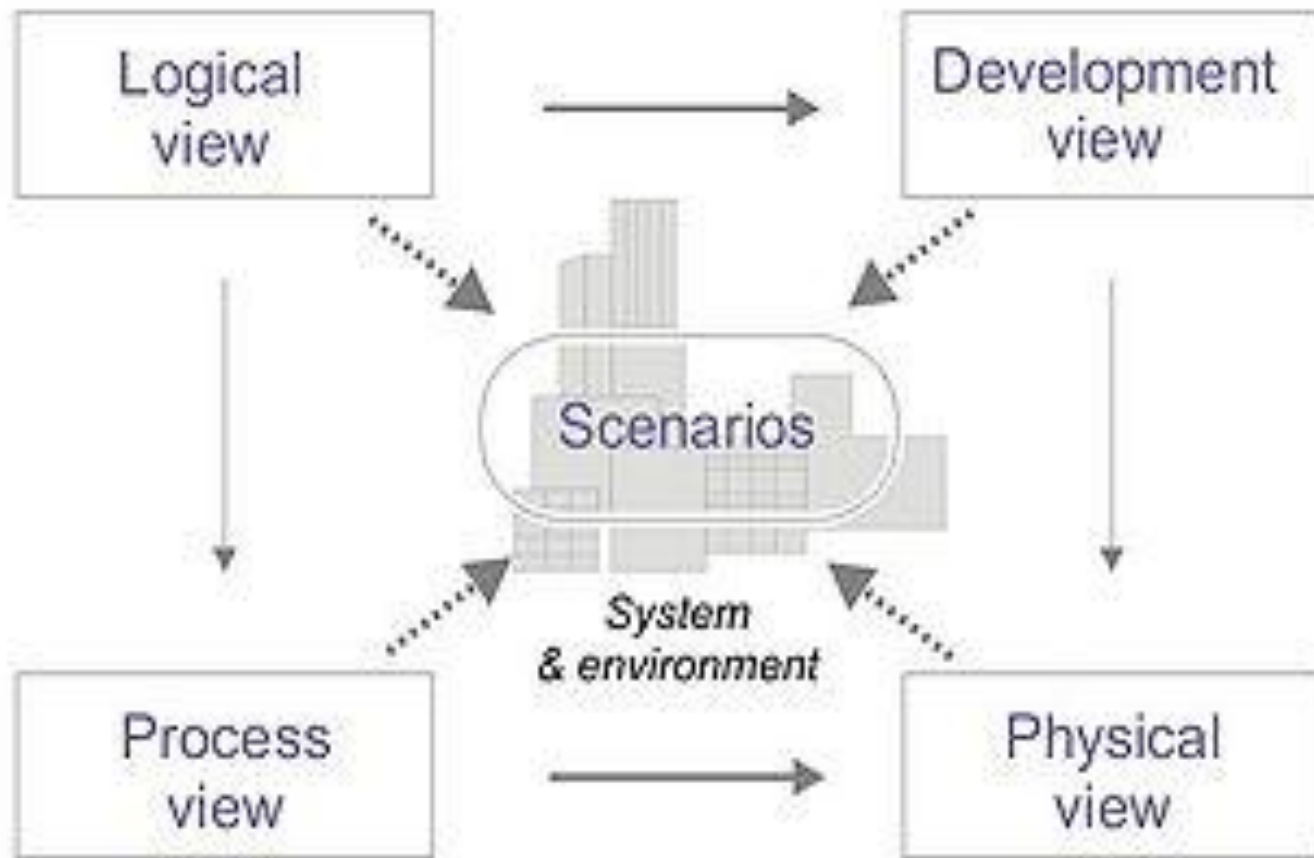
- CPS Characterization
- **Engineering Environments**
  - Problem analysis
  - Multiview modeling
- Integrating viewpoints
  - Contracts, Dependency modeling, Tool integration
  - Discussion
- Wrap-up

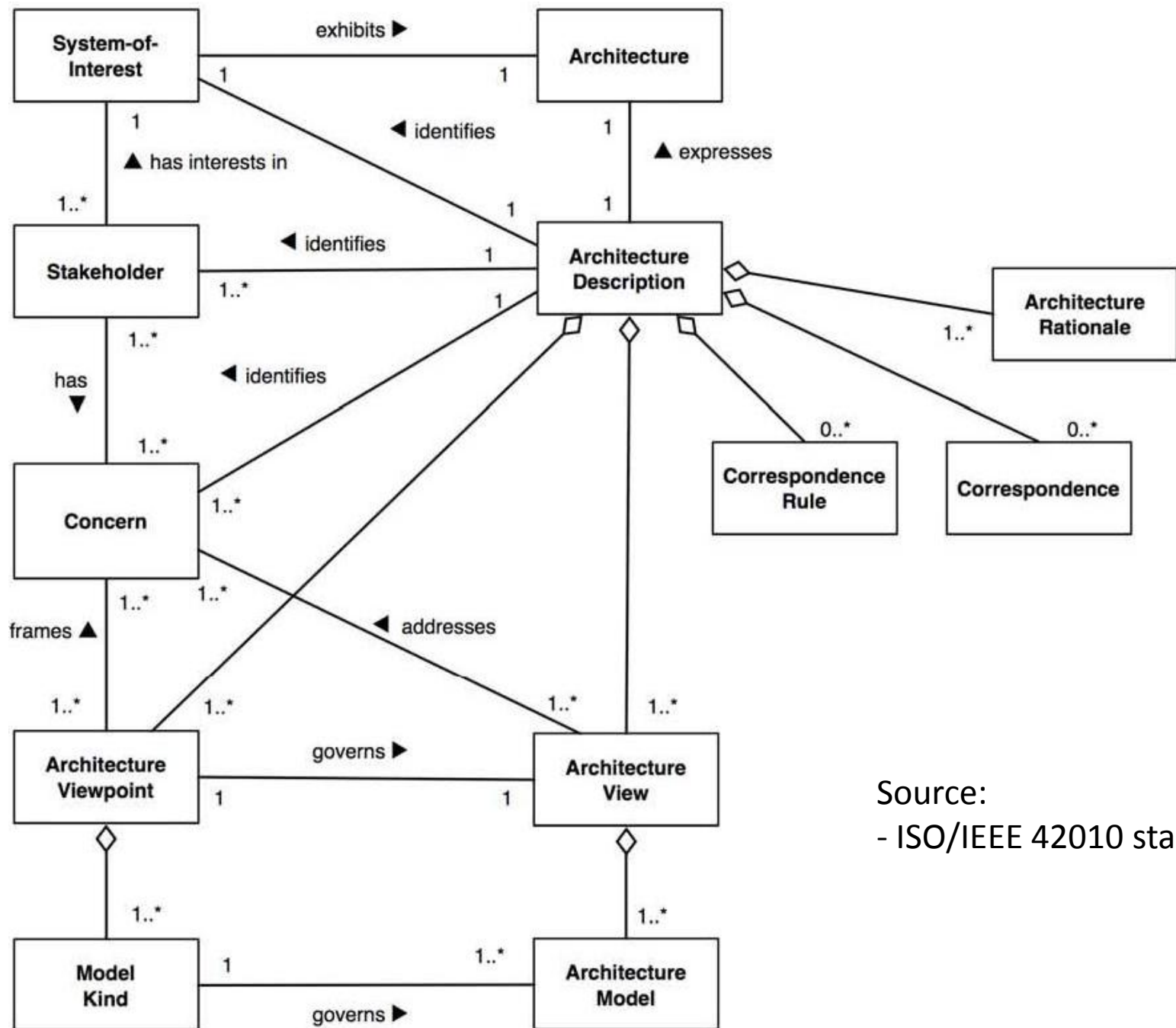
# Viewpoints and views



Terminology partly from the ISO/IEEE 42010 standard: Systems and software engineering — Architecture description

# One example multi-view reference model: 4+1 view model



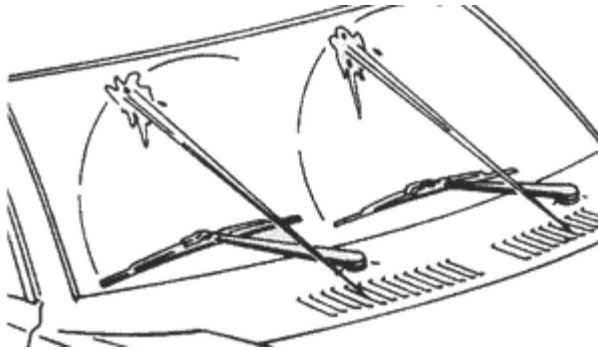


Source:  
- ISO/IEEE 42010 standard

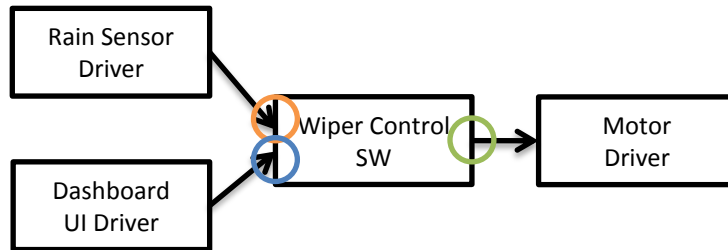
# Multiple views - Example

## Allocation

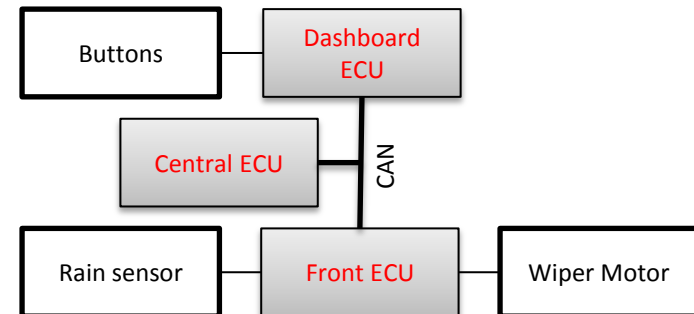
Rain Sensor Driver - F ECU  
 Dashboard UI Driver - DB ECU  
 Wiper Control SW - C ECU  
 Motor Driver - DB ECU



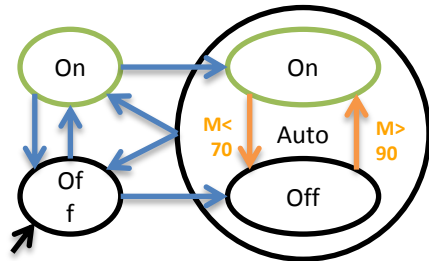
## Software structure



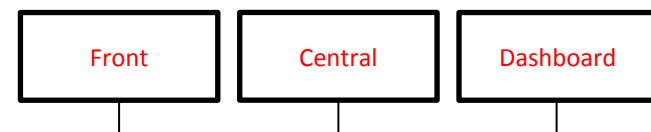
## Hardware structure



## Software mode description



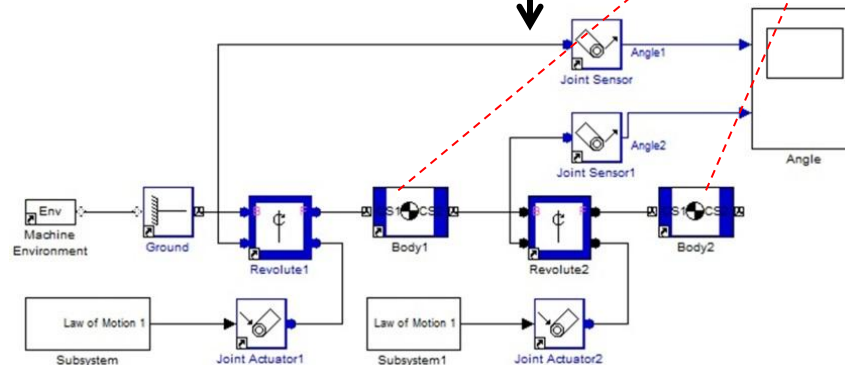
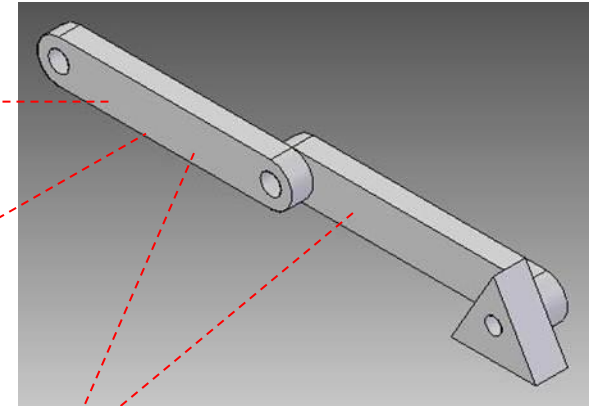
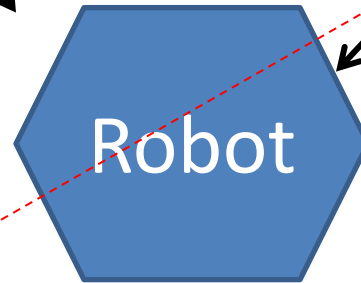
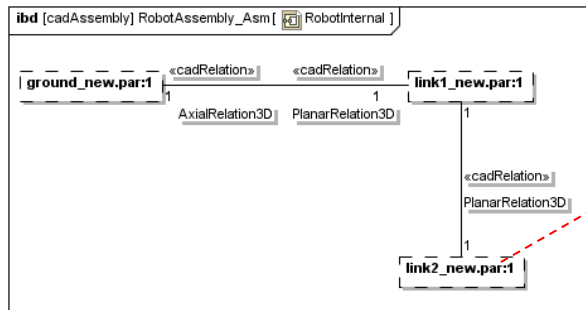
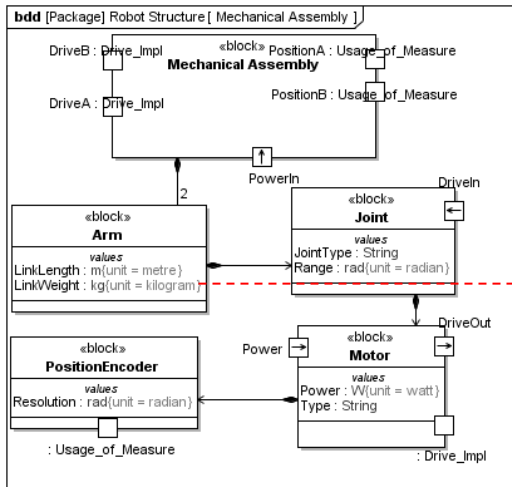
## Network diagram



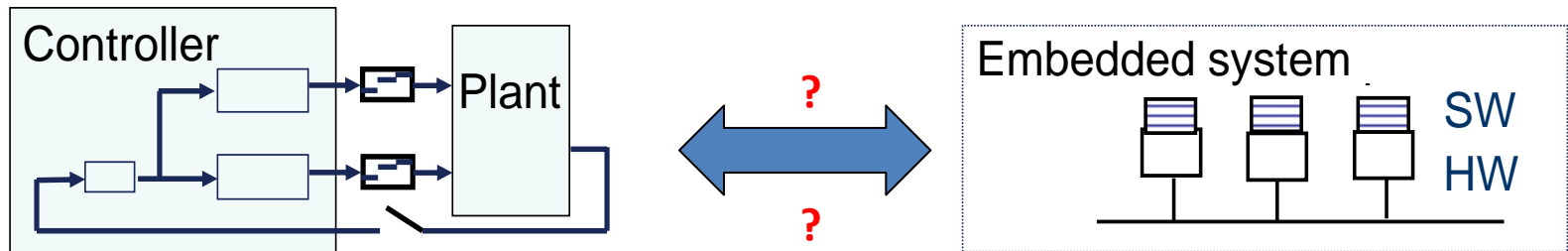
Example models by Magnus Persson



# Multiple views - Example



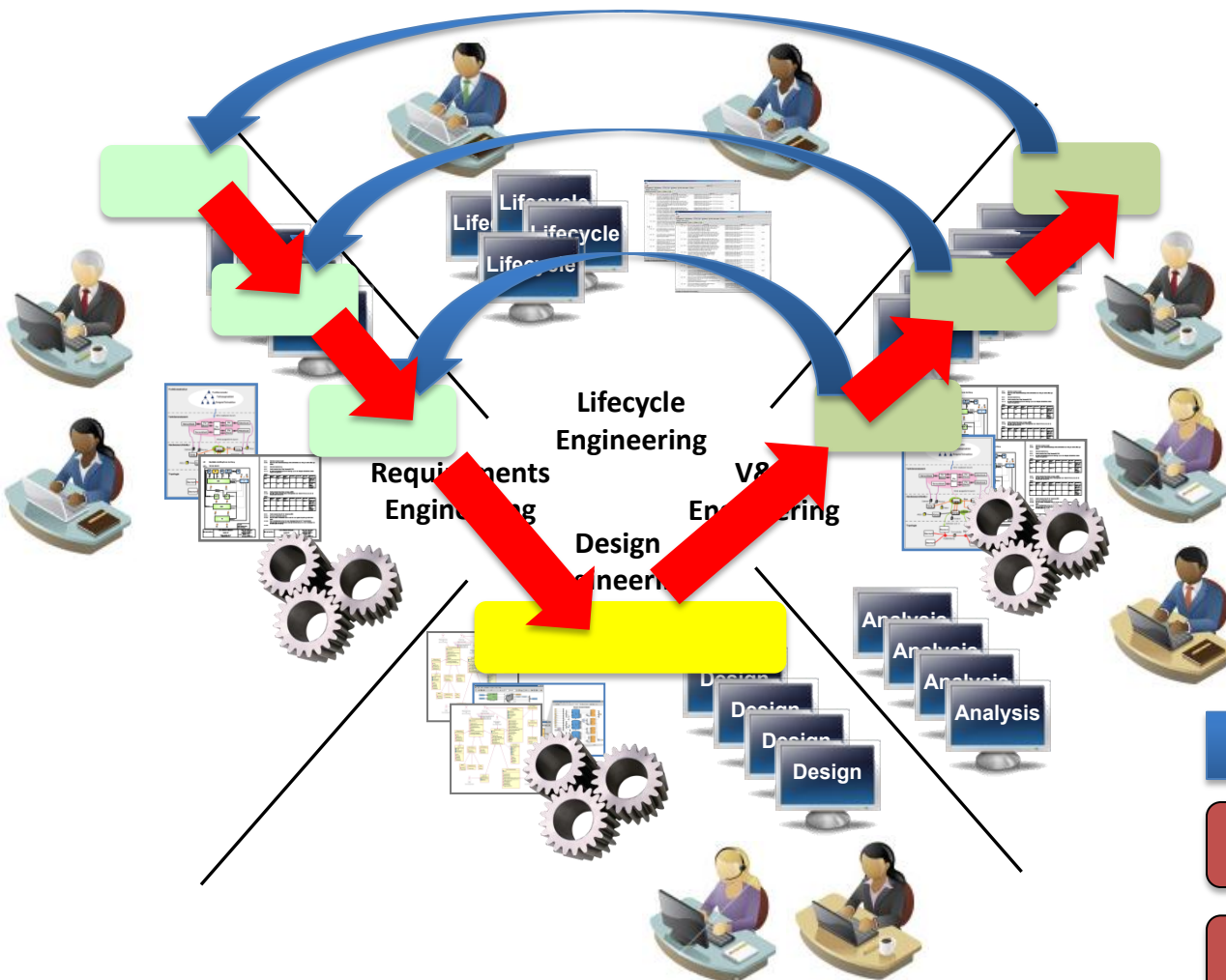
# Controller and Embedded system – gaps



- Little emphasis on effects due to embedded systems realization
- Delays, quantization, partial failures, resource sharing
- Little emphasis on control specific requirements
- Assumed behavior, error handling

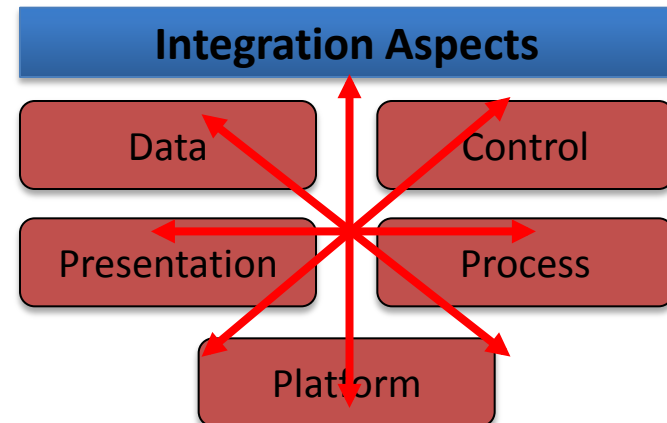
Concepts, theory and tools mismatches  
- e.g. Discrete-time control theory vs.  
Fixed priority scheduling /rate monotonic

# Tool Integration Challenge



## Multiple

- Concerns & Domains
- Stakeholders
- Isolated Tools & Models
- Processes and Tasks
- Integration Aspects
- Dependencies



Lack of methodologies for tool integration  
Lack of mature standards

# iFEST – Desired tool integration capabilities

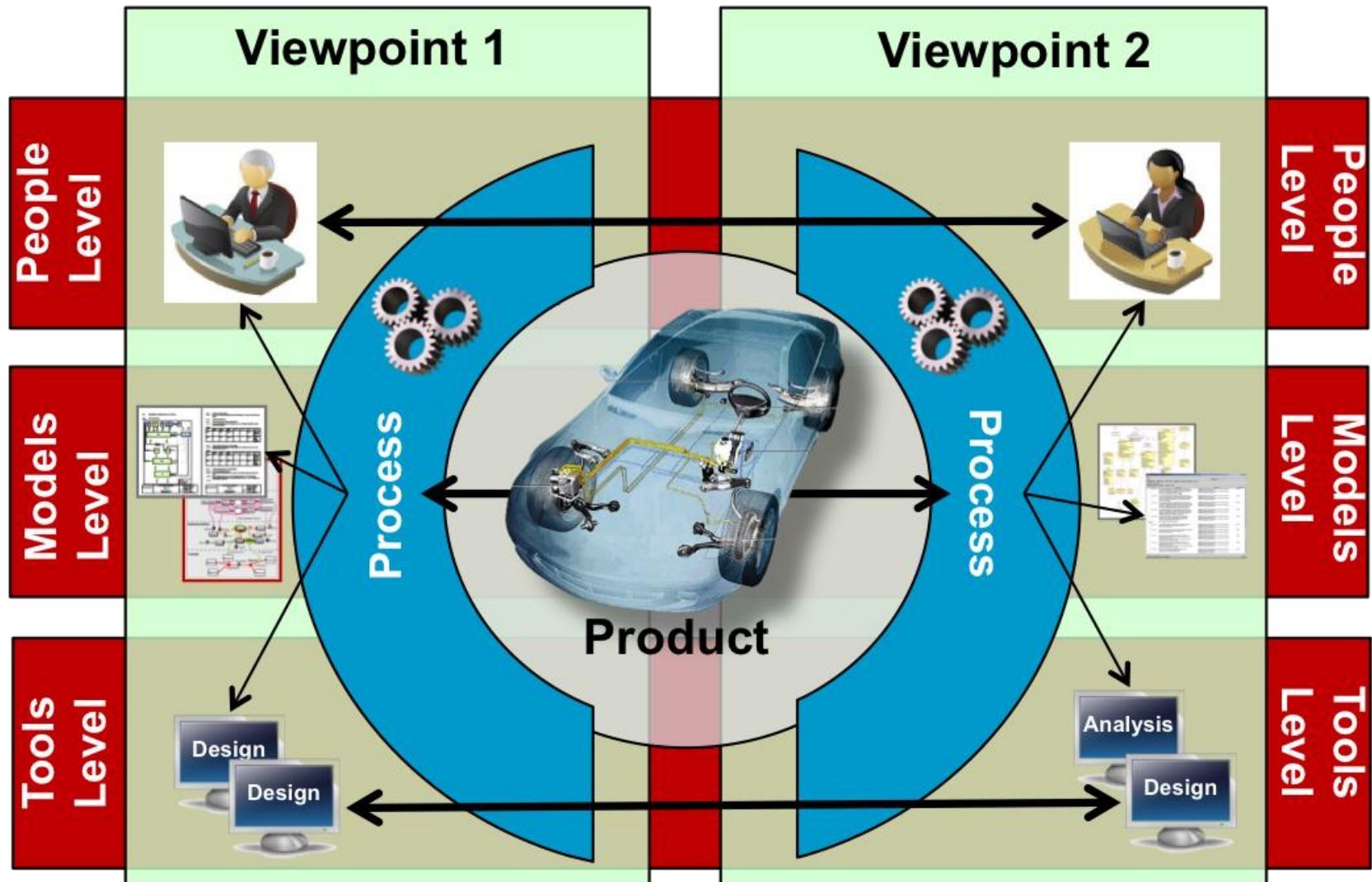
- A uniform approach to tool integration
  - Dealing with complexity
  - Tailorability, Evolvability and Scalability
- Lower threshold to integrate tools
  - Ability to define information and functionalities
  - Access to information and functionalities of tools
  - Manage information
  - Data, control, presentation, process
  - Support relevant standards

# Outline

- CPS Characterization
- Engineering Environments
  - Problem analysis
  - Multiview modeling
- **Integrating viewpoints**
  - Contracts, Dependency modeling, Tool integration
  - Discussion
- Wrap-up



# Towards integrating viewpoints



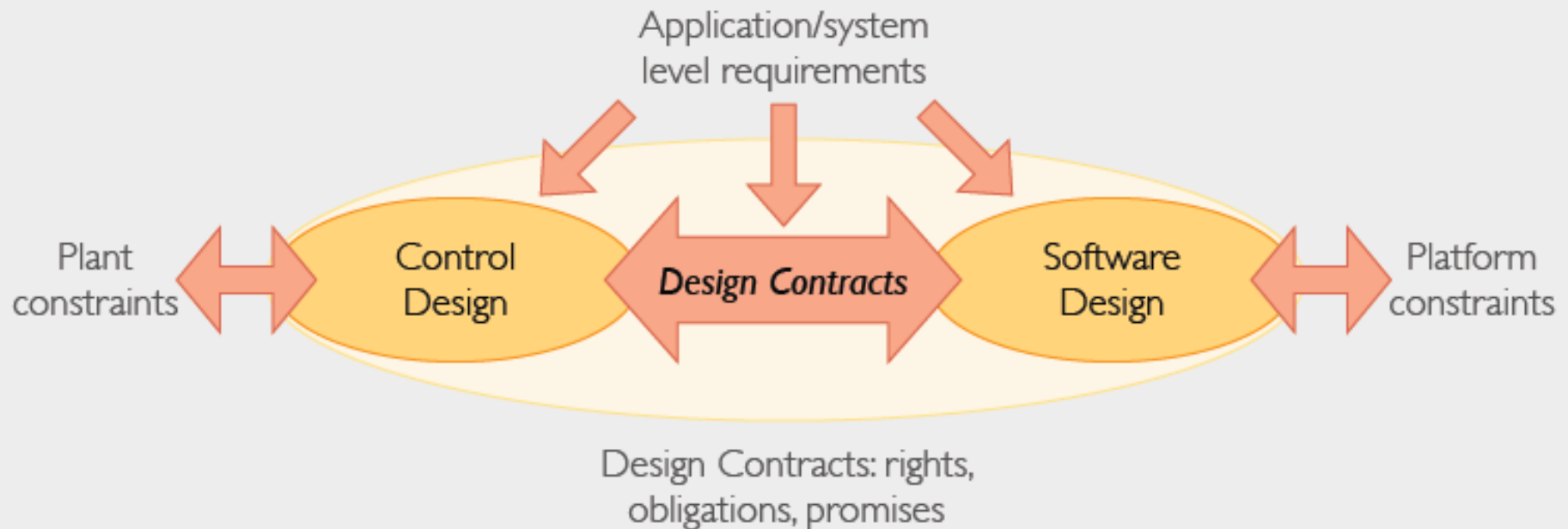
# Techniques

- Viewpoint contracts
  - “people level”
  - Establishing shared concepts, assumptions and constraints
- Dependency models
  - Explicitly captures dependencies among properties of models
- Tool integration models
  - Explicitly describes tools and their interactions

# Viewpoint contracts

- Bridging conceptual and communication gaps
- Capturing
  - Common concepts shared among two or more viewpoints – Minimalistic
  - Assumptions
  - Constraints
- Currently dealt with informally

# Example: Control-Embedded SW contracts



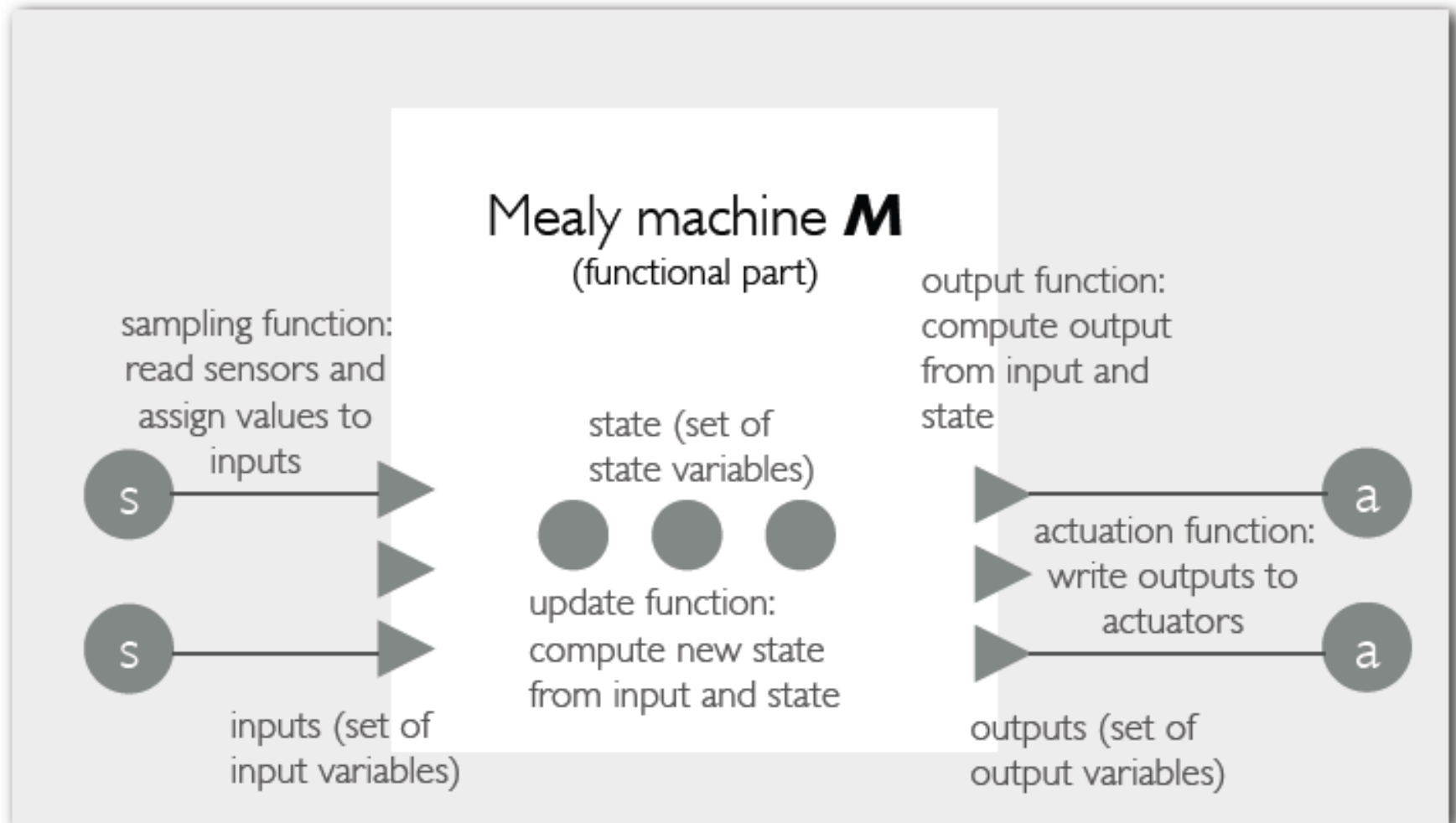
Courtesy of P. Derler, Design contracts presentation at CPS conference 2013

# Control vs. Embedded software

<i>Concepts/Domain</i>	<i>Control</i>	<i>Embedded software</i>
Metrics, Constraints	Robustness, noise sensitivity, bandwidth, overshoot, settling time	Utilization, response time, memory footprint, WCET, slack, power consumption
Design parameters	Choice of strategy (PID, optimal, adaptive, etc.), noise/robustness trade-off	Task partitioning, scheduling, inter-process communication
Formalisms, Theory	Continuous time (ODEs), discrete time, sampled data control theory	C code, synchronous languages, scheduling theory, model-checking, task models, UML/SysML

Courtesy of P. Derler, Design contracts presentation at CPS conference 2013

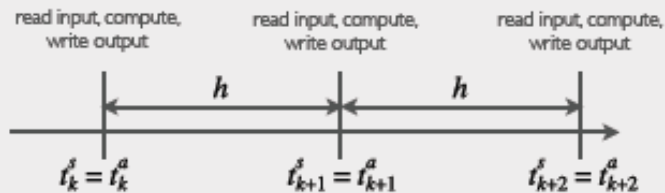
# Example: Control-Embedded SW contracts



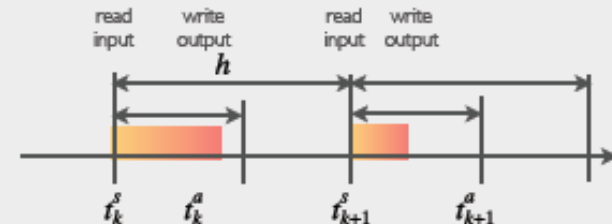
Courtesy of P. Derler, Design contracts presentation at CPS conference 2013

# Example contracts - Control-Embedded SW with timing constraints

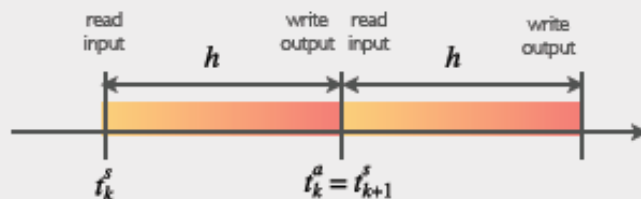
ZET Zero Execution Time  
single periodic



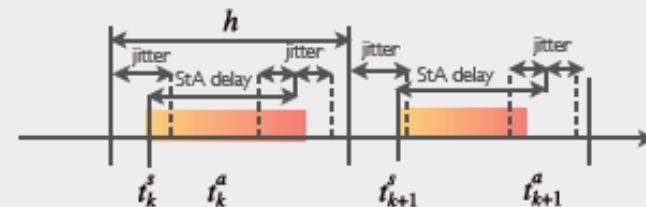
BET & DET Bounded Execution Time, Deadline Execution Time



LET Logical Execution Time



TOL Time Tolerances





# Example contracts continued;

## Control-Embedded SW with timing constraints

Agreement and obligations regarding functionalities and timing properties

- SW engineers: execute functions; meet timing requirements
- Control engineers: ensure correct closed-loop behavior

Example contracts:

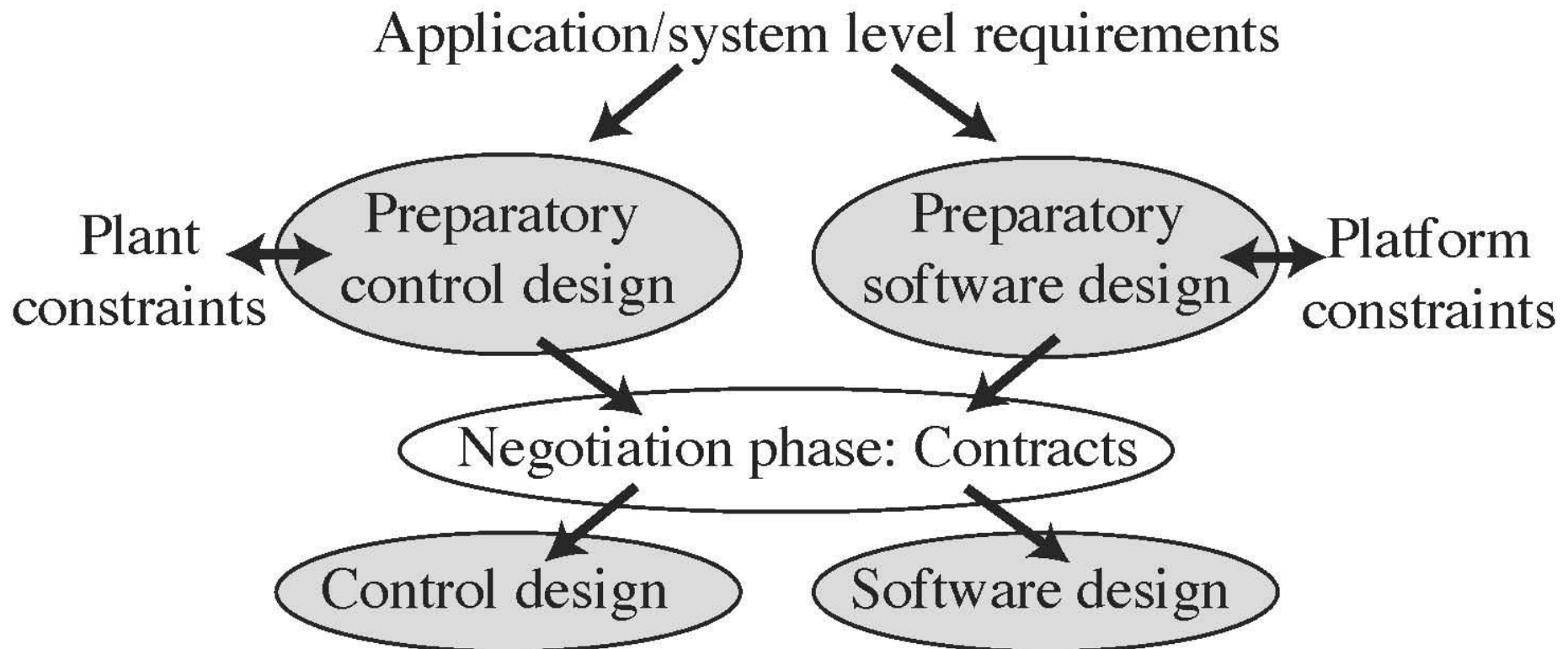
- 'ZET' ~ the synchronous approach
- 'LET' ~ the PLC / Giotto approach
- 'BET' ~ interpretation of FPS
- 'DET' ~ deadline monotonic scheduling
- 'TOL' ~ Tolerances on time variations

Basis for discussions and agreements

Baseline for implementation

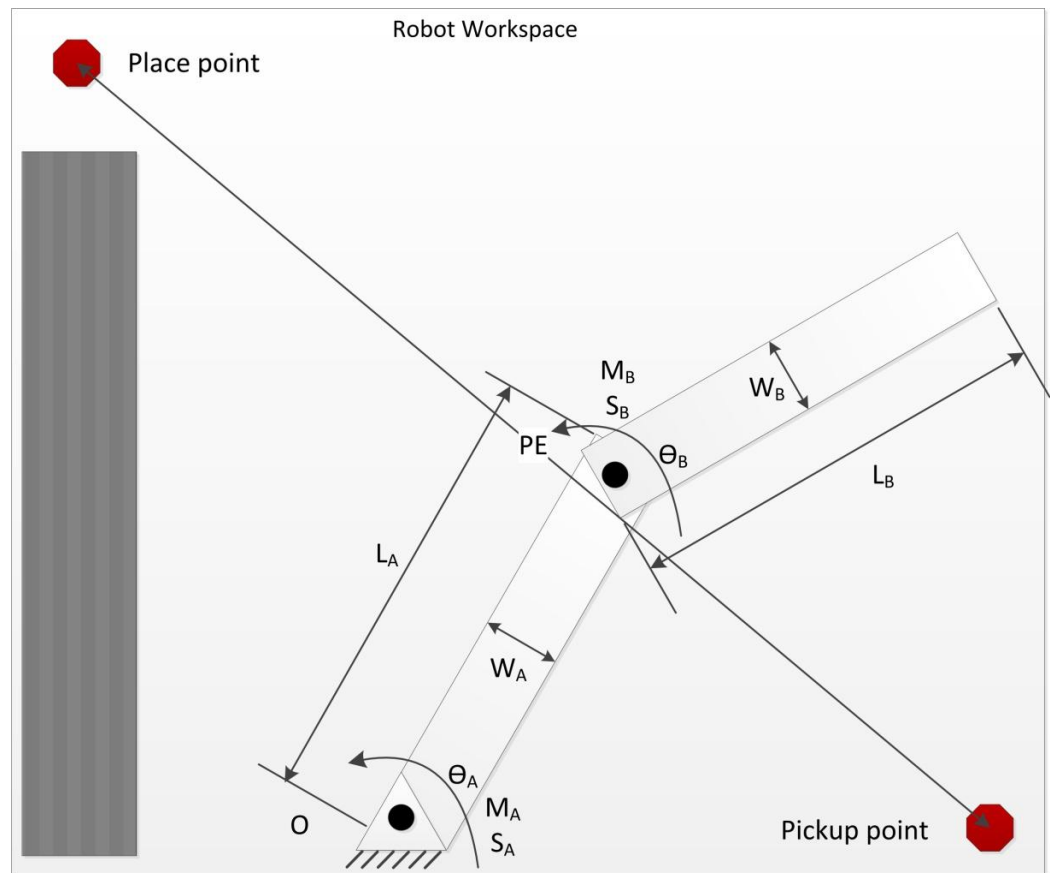
Support for modeling and simulation

# Support for Contract based design- methodology outline

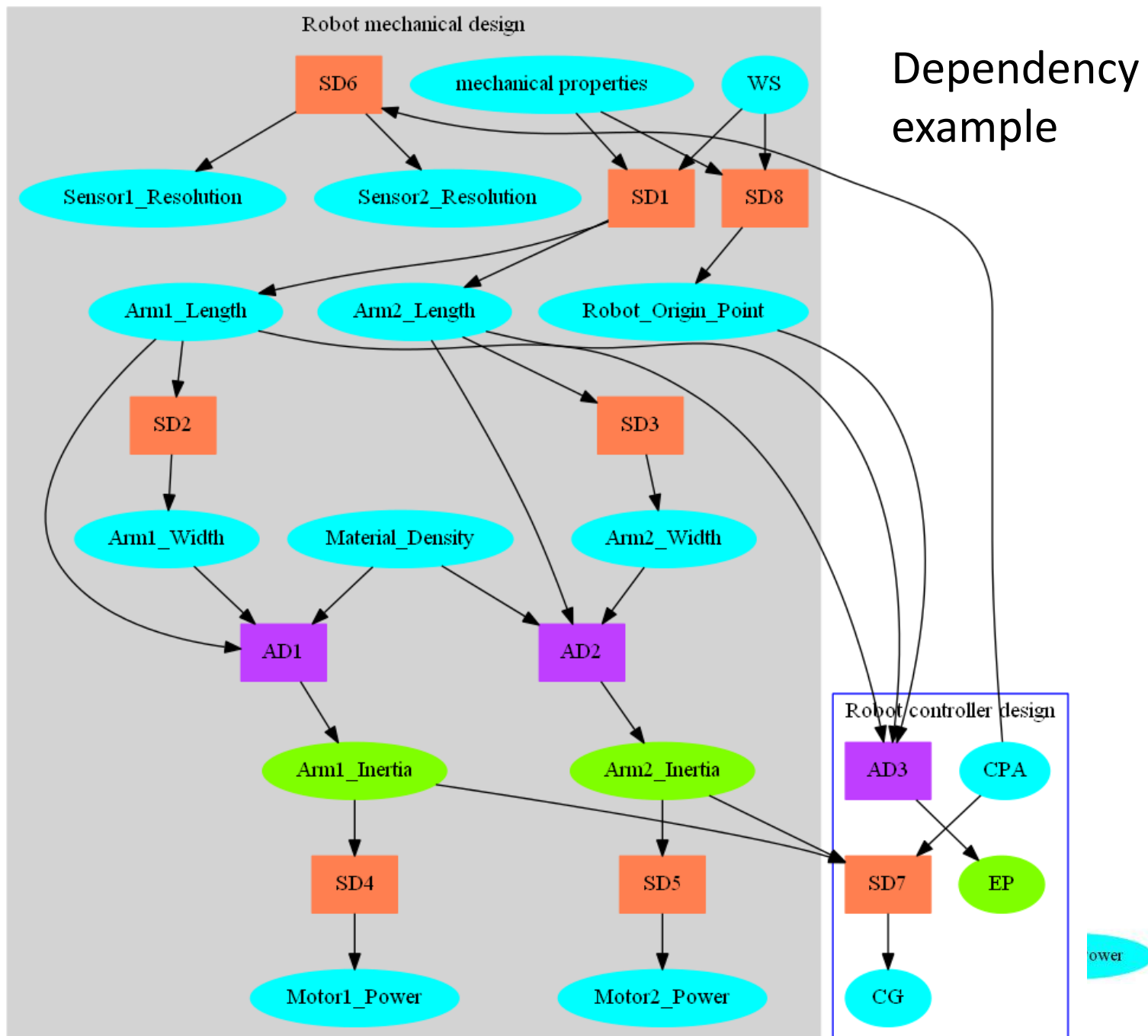


# Two Degree-Of-Freedom Robot

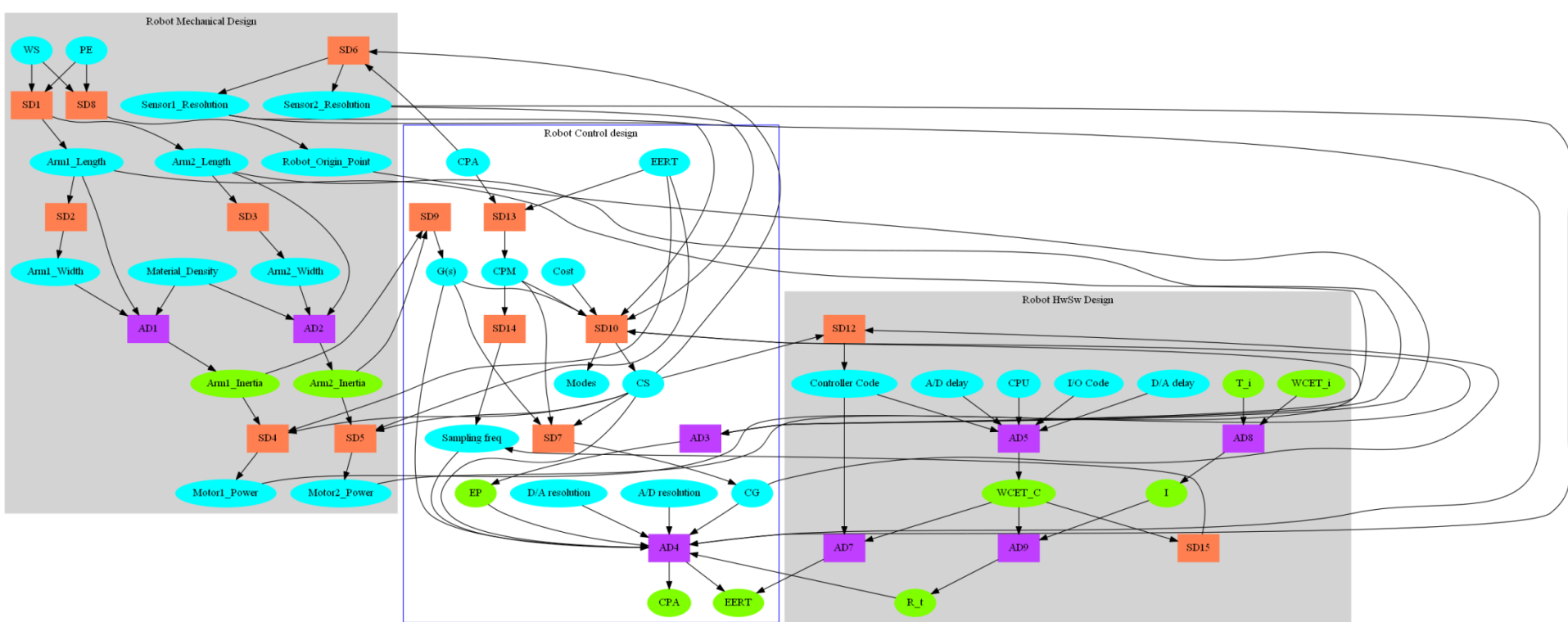
- Control the position of the robot with accuracy (CPA) and avoid obstacle within the specified workspace (WS)
- Two design domains
  - Mechanical design
  - Control design
  - Hw/Sw design
- Design variables
  - $L_A$ ,  $L_B$
  - $W_A$ ,  $W_B$
  - $\theta_A$ ,  $\theta_B$
  - $\varrho$
  - $S_A$ ,  $S_B$
  - $M_A$ ,  $M_B$
  - Origin 'O'
  - PE

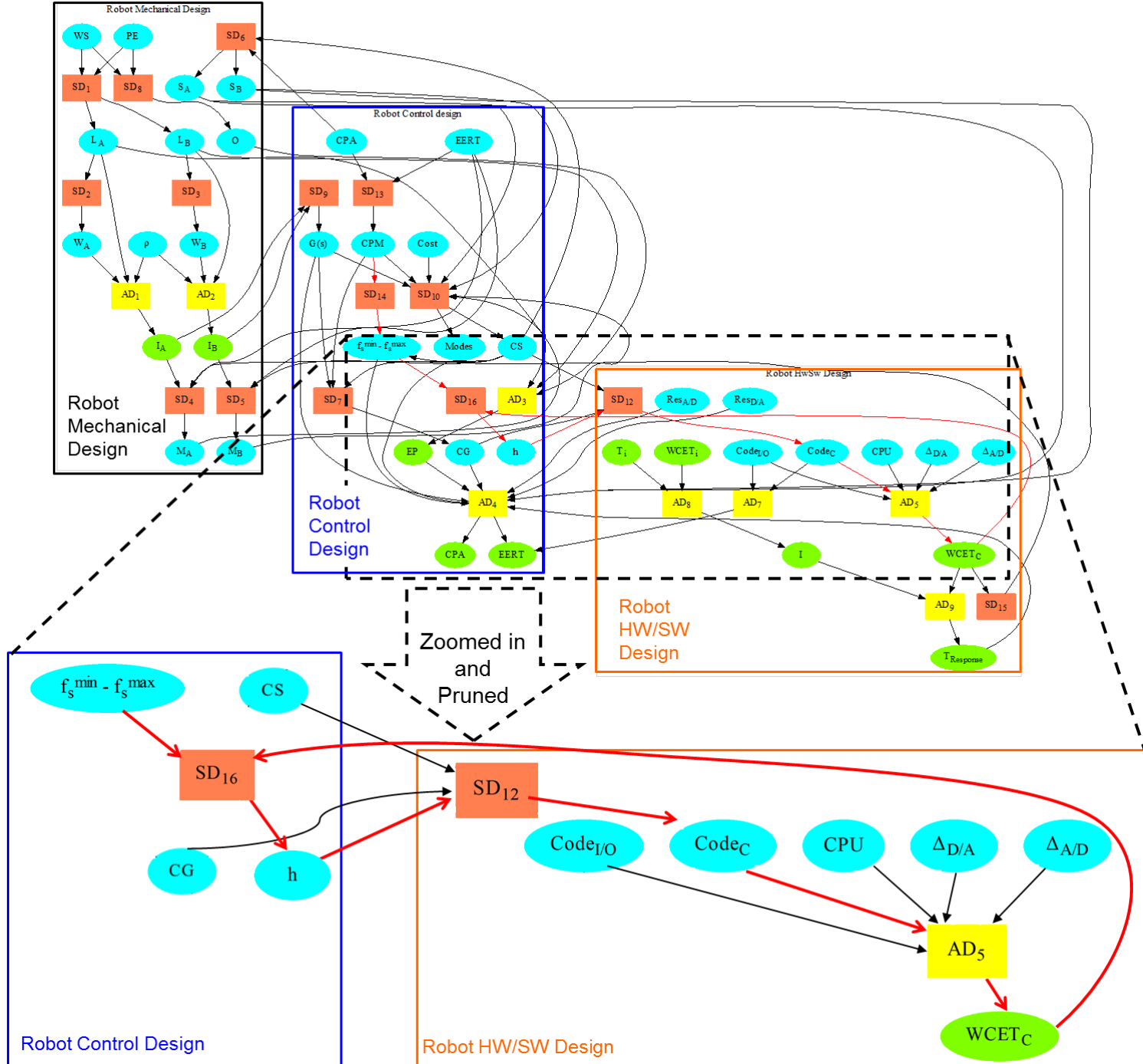


# Dependency model example

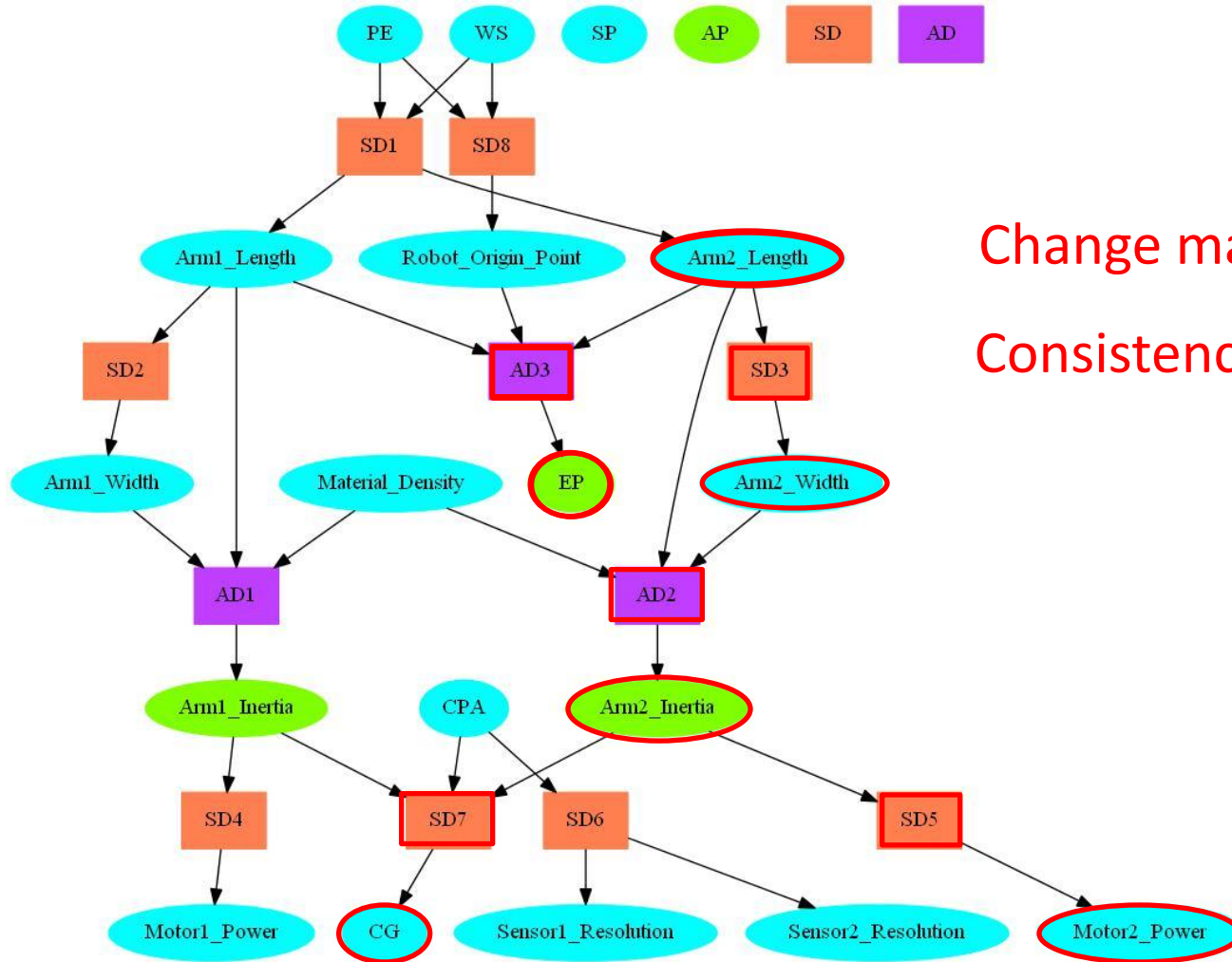


# Snapshot of the dependency graph for Mechanical, Control and Hw/Sw design of the robot





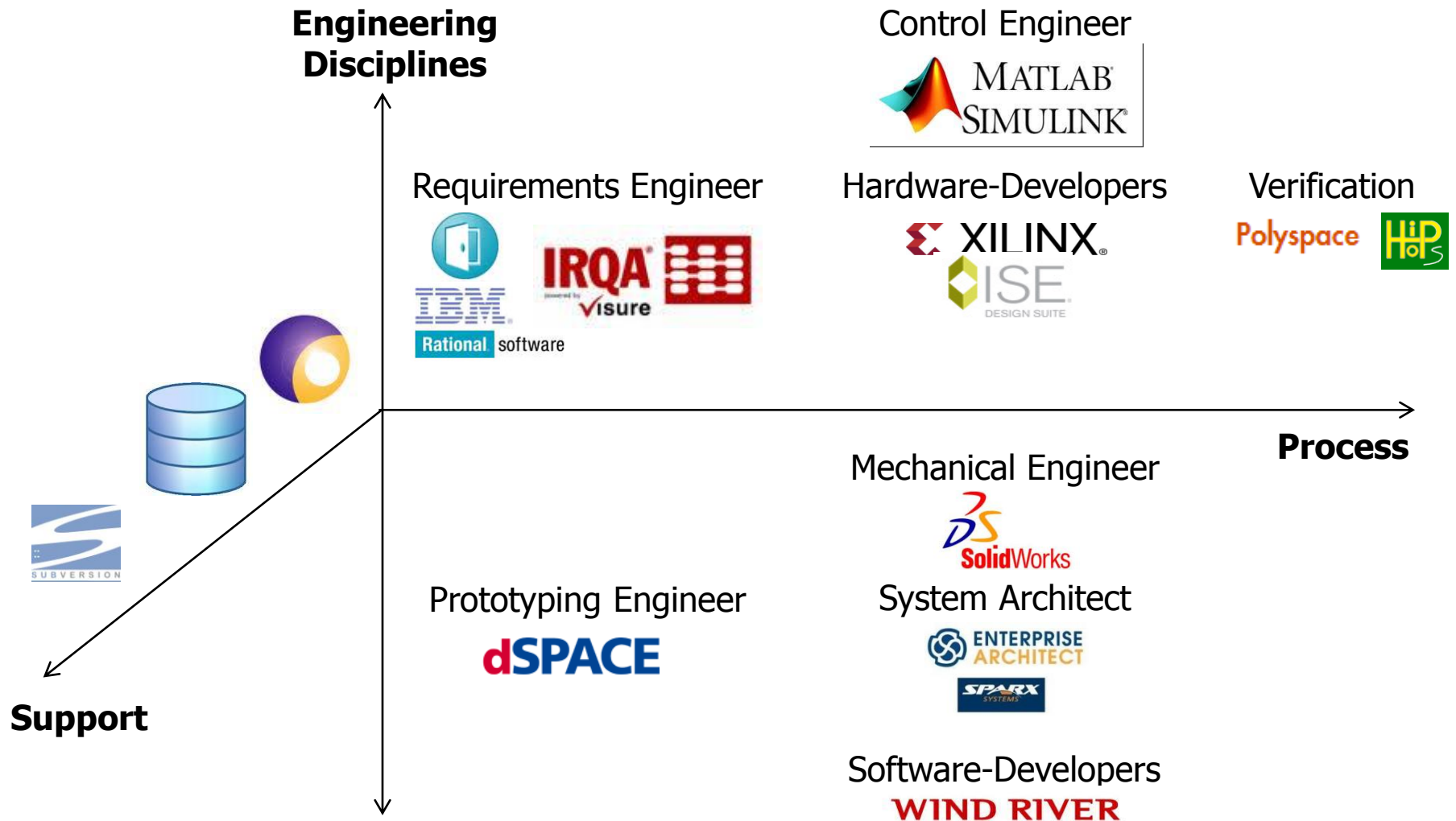
# Analysis made possible



Change management  
Consistency checking



# Development of Mechatronic products



# Challenges of Tool Integration



**Tools** provide limited native integration

- **island** solutions
- **difficult** to connect tools
  - to **transfer** data,
  - to create **traces**,
  - to create **scripts** involving multiple tools
- Hard to push workable tool integration standards

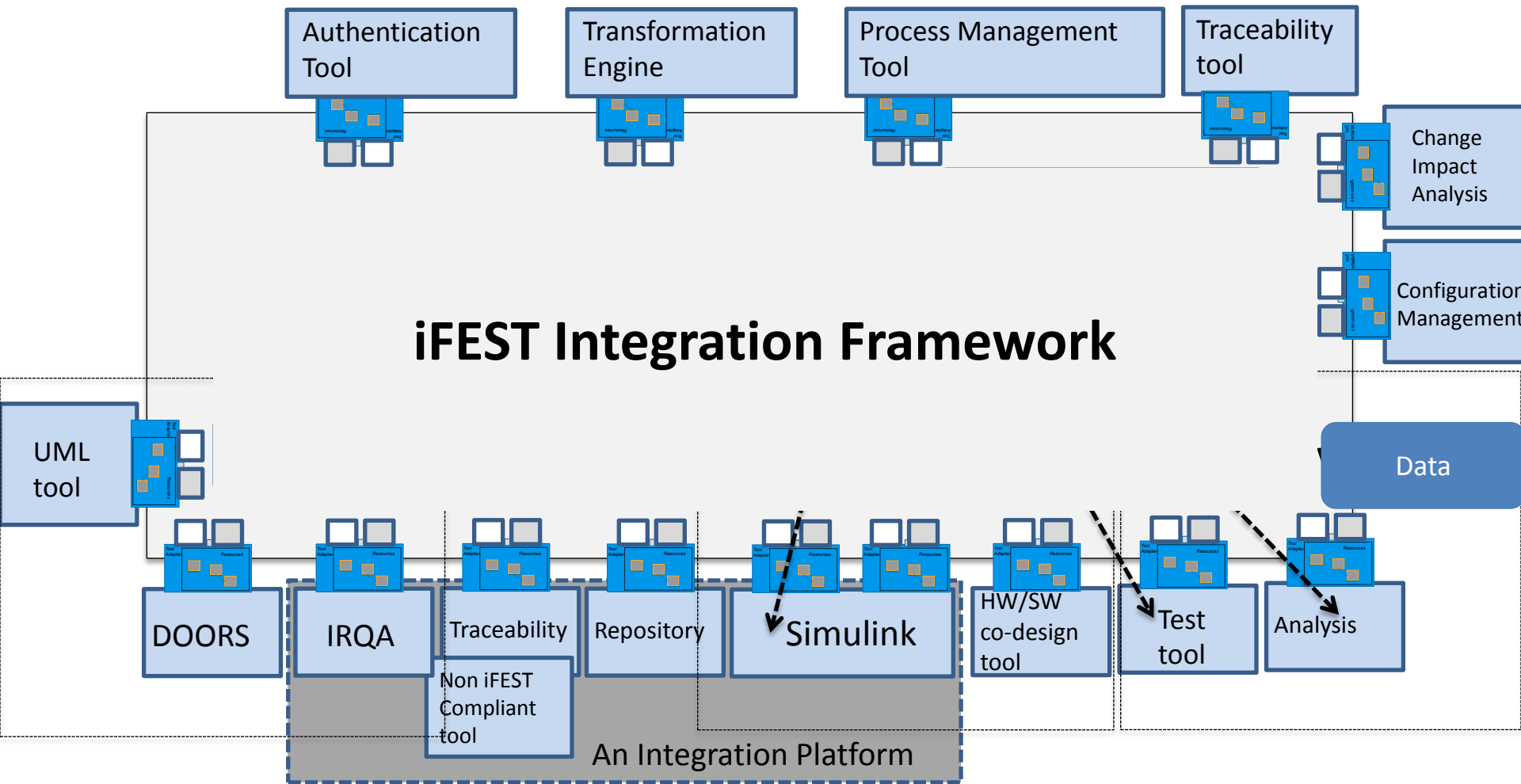
## Integration Concerns


Semantic

Structural

Technical

# Goal: Tool Chains Instantiation



- 
- Highly variable
- Lack of consistency



## WSDL

[illegible]

## BPEL

```

1 // ...
2
3 // Connect to the Apache OpenStack Foundation (ASF),
4 // or more conveniently Internet Archive, via the HTTP
5 // GET method.
6 // Note that the ASF website is not available in the
7 // US you want the Apache License, Version 2.0
8 // license, so we use the Internet Archive instead.
9 // Note that the ASF website is not available in the
10 // US you want the Apache License, Version 2.0
11 // license, so we use the Internet Archive instead.
12
13 // Get the Apache License, Version 2.0 license
14 // from the Internet Archive.
15
16 // ...
17
18 // ...
19
20 // ...
21
22 // ...
23
24 // ...
25
26 // ...
27
28 // ...
29
30 // ...
31
32 // ...
33
34 // ...
35
36 // ...
37
38 // ...
39
40 // ...
41
42 // ...
43
44 // ...
45
46 // ...
47
48 // ...
49
50 // ...
51
52 // ...
53
54 // ...
55
56 // ...
57
58 // ...
59
60 // ...
61
62 // ...
63
64 // ...
65
66 // ...
67
68 // ...
69
70 // ...
71
72 // ...
73
74 // ...
75
76 // ...
77
78 // ...
79
80 // ...
81
82 // ...
83
84 // ...
85
86 // ...
87
88 // ...
89
90 // ...
91
92 // ...
93
94 // ...
95
96 // ...
97
98 // ...
99
100 // ...
101
102 // ...
103
104 // ...
105
106 // ...
107
108 // ...
109
110 // ...
111
112 // ...
113
114 // ...
115
116 // ...
117
118 // ...
119
120 // ...
121
122 // ...
123
124 // ...
125
126 // ...
127
128 // ...
129
130 // ...
131
132 // ...
133
134 // ...
135
136 // ...
137
138 // ...
139
140 // ...
141
142 // ...
143
144 // ...
145
146 // ...
147
148 // ...
149
150 // ...
151
152 // ...
153
154 // ...
155
156 // ...
157
158 // ...
159
160 // ...
161
162 // ...
163
164 // ...
165
166 // ...
167
168 // ...
169
170 // ...
171
172 // ...
173
174 // ...
175
176 // ...
177
178 // ...
179
180 // ...
181
182 // ...
183
184 // ...
185
186 // ...
187
188 // ...
189
190 // ...
191
192 // ...
193
194 // ...
195
196 // ...
197
198 // ...
199
200 // ...
201
202 // ...
203
204 // ...
205
206 // ...
207
208 // ...
209
210 // ...
211
212 // ...
213
214 // ...
215
216 // ...
217
218 // ...
219
220 // ...
221
222 // ...
223
224 // ...
225
226 // ...
227
228 // ...
229
230 // ...
231
232 // ...
233
234 // ...
235
236 // ...
237
238 // ...
239
240 // ...
241
242 // ...
243
244 // ...
245
246 // ...
247
248 // ...
249
250 // ...
251
252 // ...
253
254 // ...
255
256 // ...
257
258 // ...
259
260 // ...
261
262 // ...
263
264 // ...
265
266 // ...
267
268 // ...
269
270 // ...
271
272 // ...
273
274 // ...
275
276 // ...
277
278 // ...
279
280 // ...
281
282 // ...
283
284 // ...
285
286 // ...
287
288 // ...
289
290 // ...
291
292 // ...
293
294 // ...
295
296 // ...
297
298 // ...
299
300 // ...
301
302 // ...
303
304 // ...
305
306 // ...
307
308 // ...
309
310 // ...
311
312 // ...
313
314 // ...
315
316 // ...
317
318 // ...
319
320 // ...
321
322 // ...
323
324 // ...
325
326 // ...
327
328 // ...
329
330 // ...
331
332 // ...
333
334 // ...
335
336 // ...
337
338 // ...
339
340 // ...
341
342 // ...
343
344 // ...
345
346 // ...
347
348 // ...
349
350 // ...
351
352 // ...
353
354 // ...
355
356 // ...
357
358 // ...
359
360 // ...
361
362 // ...
363
364 // ...
365
366 // ...
367
368 // ...
369
370 // ...
371
372 // ...
373
374 // ...
375
376 // ...
377
378 // ...
379
380 // ...
381
382 // ...
383
384 // ...
385
386 // ...
387
388 // ...
389
390 // ...
391
392 // ...
393
394 // ...
395
396 // ...
397
398 // ...
399
400 // ...
401
402 // ...
403
404 // ...
405
406 // ...
407
408 // ...
409
410 // ...
411
412 // ...
413
414 // ...
415
416 // ...
417
418 // ...
419
420 // ...
421
422 // ...
423
424 // ...
425
426 // ...
427
428 // ...
429
430 // ...
431
432 // ...
433
434 // ...
435
436 // ...
437
438 // ...
439
440 // ...
441
442 // ...
443
444 // ...
445
446 // ...
447
448 // ...
449
450 // ...
451
452 // ...
453
454 // ...
455
456 // ...
457
458 // ...
459
460 // ...
461
462 // ...
463
464 // ...
465
466 // ...
467
468 // ...
469
470 // ...
471
472 // ...
473
474 // ...
475
476 // ...
477
478 // ...
479
480 // ...
481
482 // ...
483
484 // ...
485
486 // ...
487
488 // ...
489
490 // ...
491
492 // ...
493
494 // ...
495
496 // ...
497
498 // ...
499
500 // ...
501
502 // ...
503
504 // ...
505
506 // ...
507
508 // ...
509
510 // ...
511
512 // ...
513
514 // ...
515
516 // ...
517
518 // ...
519
520 // ...
521
522 // ...
523
524 // ...
525
526 // ...
527
528 // ...
529
530 // ...
531
532 // ...
533
534 // ...
535
536 // ...
537
538 // ...
539
540 // ...
541
542 // ...
543
544 // ...
545
546 // ...
547
548 // ...
549
550 // ...
551
552 // ...
553
554 // ...
555
556 // ...
557
558 // ...
559
560 // ...
561
562 // ...
563
564 // ...
565
566 // ...
567
568 // ...
569
570 // ...
571
572 // ...
573
574 // ...
575
576 // ...
577
578 // ...
579
580 // ...
581
582 // ...
583
584 // ...
585
586 // ...
587
588 // ...
589
590 // ...
591
592 // ...
593
594 // ...
595
596 // ...
597
598 // ...
599
600 // ...
601
602 // ...
603
604 // ...
605
606 // ...
607
608 // ...
609
610 // ...
611
612 // ...
613
614 // ...
615
616 // ...
617
618 // ...
619
620 // ...
621
622 // ...
623
624 // ...
625
626 // ...
627
628 // ...
629
630 // ...
631
632 // ...
633
634 // ...
635
636 // ...
637
638 // ...
639
640 // ...
641
642 // ...
643
644 // ...
645
646 // ...
647
648 // ...
649
650 // ...
651
652 // ...
653
654 // ...
655
656 // ...
657
658 // ...
659
660 // ...
661
662 // ...
663
664 // ...
665
666 // ...
667
668 // ...
669
670 // ...
671
672 // ...
673
674 // ...
675
676 // ...
677
678 // ...
679
680 // ...
681
682 // ...
683
684 // ...
685
686 // ...
687
688 // ...
689
690 // ...
691
692 // ...
693
694 // ...
695
696 // ...
697
698 // ...
699
700 // ...
701
702 // ...
703
704 // ...
705
706 // ...
707
708 // ...
709
710 // ...
711
712 // ...
713
714 // ...
715
716 // ...
717
718 // ...
719
720 // ...
721
722 // ...
723
724 // ...
725
726 // ...
727
728 // ...
729
730 // ...
731
732 // ...
733
734 // ...
735
736 // ...
737
738 // ...
739
740 // ...
741
742 // ...
743
744 // ...
745
746 // ...
747
748 // ...
749
750 // ...
751
752 // ...
753
754 // ...
755
756 // ...
757
758 // ...
759
760 // ...
761
762 // ...
763
764 // ...
765
766 // ...
767
768 // ...
769
770 // ...
771
772 // ...
773
774 // ...
775
776 // ...
777
778 // ...
779
780 // ...
781
782 // ...
783
784 // ...
785
786 // ...
787
788 // ...
789
790 // ...
791
792 // ...
793
794 // ...
795
796 // ...
797
798 // ...
799
800 // ...
801
802 // ...
803
804 // ...
805
806 // ...
807
808 // ...
809
810 // ...
811
812 // ...
813
814 // ...
815
816 // ...
817
818 // ...
819
820 // ...

```

# Java

```

1 package org.nuxeo.upcoming.test;
2
3 import java.io.Serializable;
4
5 public class EventTest extends Runnable {
6     // If the current class does not respect the contract
7     // rule, RuleChecker.setPropertyValue()
8     // from
9     // a : URL-Class (not S.checkIfPropertyWithValue())
10     // to
11     // t : Problem/Problem
12     // severity <= Error,
13     // description <= "default value of the pr
14     // a : name
15     // b : should be different
16     // location <= "11-11"
17     // )
18     // MappingStr, Serializable
19     props.put("test-name", "testEventBasic");
20     //The event service is a really an "EventGenerator" in some sense
21     EventContext contextName = InlineContext.createNull(props);
22     Event syntheticEvent = contextName.newEvent(EventService.class);
23
24     //the event service is the center of all event processing...
25     EventService service = new EventService(getClassLoader(),
26     assertNotNull("Service not found", service);
27
28     //EventListenerHelper is the wrapper around our DocumentCreationListener
29     service.addEventListener(new EventListenerHelper() {
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

ATL

```

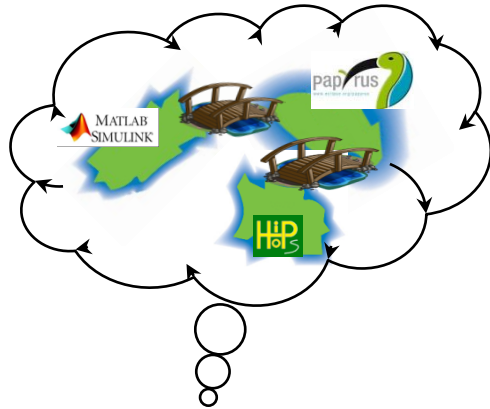
----- Check property value
-----
-- This helper verifies if the value of the property 'myProperty' is different of 'value'
-- helper content UML:Class def : checkMyProperty(value) : Boolean =
    if (self.hasStereoType(MyStereoType)) then
        if (self.getTaggedValueFromStereoTypeName('MyStereoType', 'myProperty') <> 'value') then
            true
        else
            false
        end if
    else
        true
    end if;
end if;

-- If the current class does not respect the condition an error is created
rule Rule_checkMyProperty {
    from
        s : UML:Class (not s.checkMyPropertyValue())
    to
        t : Problem:Problem {
            severity <= $error;
            description <- 'Default value of the property MyProperty in the class '
                + s.name
                + ' should be different from "value"',
            location <- 'l11-l11'
        }
}

```



# Approach



Stakeholders



Modeling  
Language

Development Support:

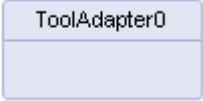





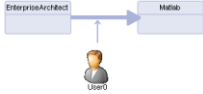

- Design/Refinement
- Analysis
- Synthesis

Tool Chain Impl.

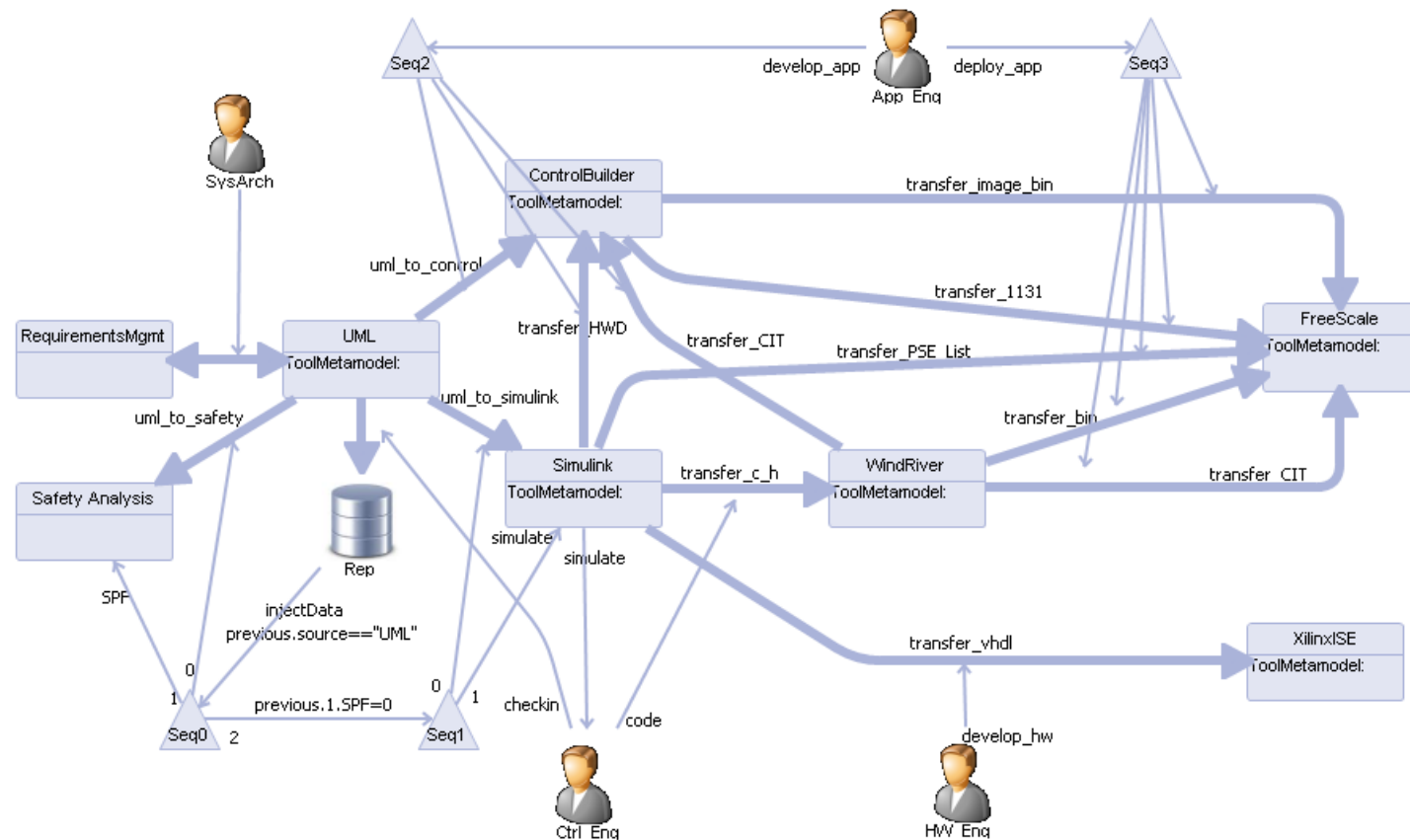


# Tool Integration Language (TIL)

## Concrete Syntax

	ToolAdapter		DataChannel
	User		TraceChannel
	Repository		ControlChannel
	ToolChain		Sequencer

# Case Study: A TIL Model for an industrial embedded systems tool chain





# Status

- Several papers, reports, case studies and PhD thesis on the presented viewpoint integration techniques
- Dependency modeling and Tool integration implemented as Domain specific modeling languages
- Viewpoint contracts so far applied to Control-Embedded systems and Control-Mechanical design

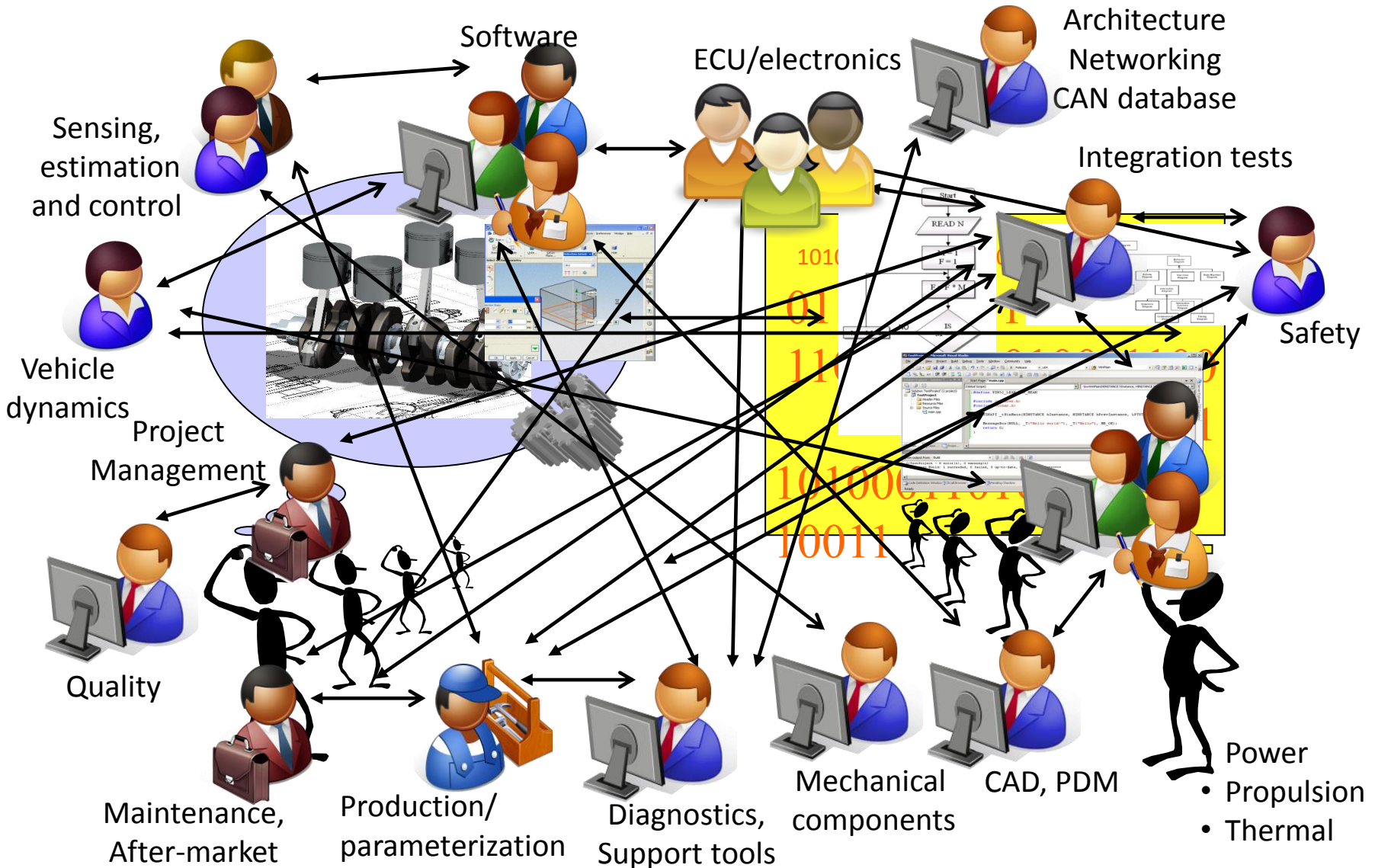
# Discussion

- Viewpoint integration techniques can be applied in different ways
  - Informally / Formally
  - Individually / Together
  - Top-down / Bottom-up
- Complements other techniques such as
  - Co-simulation
  - Component contracts
  - Integration specific views (e.g. ADLs, function models)

# Outline

- CPS Characterization
- Engineering Environments
  - Problem analysis
  - Multiview modeling
- Integrating viewpoints
  - Contracts, Dependency modeling, Tool integration
  - Discussion
- **Wrap-up**

# Viewpoints and relationships



# Summary and directions

- Trends: Functional growth, connectivity, scale, complexity
- Engineering of increasing no. of viewpoints and dependencies
  - Make viewpoint, dependencies and tool interactions explicit
- Dealing with and designing dependencies
  - Design contracts – towards viewpoint contracts
  - Dependency modeling
  - Systematic tool integration

Contact [MartinT@kth.se](mailto:MartinT@kth.se)

# References

Fredrik Asplund, Lic. Thesis, Mechatronics, KTH (Oct. 2012): “Tool Integration and Safety: A Foundation for Analyzing the Impact of Tool Integration on Non-functional Properties”.

Matthias Biehl, PhD thesis, Mechatronics KTH (Jan. 2013): “A Modeling Language for the Description and Development of Tool Chains for Embedded Systems”.

Ahsan Qamar, PhD thesis, Mechatronics, KTH (March, 2013): “Model and Dependency Management in Mechatronic Design”.

Magnus Persson, PhD thesis, Mechatronics, KTH (June, 2013): “A formalized approach to multi-view components for embedded systems”.

Martin Törngren, Matthias Biehl, Ahsan Qamar, Jad Elkhoury, and Frederic Loiret. “Multiview Modeling and Integration for Mechatronics Engineering”. Proc. of the ACCM-Workshop on Mechatronic Design, Linz, Austria, November, 2012.

Patricia Derler, Edward Lee, Martin Törngren, Stavros Tripakis. Cyber-Physical System Design Contracts, ICCPS '13: ACM/IEEE 4th International Conference on Cyber-Physical Systems, 2013.