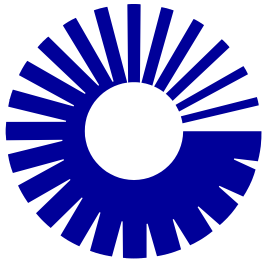


United Technologies

Systems Engineering

Status of Industrial Use, Opportunities and Needs

Clas A. Jacobson
Chief Scientist
Systems & Controls Engineering
LCCC
Lund

September 19, 2012

TEAM

Alberto Sangiovanni Vincentelli, Alberto Ferrari, Mark Myers, John Cassidy, Richard Murray, Andrzej Banaszuk, Sean Meyn, Johan Akesson, Hubertus Tummescheit, Karl Astrom, Manfred Morari, Eelco Scholte, Rich Poisson, Satish Narayanan, Kevin Otto, John Burns, Igor Mezic, Marco Di Natale, Scott Bortoff…

AGENDA

System Design

Systems engineering:
(1) requirements,
(2) architecture,
(3) model based design,
(4) (design/development) process

Platform Based Design – design flows (orthogonalize concerns; hierarchy)

Opportunities & progress

System level modeling – positive on reusabality, speed…
Architecture exploration – not fully exploited - but enabled
Requirements – potential to move between formal languages (in progress for embeddded systems)
Model based development – positive on controls - MPC (and optimization), uncertainty (and use for robust design not there yet)
Process – progress on integration of tool chains; level of abstraction change (slightly) with domain (but separate into main product development cycles)

DRIVERS

System interactions (“emergent behavior”)

Requirements & acceptance testing (verification)

Safety (critical) (software intensive) systems

Reusable architectures (modularity)

Robustness (risk, lifing)

SYSTEMS ENGINEERING (DESIGN)

Process

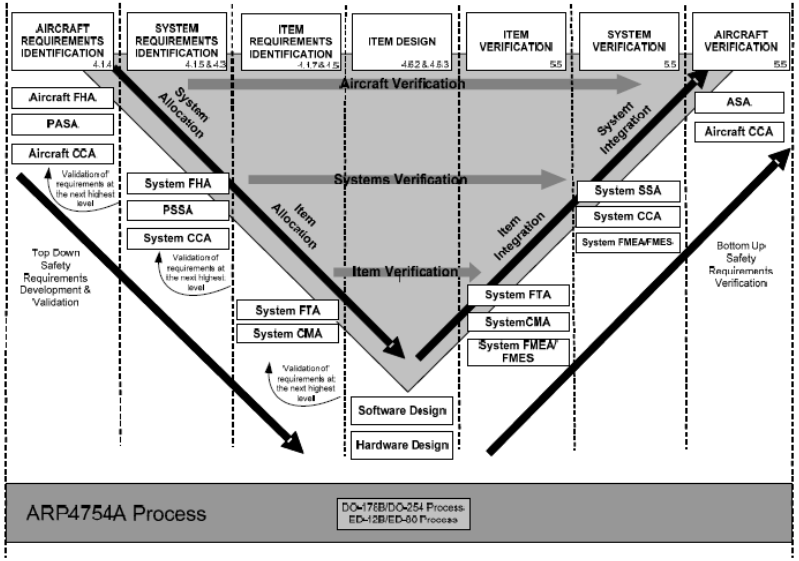
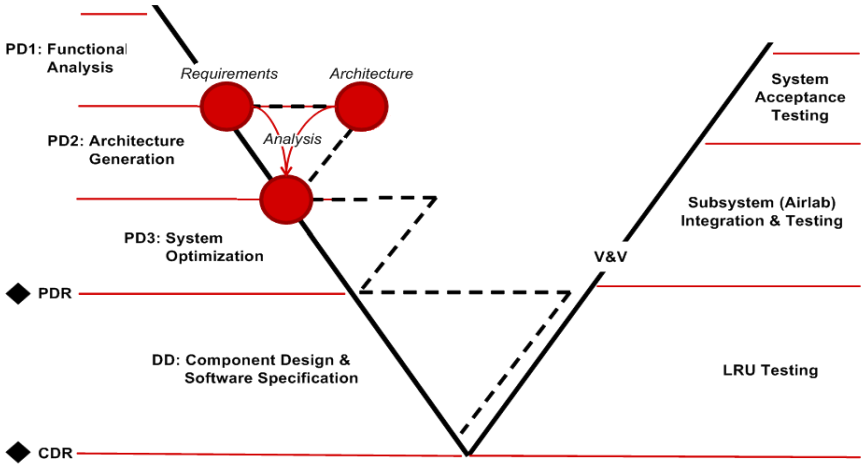


FIGURE 5 - INTERACTION BETWEEN SAFETY AND DEVELOPMENT PROCESSES



From process to analysis (model based development)

Bring forward in time the verification testing (SIL => HIL => acceptance)

Orthogonalize requirements (requested behavior) and architecture (delivering services)

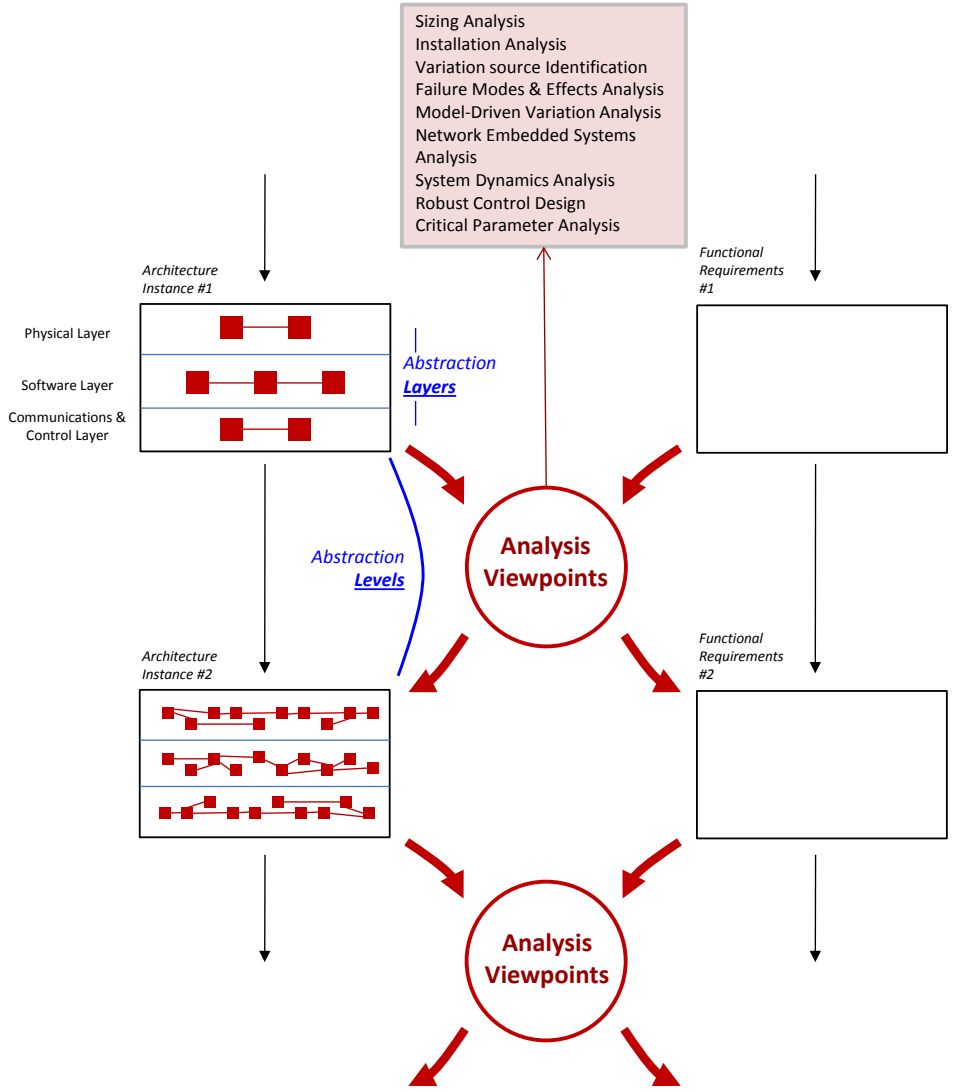
DESIGN PROCESS

Status & Opportunities

Range of analyses (views)

Hierarchy (refinement)

Separation of concerns (requirements, architecture, analysis)



SYSTEMS ENGINEERING (DESIGN)

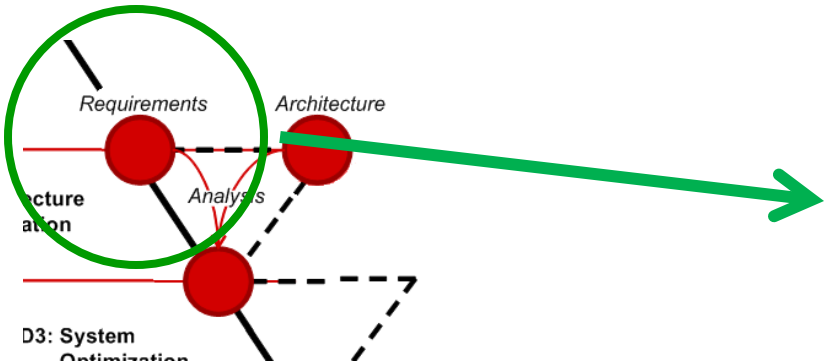
Definition

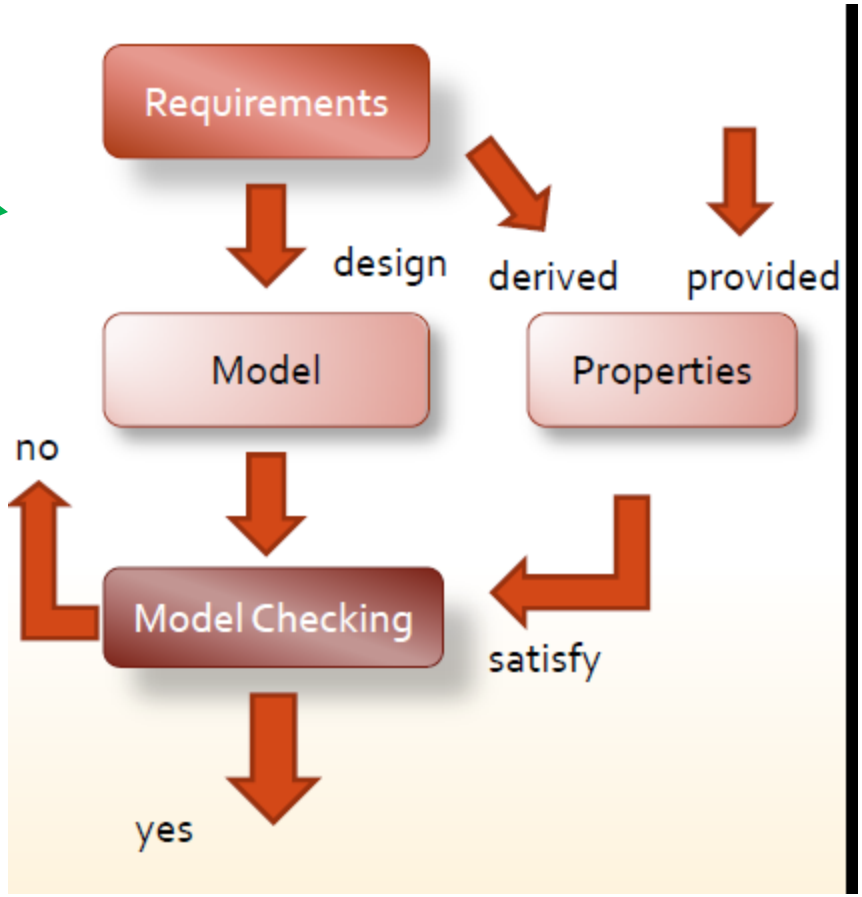
Systems engineering is a methodology for product system level design, optimization and verification that:

1. Provides guarantees of performance and reliability against customer **requirements** while achieving business cost and time-to-market objectives;
2. Produces modular, extensible **architectures** for products incorporating mechanical components, embedded systems and application software;
3. Exploits **model-based analytical tools and techniques** to determine design choices and ensure robust system performance despite variations caused by product manufacturing, integration with other products and customer operation; and
4. achieves these objectives through the coordinated execution of a prescriptive, repeatable and measurable **process.**

REQUIREMENTS

Status & Opportunities





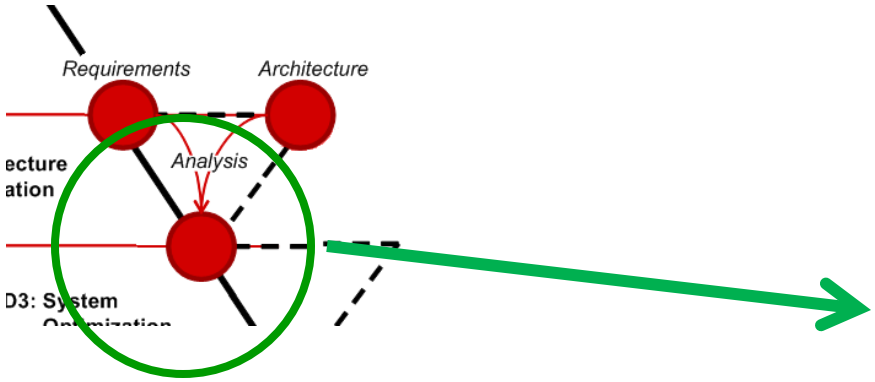
Enabler – formal language (not just equation based language)

Status – strong for embedded systems; weak for continuous time (non-simulation based verification)

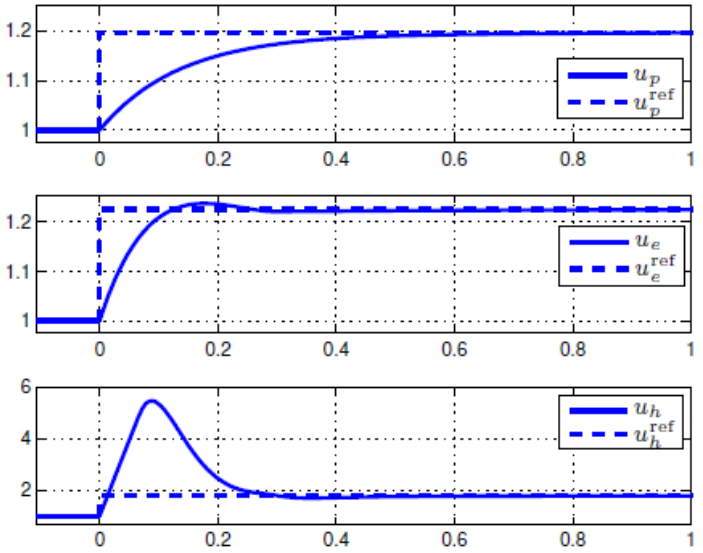
Opportunity – robust design/uncertainty

MODEL BASED DEVELOPMENT

Status & Opportunities




```
optimization VDP_Opt(objective=cost(finalTime),
      startTime=0,
      finalTime(free=true, initialGuess=1))
   VDP vdp(u(free=true,initialGuess=0.0));
    Real cost (start=0);
equation
    der(cost) = 1;
constraint
    vdp.x1(finalTime) = 0;
    vdp.x2(finalTime) = 0;
    vdp.u >= -1; vdp.u <= 1;
end VDP_Opt;
```

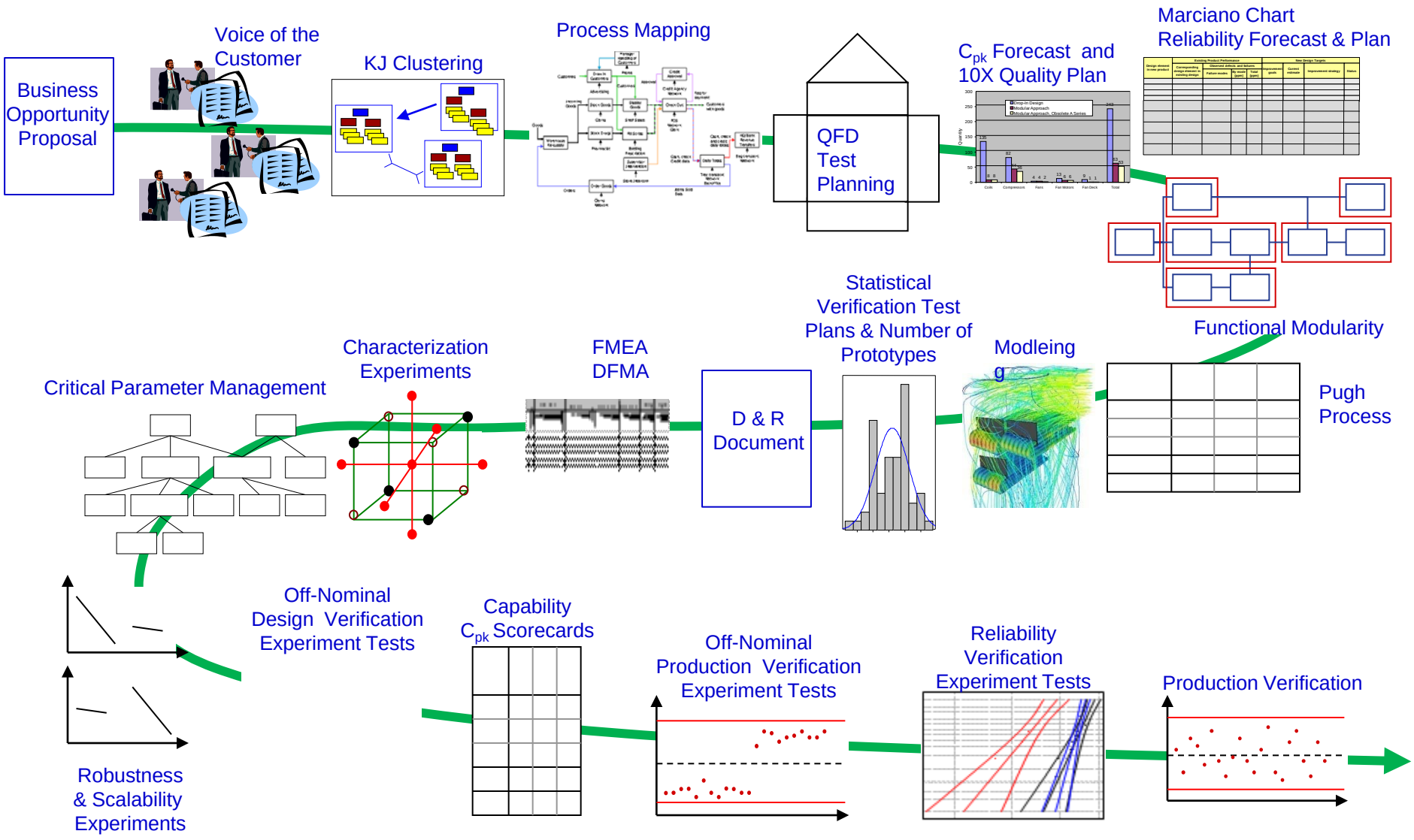


Enabler – equations; interconnection structure

Status of use of equation based language – strong for optimization (MPC; Akesson-Optimica) ; not exploited for robust design; weak for architecture exploration

Opportunity – robust design/uncertainty

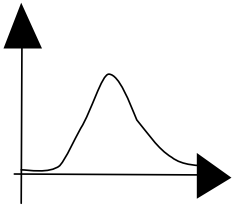
ROBUST DESIGN

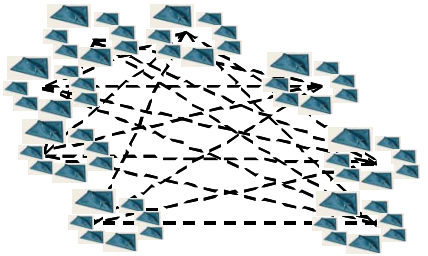


ROBUST DESIGN & UNCERTAINTY

Status & Opportunities: Exploit Structure

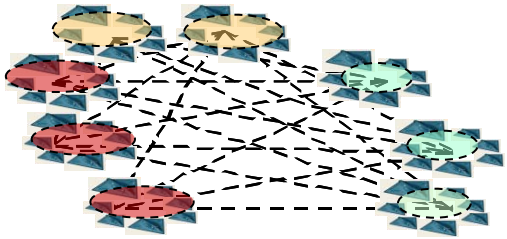
Probability Distribution of input parameters



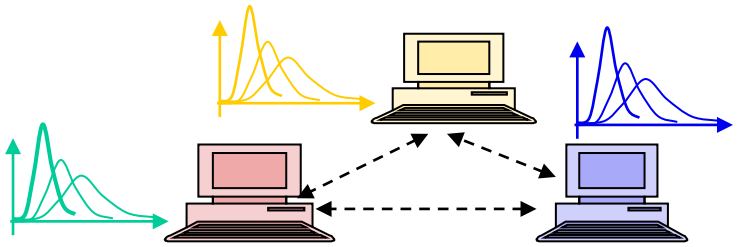


Utilize interconnection structure to tear system into strong & weak connections; propagate uncertainty (Meyn-Mathew (and others) DARPA RUM 2008)

Find Weak Interaction



Exploit Weak Interaction



SUMMARY

System Design

Systems engineering :
(1) requirements,
(2) architecture,
(3) model based design,
(4) process

Platform Based Design – design flows (orthogonalize concerns; hierarchy)

Opportunities & progress

System level modeling – positive on reusabality, speed…
Architecture exploration – not fully exploited - but enabled
Requirements – potential to move between formal languages (in progress for embeddded systems)
Model based development – positive on controls - MPC (and optimization), uncertainty (and use for robust design not there yet)
Process – progress on integration of tool chains; level of abstraction change (slightly) with domain (but separate into main product development cycles)

Summary

Big needs on uncertainty/robust design (much wider view of product development);
Opportunity for realizing potential of tool integration (FMI) and with PLM (data management)

KEY POINTS

System Design

Systems engineering :
(1) requirements,
(2) architecture,
(3) model based design,
(4) process

Platform Based Design – design flows (orthogonalize concerns; hierarchy)

Opportunities & progress

System level modeling – positive on reusabality, speed…
Architecture exploration – not fully exploited - but enabled
Requirements – potential to move between formal languages (in progress for embeddded systems)
Model based development – MPC (and optimization), uncertainty (not there yet)
Process – integration of tool chains; level of abstraction change (slightly) with domain (but separate into main product development cycles)

Summary

Big needs on uncertainty/robust design;
Opportunity for realizing potential of integration (FMI) with tool chain and PLM