Machine Learning In Model Based Engineering

GLOBAL PRODUCT DATA INTEROPERABILITY
SUMMIT
2018

ELYSIUM Parker Aerospace NORTHROP GRUMMAN BOEING

Sunil Elanayar, Ph.D.
Phillip Austin
Juan Sanchez De Muniain

Presentation outline

- **Model Based Engineering and Computational Design Synthesis (CDS)**
  - **Design Synthesis done manually**
  - **CDS using Graph Transformations**
- **Function Behavior Structure (FBS) role in CDS**
  - **Framework for CDS**
  - **FBS Levels in Modeling**
  - **Role of Knowledge Based Engineering (KBE) in CDS**
  - **Building Functional Structures using MOKA (M**ethodology and software tools **O**riented to **K**nowledge-based engineering **A**pplications)
- **Behavior Modeling and Digital Twins**
  - **Auto Suspension Systems, Production Systems**
  - **Need for Black Box modeling**
  - **Radial Basis Function Networks for Dynamic Black Box Modeling**
  - **Learning Dynamic Models**
- **Proposed Workflow for CDS with Digital Twin**

Authors' bios

- ## Sunil Elanayar

Joined Boeing in 2007 from Dassault Systemes. Currently an IT Manager in Engineering Systems. Managing teams working in MBSE, Product Standards. In the past, he has worked in Visualization, New Wiring Systems, Aerodynamics, Knowledge Based Engineering (KBE), Machine Learning, and Optimization.

- ## Phillip Austin

Boeing Software Developer with experience in BDS Engineering Systems, BDS Manufacturing Systems, IPDM Teamcenter, BCA Enovia. Currently working in BCA Product Standards and 2CES MBSE Development.

- ## Juan Sanchez De Muniain

Software developer at Boeing since 2014, now on the 2CES PLM effort at Boeing; before that, developing computer-aided engineering and manufacturing applications primarily around product standards.

Enabling Model Based Engineering

To Enable MBE capabilities, tools need the ability to:

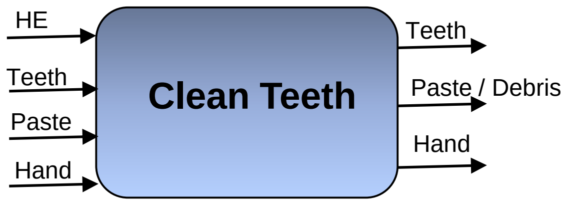
- Model products at multiple levels of abstraction, i. e. Function, Behavior and Structure (FBS)
- Formalize Engineering Knowledge with a view to manipulate, transform, and reuse it.
- Enable the quick generation and adaptation of designs (especially in the conceptual phases.)
- Decrease the tedium in routine design tasks
- Support innovation in the design process
- Help keep behavioral models current in the light of IOT
- Support Computational Design Synthesis
- Support Digital Twin concepts

Benefits of Computational Design Synthesis (CDS):

1. Increases the efficiency of the design process and the creation of new solutions,
2. Facilitates design reuse during concept generation,
3. Enables the exploration of larger design solution spaces, and thereby removes psychological bias that may limit designers to previous solutions or to specific engineering domains.

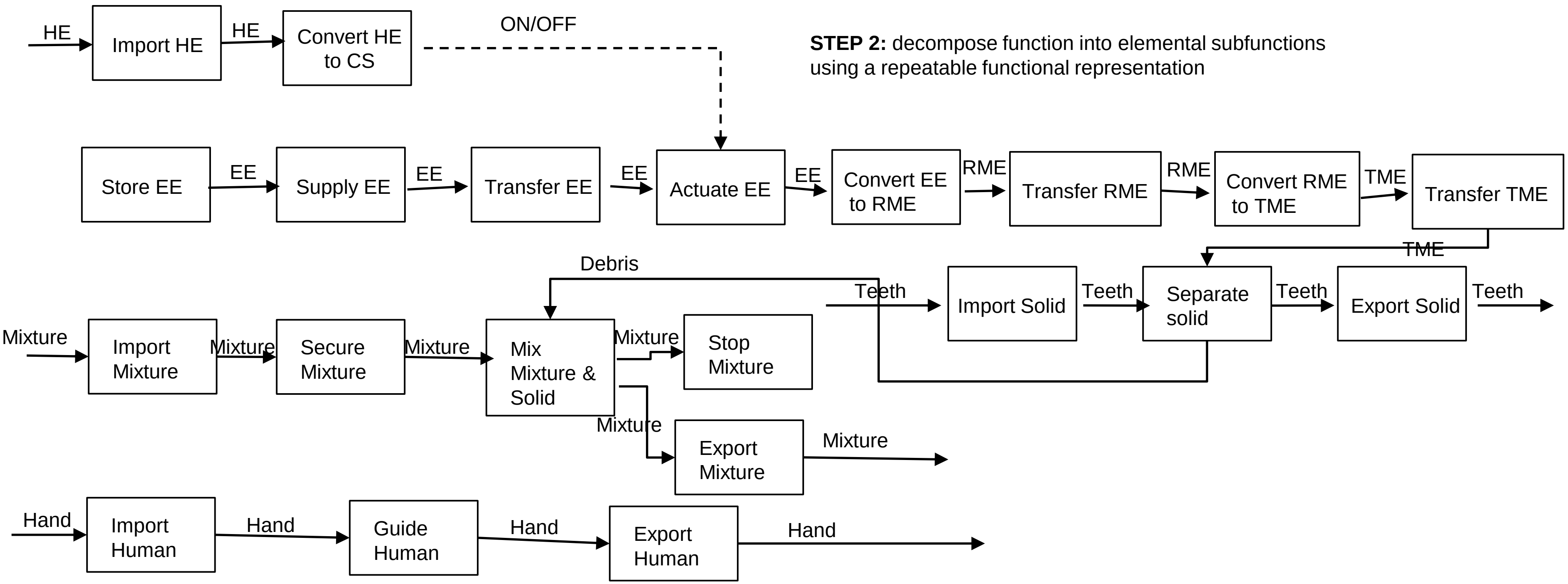
The Process of Design Synthesis: Functional Decomposition

STEP1: transform customer needs into an overall product function



Legend:
HE - Human Energy
TME - Translational Mechanical Energy
RME – Rotational Mechanical Energy
EE – Electrical Energy
CS – Control Signal

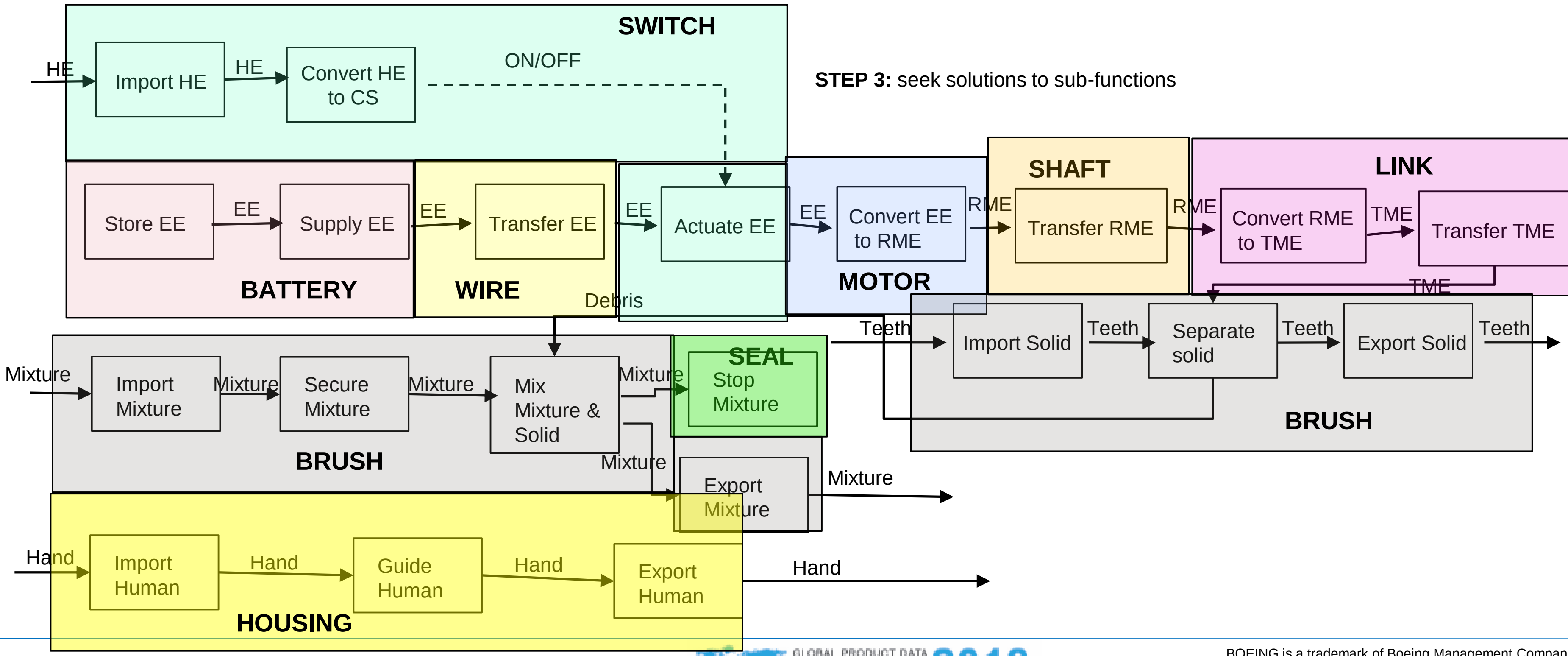
STEP 2: decompose function into elemental subfunctions using a repeatable functional representation



The Process of Design Synthesis: Form Mapping

Legend:
HE - Human Energy
TME - Translational Mechanical Energy
RME – Rotational Mechanical Energy
EE – Electrical Energy
CS – Control Signal

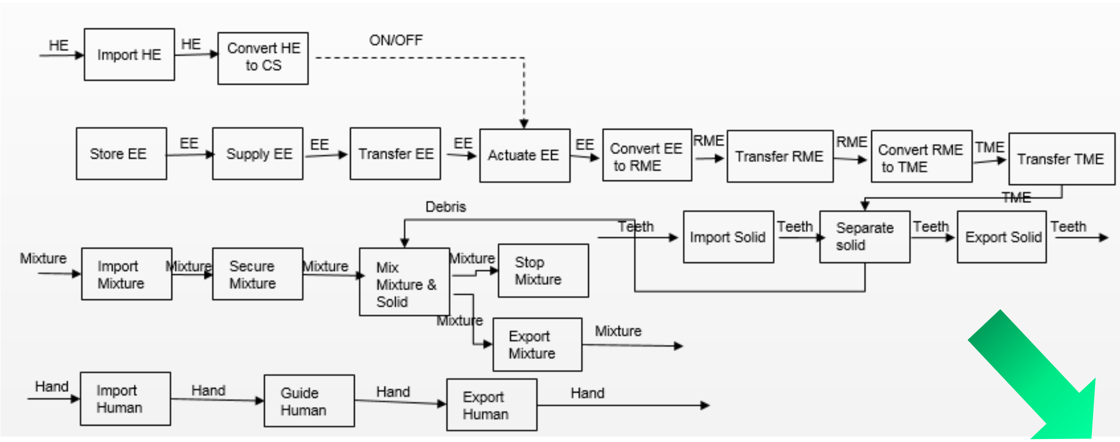
STEP 3: seek solutions to sub-functions



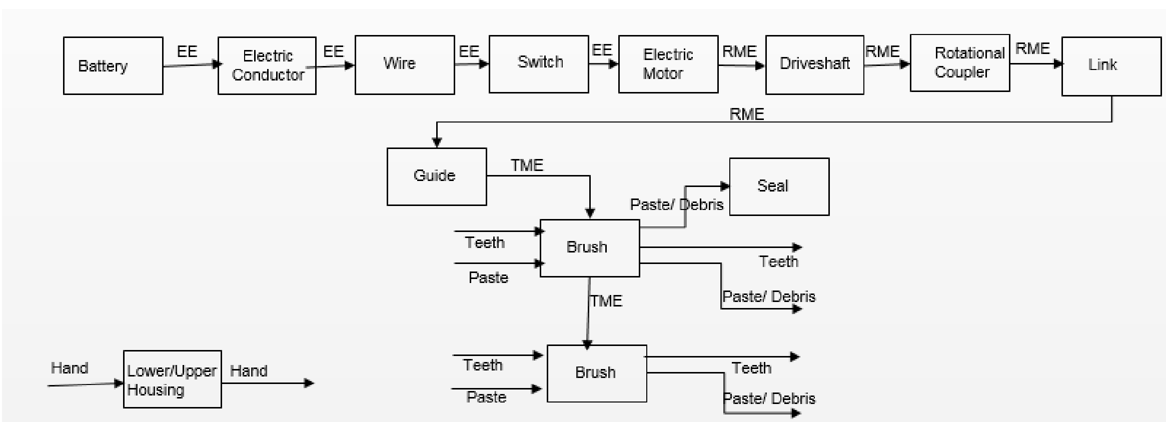
Computational Design Synthesis: Graph Transformations

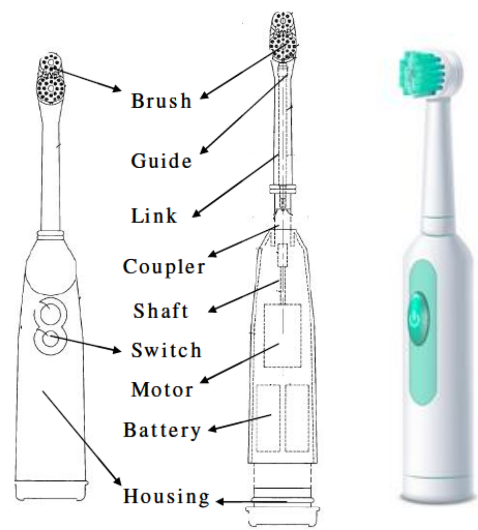
Global Product Data Interoperability Summit | **2018**

Function Structure of Electric Toothbrush



Configuration Flow Graph of Electric Toothbrush





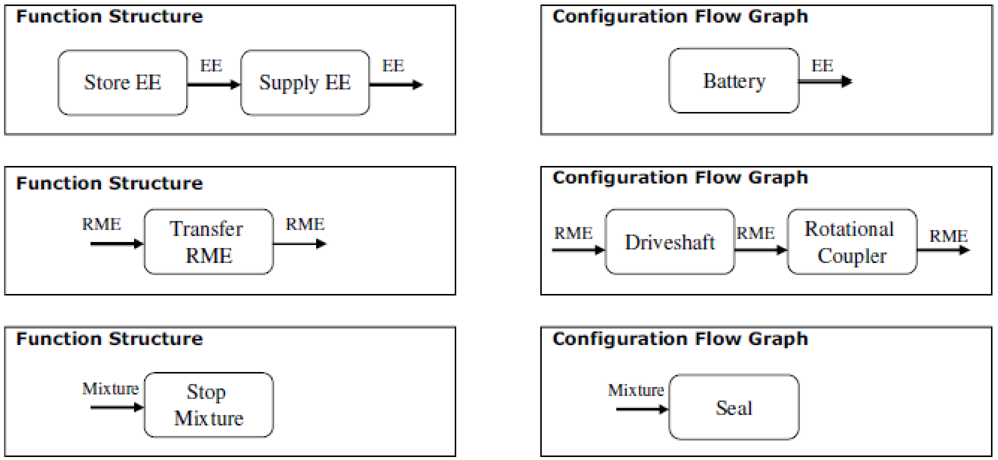
Graph Transformations / Graph Grammars

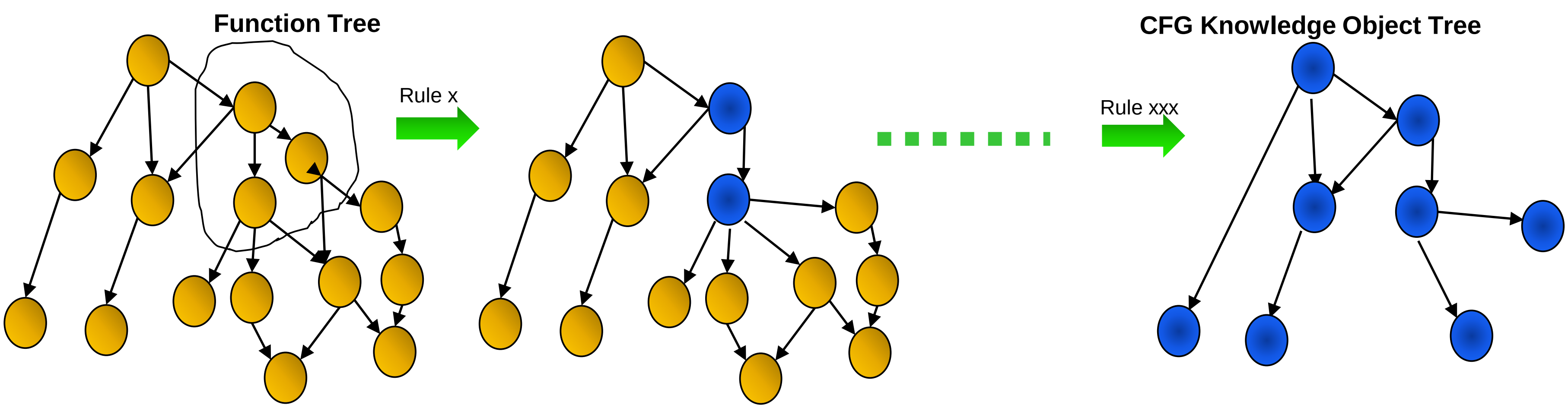
Global Product Data Interoperability Summit | **2018**

rule I

rule II

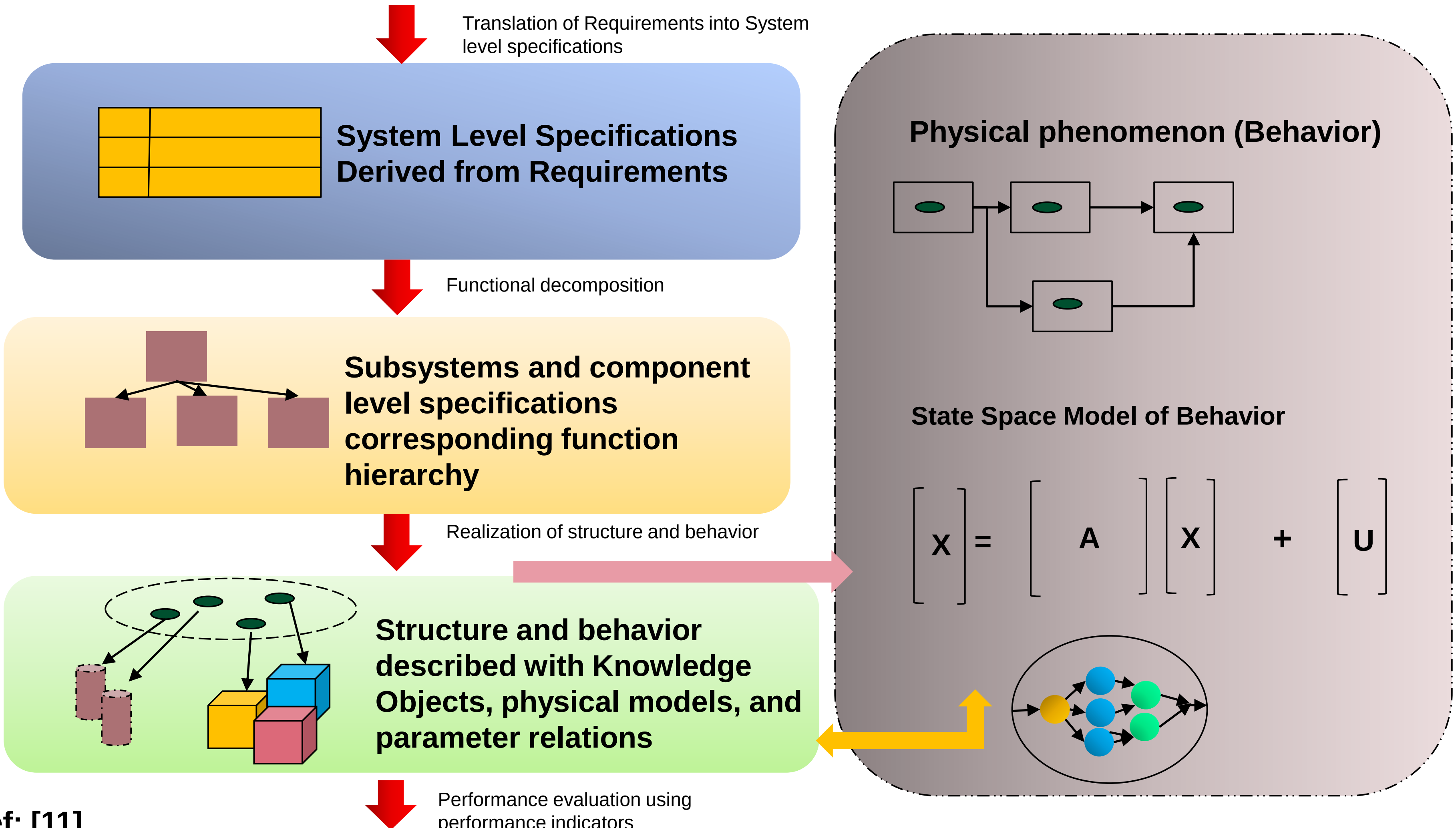
rule III





FBS Framework for Computational Design Synthesis in MBE

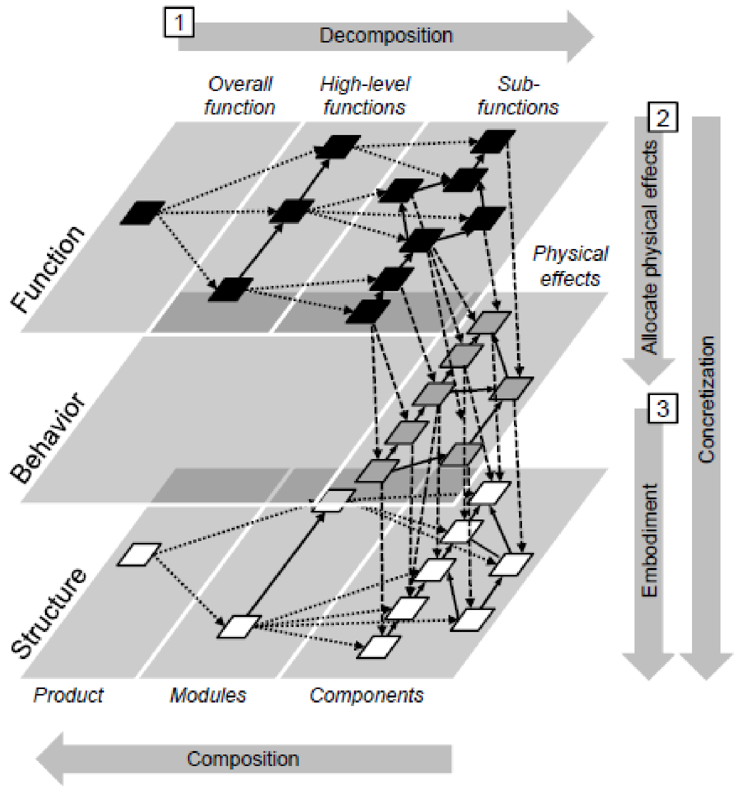
Global Product Data Interoperability Summit | **2018**

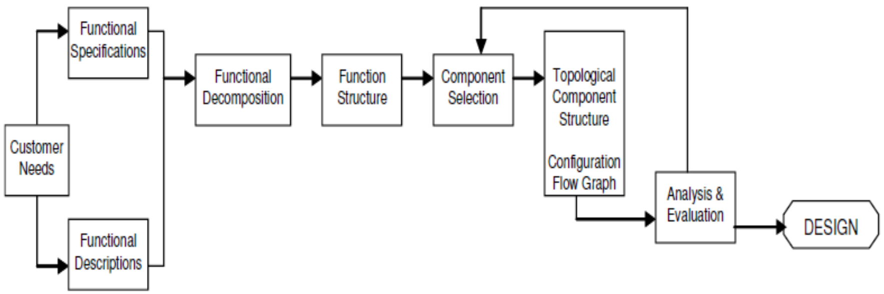


Ref: [11]

FBS Levels and Process of Computational Design Synthesis

Global Product Data Interoperability Summit | **2018**

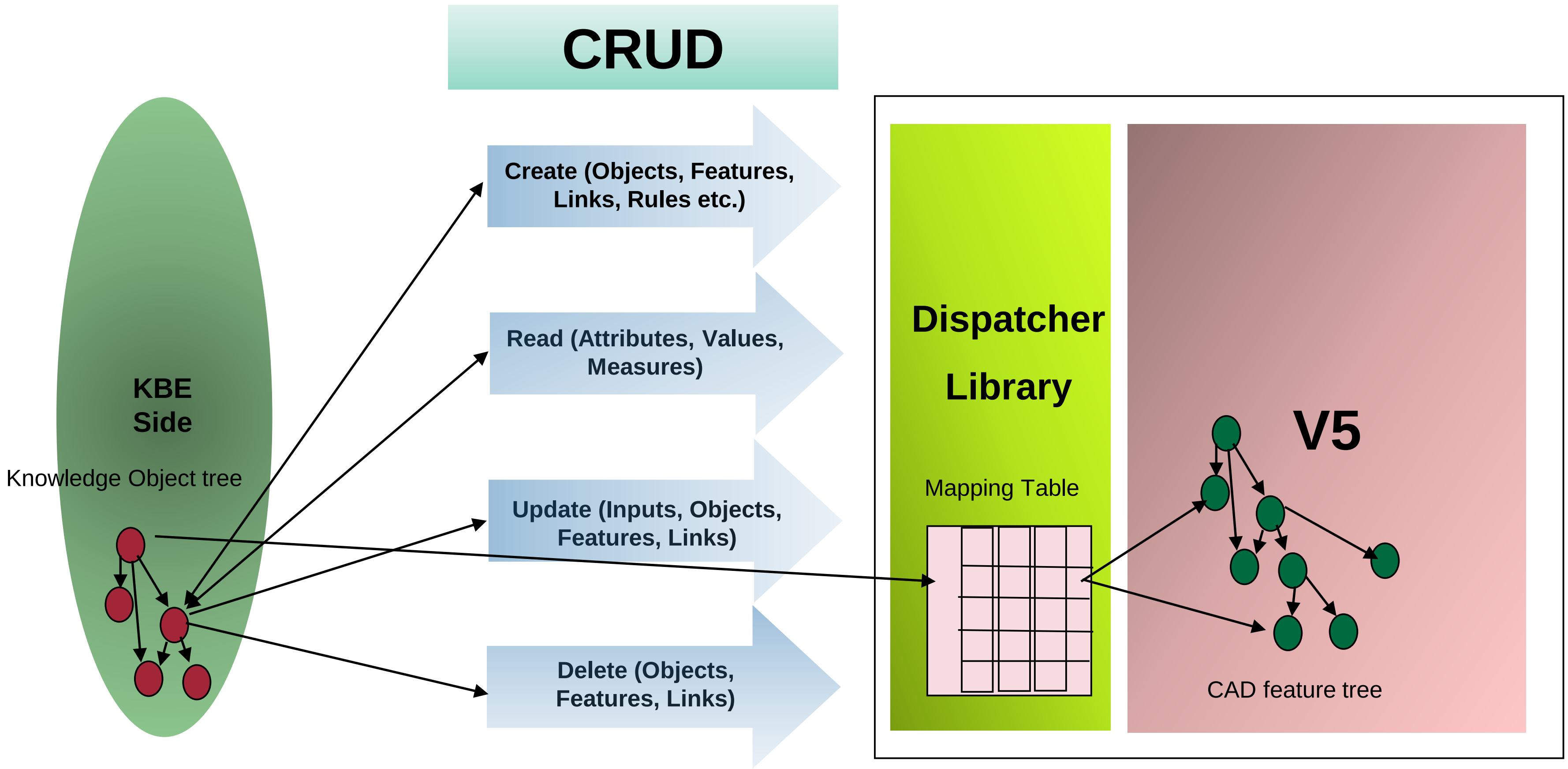




Ref: [7,9]

Knowledge Object Trees and CAD Interactions

Global Product Data Interoperability Summit | **2018**



Enriching Computational Design Synthesis Using Knowledge Objects

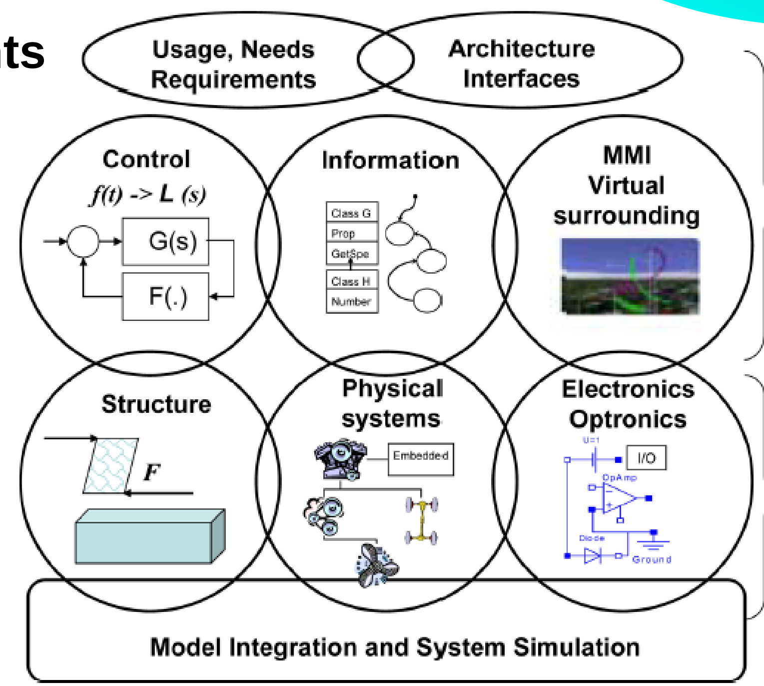
Global Product Data Interoperability Summit | **2018**

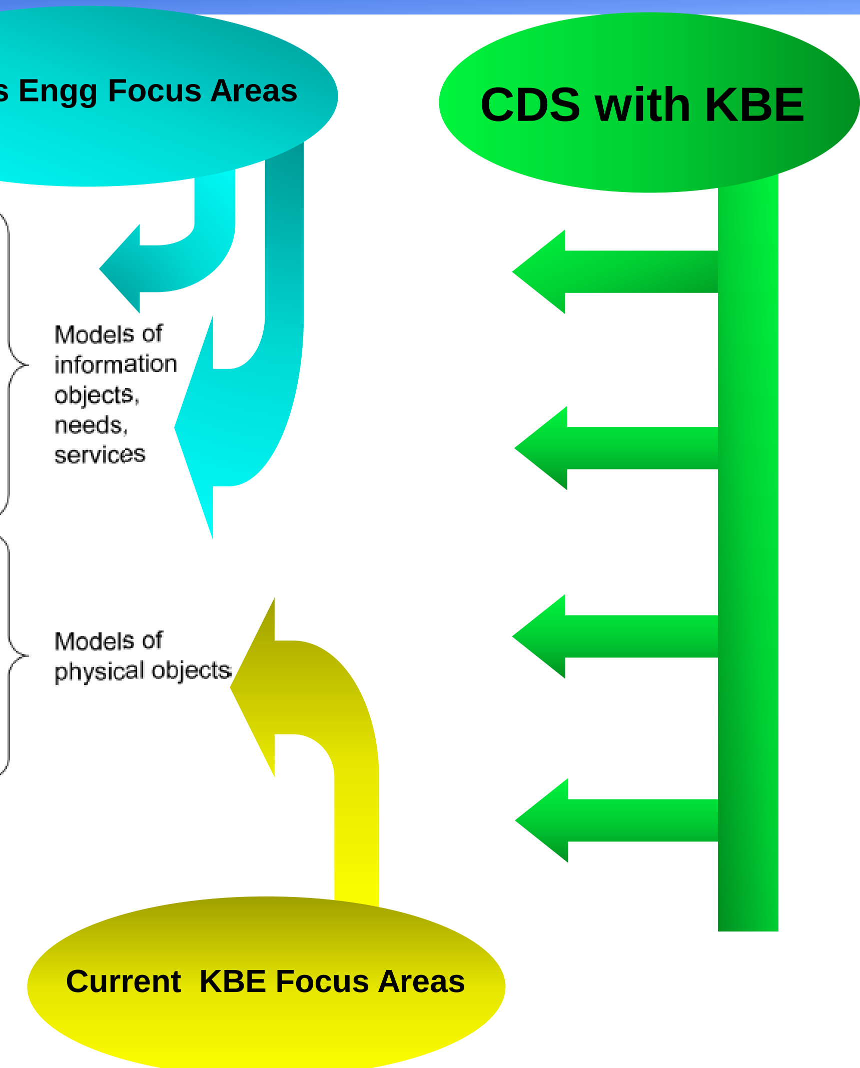
Requirements

Functional

Logical

Physical

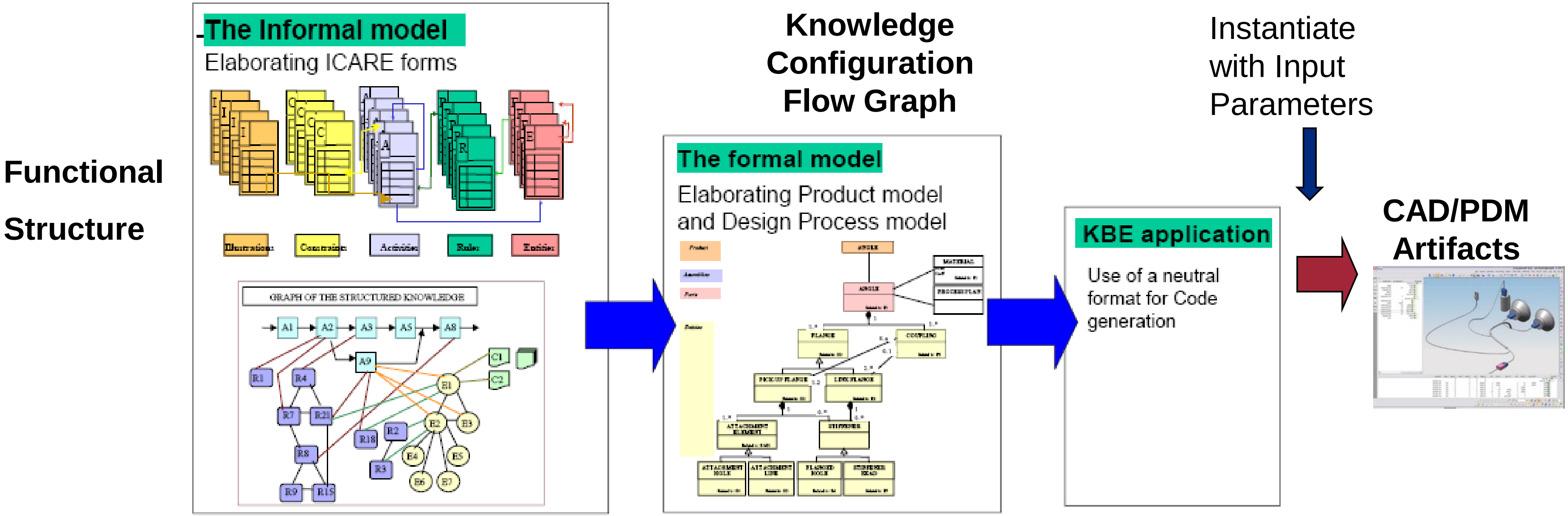




Allows Extension into Detailed Design Phases

Using MOKA ICARE forms to build Functional Structure Graphs

Global Product Data Interoperability Summit | **2018**



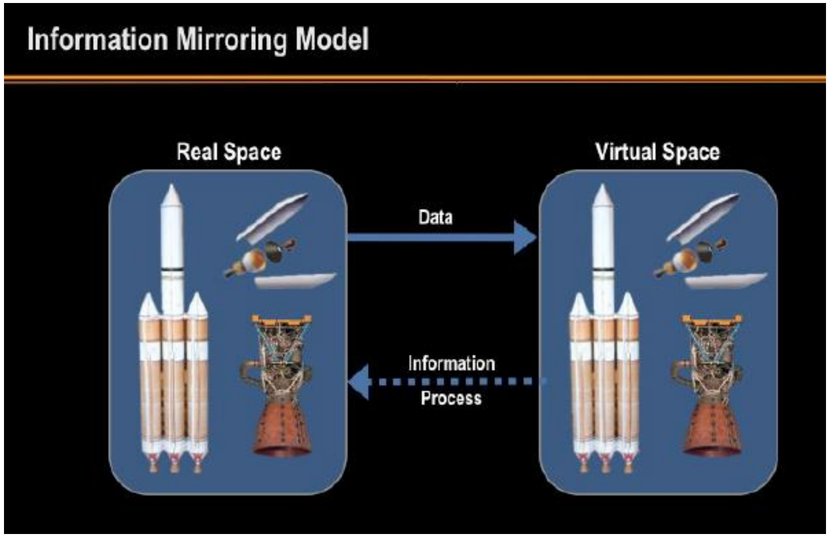
- **ICARE forms are templates that store the knowledge in five categories**:

- Illustrations– for describing any case studies or relevant examples
- Constraints– limitations on Entities
- Activities– the description of the elements of the design process
- Rules– the means of regulating the Activities and providing the ‘know-how’ or strategy of the design process
- Entities– the objects that describe the product (Entities may be further classified into E-structure and E-function)

Digital Twins: What are they good for?

Digital twins help manufacturers and OEMs by helping with:

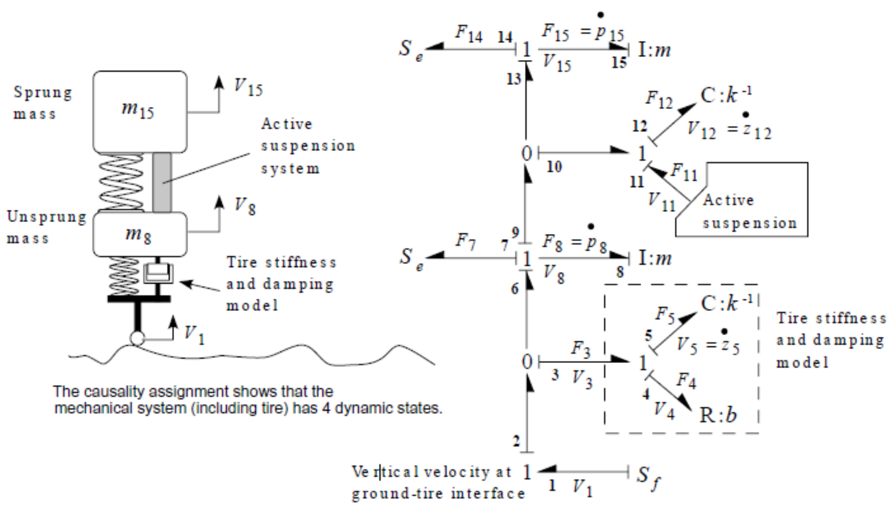
- Visualizing products in use, by real users, in real-time
- Building a digital thread, connecting disparate systems and promoting traceability
- Refining assumptions with predictive analytics
- Troubleshooting far away equipment
- Managing complexities and linkage within systems-of-systems
- Shared Conceptualization, Comparison, and Collaboration



Ref: [16]

Behavior Modeling and The Digital Twin

Global Product Data Interoperability Summit | **2018**



The causality assignment shows that the mechanical system (including tire) has 4 dynamic states.

$$M_s\ddot{Z}_s + C_s(\dot{Z}_s - \dot{Z}_{us}) + K_s(Z_s - Z_{us}) = 0 \qquad (1)$$

$$M_{us}\ddot{Z}_{us} + C_s(\dot{Z}_{us} - \dot{Z}_s) + K_s(Z_{us} - Z_s) + K_{us}(Z_{us} - Z_r) = 0 \qquad (2)$$

$$\ddot{Z}_s = \frac{1}{M_s}[C_s(\dot{Z}_{us} - \dot{Z}_s) + K_s(Z_{us} - Z_s)] \qquad (3)$$

$$\ddot{Z}_{us} = \frac{1}{M_{us}}[C_s(\dot{Z}_s - \dot{Z}_{us}) + K_s(Z_s - Z_{us}) + K_{us}(Z_r - Z_{us})] \qquad (4)$$

Let as assume the state variables are

$$Z_1 = Z_s - Z_w$$
$$Z_2 = \dot{Z}_s$$
$$Z_3 = Z_w - Z_r$$
$$Z_4 = \dot{Z}_w$$
$$\dot{Z}_1 = \dot{Z}_s - \dot{Z}_w \approx Z_s - Z_w$$
$$\dot{Z}_2 = \ddot{Z}_s$$
$$\dot{Z}_3 = \dot{Z}_w - \dot{Z}_r \approx Z_4 - \dot{Z}_r$$
$$\dot{Z}_4 = \ddot{Z}_w$$

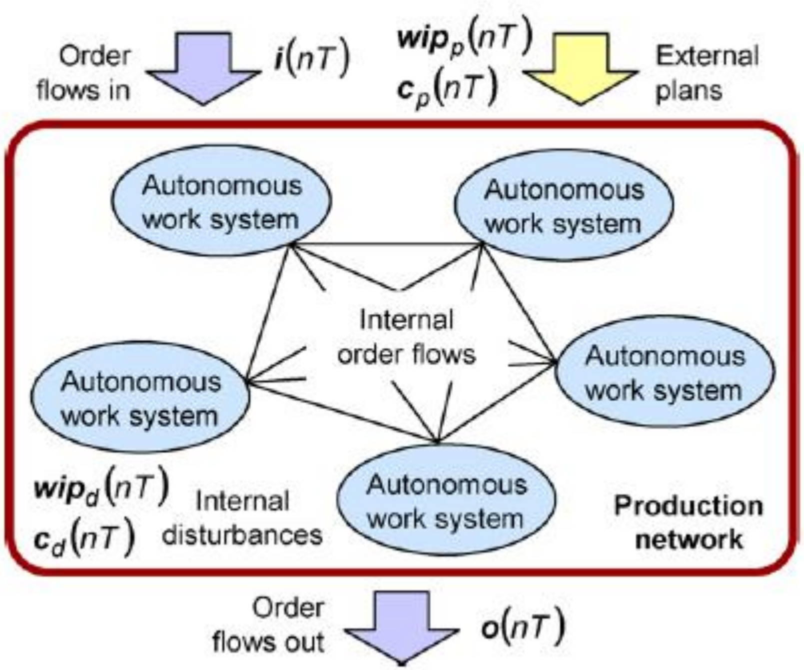
State Space Equations

$$\dot{Z} = AZ + BF_a + \dot{Z}_r$$

$$\begin{bmatrix} \dot{Z}_1 \\ \dot{Z}_2 \\ \dot{Z}_3 \\ \dot{Z}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & -1 \\ -K_s/M_{us} & -C_s/M_{us} & 0 & C_s/M_{us} \\ 0 & 0 & 0 & 1 \\ -K_s/M_s & C_s/M_s & -K_{us}/M_s & -C_s/M_s \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \end{bmatrix} \dot{Z}_r$$

Ref: [4]

Modeling Production Systems for Digital Twin



$$
\begin{bmatrix} z\mathbf{w}_i(z) \\ z\mathbf{w}_o(z) \\ z^d\mathbf{c}_m(z) \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & T\mathbf{P}^T \\ \mathbf{0} & \mathbf{I} & T\mathbf{I} \\ k_c\mathbf{I} & -k_c\mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{w}_i(z) \\ \mathbf{w}_o(z) \\ \mathbf{c}_m(z) \end{bmatrix} + \begin{bmatrix} T\mathbf{I} & \mathbf{0} & \mathbf{0} & T\mathbf{P}^T & -T\mathbf{P}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & T\mathbf{I} & -T\mathbf{I} \\ \mathbf{0} & k_c\mathbf{I} & -k_c\mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{i}(z) \\ \mathbf{w}_d(z) \\ \mathbf{wip}_p(z) \\ \mathbf{c}_p(z) \\ \mathbf{c}_d(z) \end{bmatrix}
$$

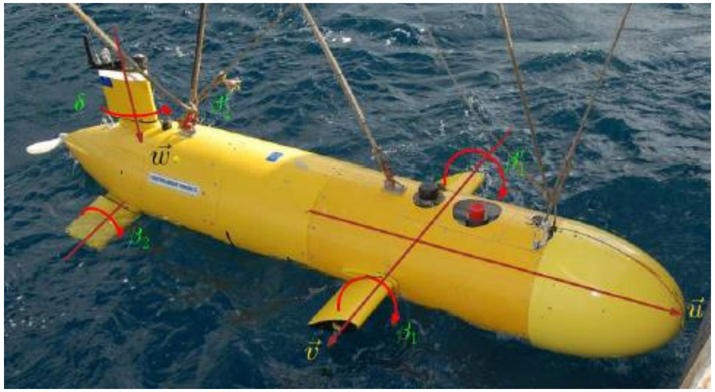
$$
\begin{bmatrix} \mathbf{w}_i(z) \\ \mathbf{w}_o(z) \\ \mathbf{c}_a(z) \\ \mathbf{wip}_a(z) \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{I} & -\mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{w}_i(z) \\ \mathbf{w}_o(z) \\ \mathbf{c}_m(z) \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & -\mathbf{I} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{i}(z) \\ \mathbf{w}_d(z) \\ \mathbf{wip}_p(z) \\ \mathbf{c}_p(z) \\ \mathbf{c}_d(z) \end{bmatrix}
$$

State Space Equations

Ref: [6]

Underwater Autonomous Vehicle Dynamics

Global Product Data Interoperability Summit | **2018**



Axial propeller to control the velocity in Ox direction and 5 independent mobile fins :

- 2 horizontals fins in the front part of the vehicle (b1, b01).
- 1 vertical fin at the tail of the vehicle (d).
- 2 fins at the tail of the vehicle (b2, b02).

State Space Equations

Ref: [17]

Physical model:

$$M\dot{\nu} = G(\nu)\nu + D(\nu)\nu + \Gamma_g + \Gamma_u$$

$$\dot{\eta} = J_c(\eta_2)\nu$$

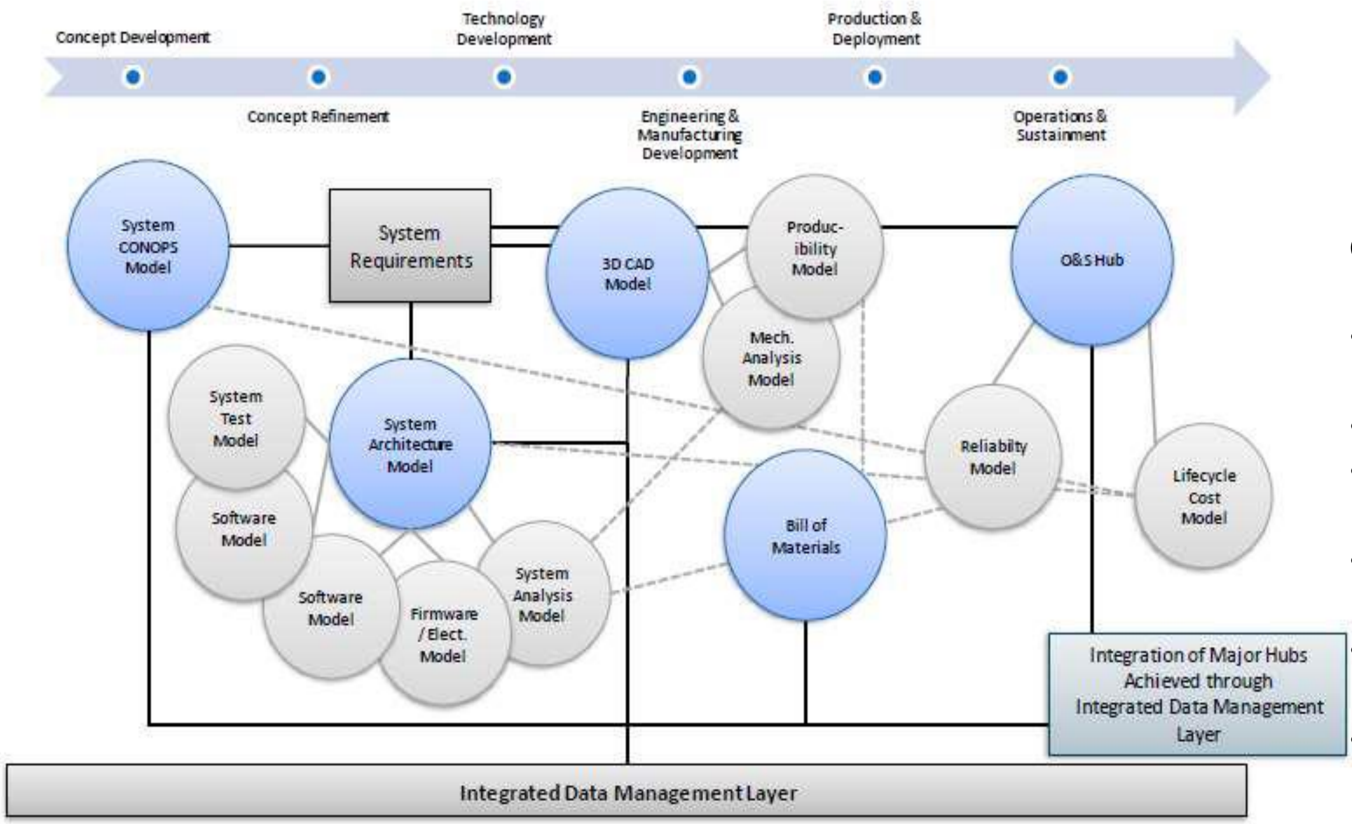
where:

- M: mass matrix: real mass of the vehicle augmented by the "water-added-mass" part,
- $G(\nu)$: action of Coriolis and centrifugal forces,
- $D(\nu)$: matrix of hydrodynamics damping coefficients,
- Γ_g: gravity effort and hydrostatic forces,
- $J_c(\eta_2)$: referential transform matrix,
- Γ_u: forces and moments due to the vehicle's actuators.

A 12 dimensional state vector : $X = [\eta(6) \quad \nu(6)]^T$.

- $\eta(6)$: position in the inertial referential: $\eta = [\eta_1 \quad \eta_2]^T$ with $\eta_1 = [x \quad y \quad z]^T$ and $\eta_2 = [\phi \quad \theta \quad \psi]^T$. x, y and z are the positions of the vehicle , and ϕ, θ and ψ are respectively the roll, pitch and yaw angles.
- $\nu(6)$: velocity vector, in the local referential (linked to the vehicle) describing the linear and angular velocities (first derivative of the position, considering the referential transform: $\nu = [\nu_1 \quad \nu_2]^T$ with $\nu_1 = [u \quad v \quad w]^T$ and $\nu_2 = [p \quad q \quad r]^T$

Need for Low fidelity Behavioral Models for Digital Twins



Ref: [1]

Models needed everywhere, but,

- What if you don't have one from first principles?
- Are they the right fidelity?
- How do you address all disciplines?
- How do you keep them up to date?
- What about the data deluge with IOT?
- Is learning and adaptation built into the models over its lifecycle?
- Do the models apply uniformly to different levels of abstraction?

Generic Dynamic Models

Global Product Data Interoperability Summit | **2018**

State Equations

Black Box Model

Feedback / Learning

$$x_{k+1} = f(x_k, u_k) + w_k$$

$$y_k = h(x_k) + v_k \text{ for } k = 0, 1, 2, \cdots.$$

Sensor 3

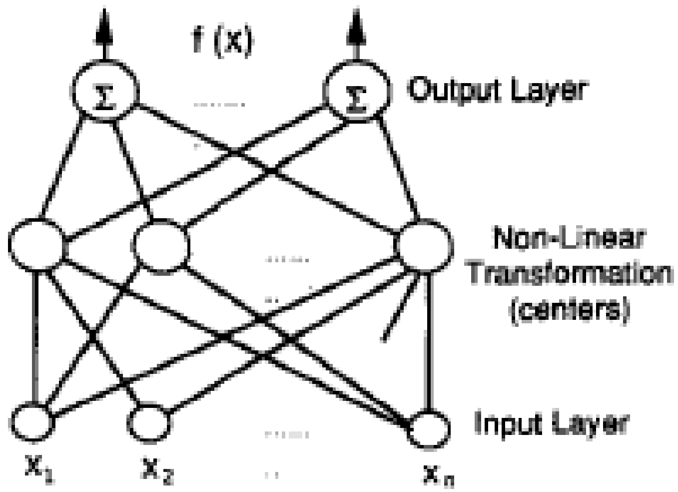
Sensor 1

Sensor 4

Sensor 2

Black Box Neural Networks Models

Radial Basis Function Networks



$$t_p(X) = \sum_{j=1}^{p} \lambda_j^p \Phi(\|X - X_j^c\|) + \lambda_0^T X$$

$$|t_p(X) - F(X)| \rightarrow 0$$

$$\Phi(r) = \begin{cases} r^2 \log r & \text{Duchon} \\ \sqrt{c + r^2} & \text{Hardy} \\ r^l \ (l \text{ an odd integer}) & \text{Thin Plate} \\ \exp(-r^2) & \text{Schagen} \end{cases}$$

$$x_{k+1}^i = [\Lambda \ \Lambda_0] \begin{bmatrix} \Psi(X_k^i) \\ X_k^i \end{bmatrix} + w_k^i$$

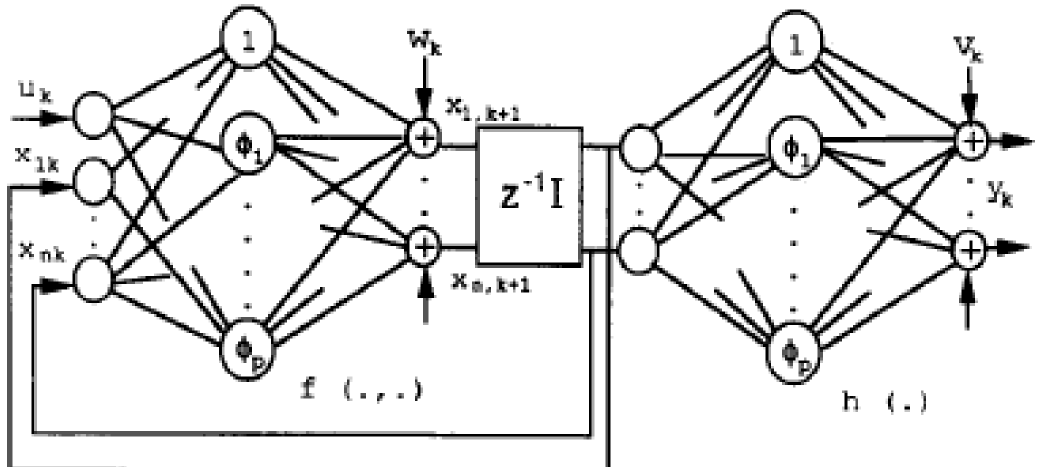
$$z_k^i = x_k^i + \zeta_k^i$$

Ref: [5]

Approximate System Equations

Training the Network

Global Product Data Interoperability Summit | **2018**



Training / System Identification

$$\hat{\Theta}_{N+1} = \hat{\Theta}_N + R_N \hat{\xi}^T_{N-1} \left[I + \hat{\xi}_N R_N \hat{\xi}^T_{N-1} \right]^{-1} \times \left[\gamma_{N+1} - \hat{\xi}_N \hat{\Theta}_N \right]$$

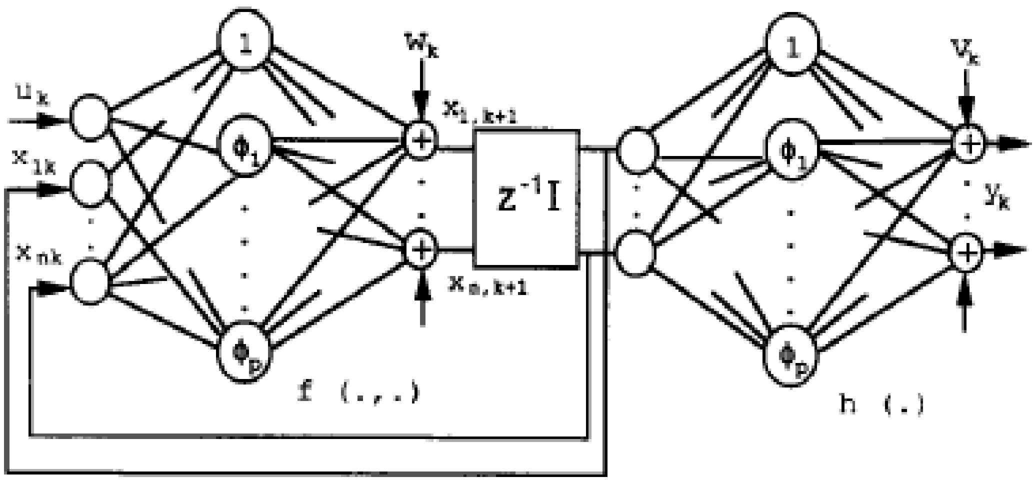
$$R_{N+1} = R_N - R_N \hat{\xi}^T_{N-1} \left[I + \hat{\xi}_N R_N \hat{\xi}^T_{N-1} \right]^{-1} \hat{\xi}_N R_N$$

Ref: [5]

Using the Learned system

Global Product Data Interoperability Summit | **2018**

State Estimation Using Learned System



$$\hat{x}(k+1) = f'(\hat{x}_k, u_k) + F\hat{x}_k + bu_k + K_k[y_k - h'(\hat{x}_k) - H\hat{x}_k]$$

$$K_k^* = (2+e_f)F\hat{P}_k H^T\left[(2+e_f)H\hat{P}_k H^T + V\right]^{-1}$$

$$\hat{P}_{k+1} = l_1(F - K_kH)\hat{P}_k(F - K_kH)^T + l_2I + l_3Tr(\hat{P}_k)I + l_4K_kK_k^T + l_5Tr(\hat{P}_k)K_kK_k^T + W + K_kVK_k^T$$

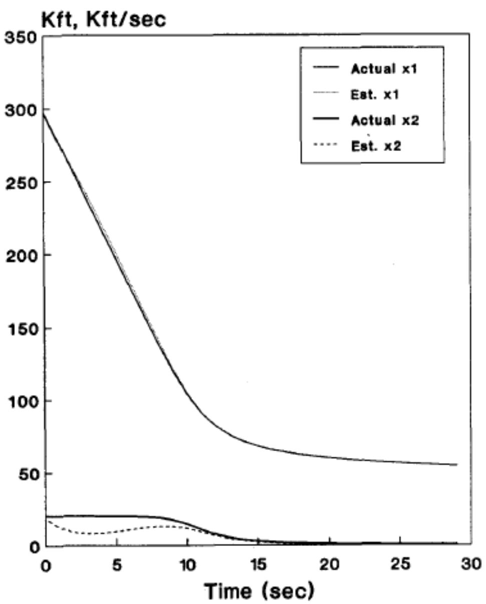
$$\hat{P}_0 = \rho_0.$$

Ref: [5]

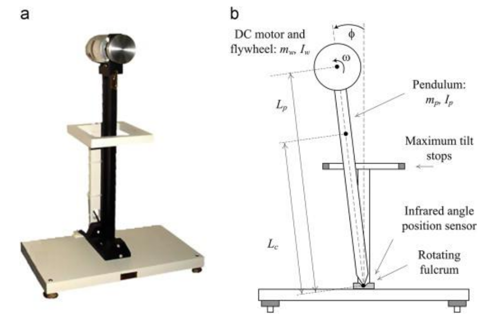
Viability of Learned Network for Modeling Dynamic Systems

$$\begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \end{bmatrix} = \begin{bmatrix} x_{1k} - Tx_{2k} \\ x_{2k} - 3Tx_{2k}^2 e^{-0.05x_{1k}} \end{bmatrix}$$

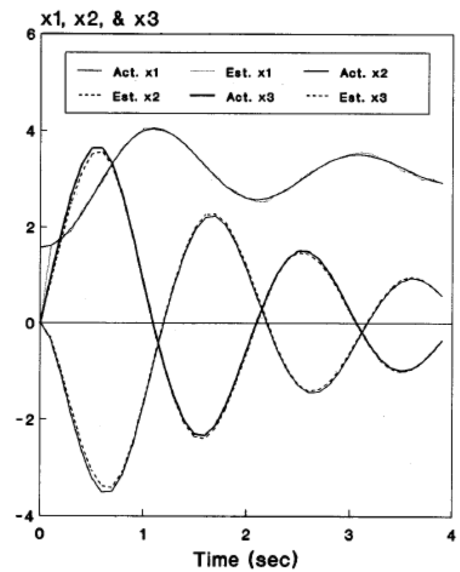
$$y_k = \sqrt{10000 + x_{1k}^2} + \nu_k$$



$$\begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \\ x_{3,k+1} \end{bmatrix} = \begin{bmatrix} x_{1k} + Tx_{2k} \\ x_{2k} + T(9.8 \sin x_{1k} + x_{3k}) \\ x_{3k} - 10T(x_{2k} + x_{3k}) \end{bmatrix}$$



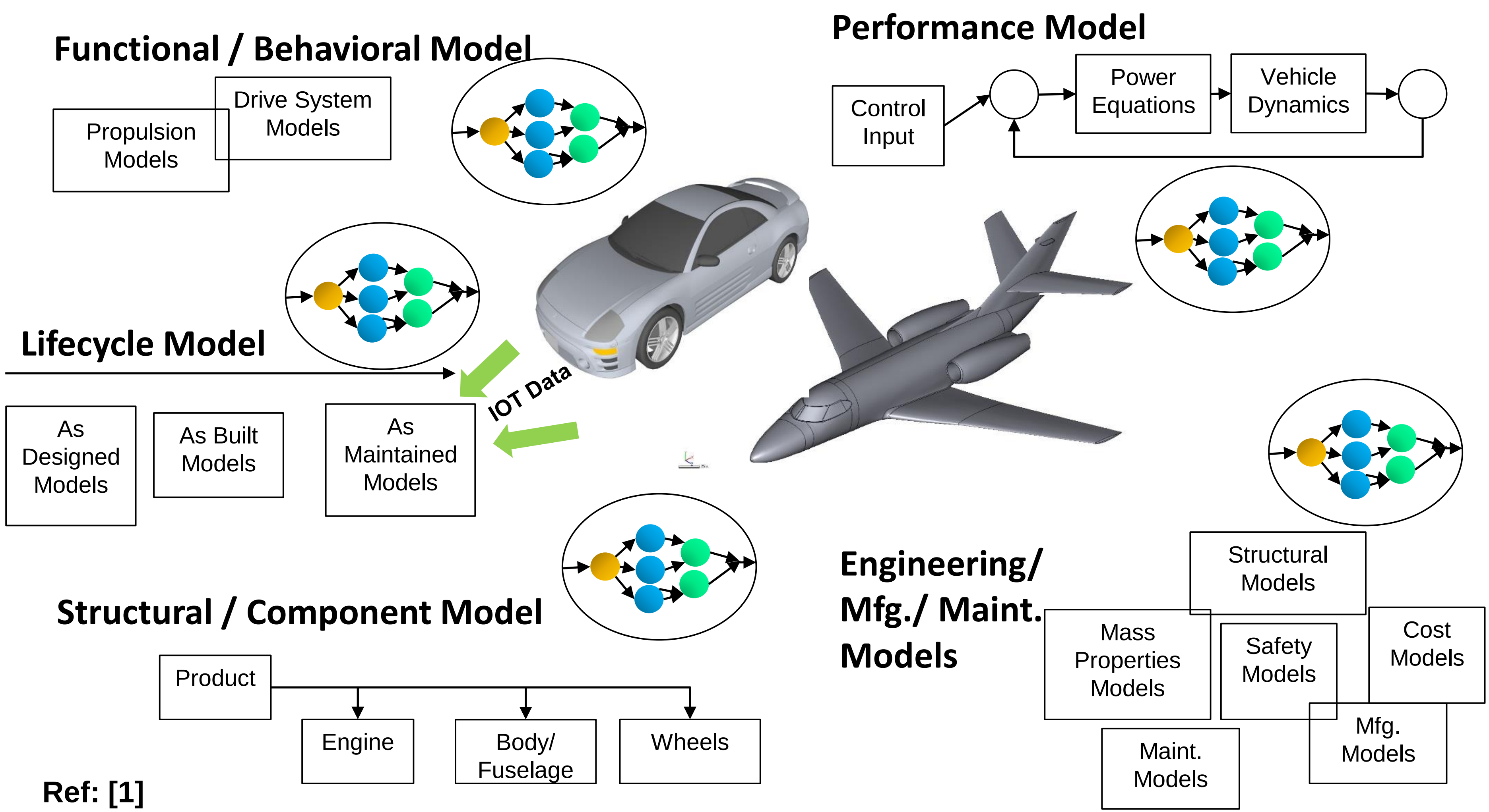
Inverted Pendulum



Ref: [5]

Learning Dynamic Black Box models

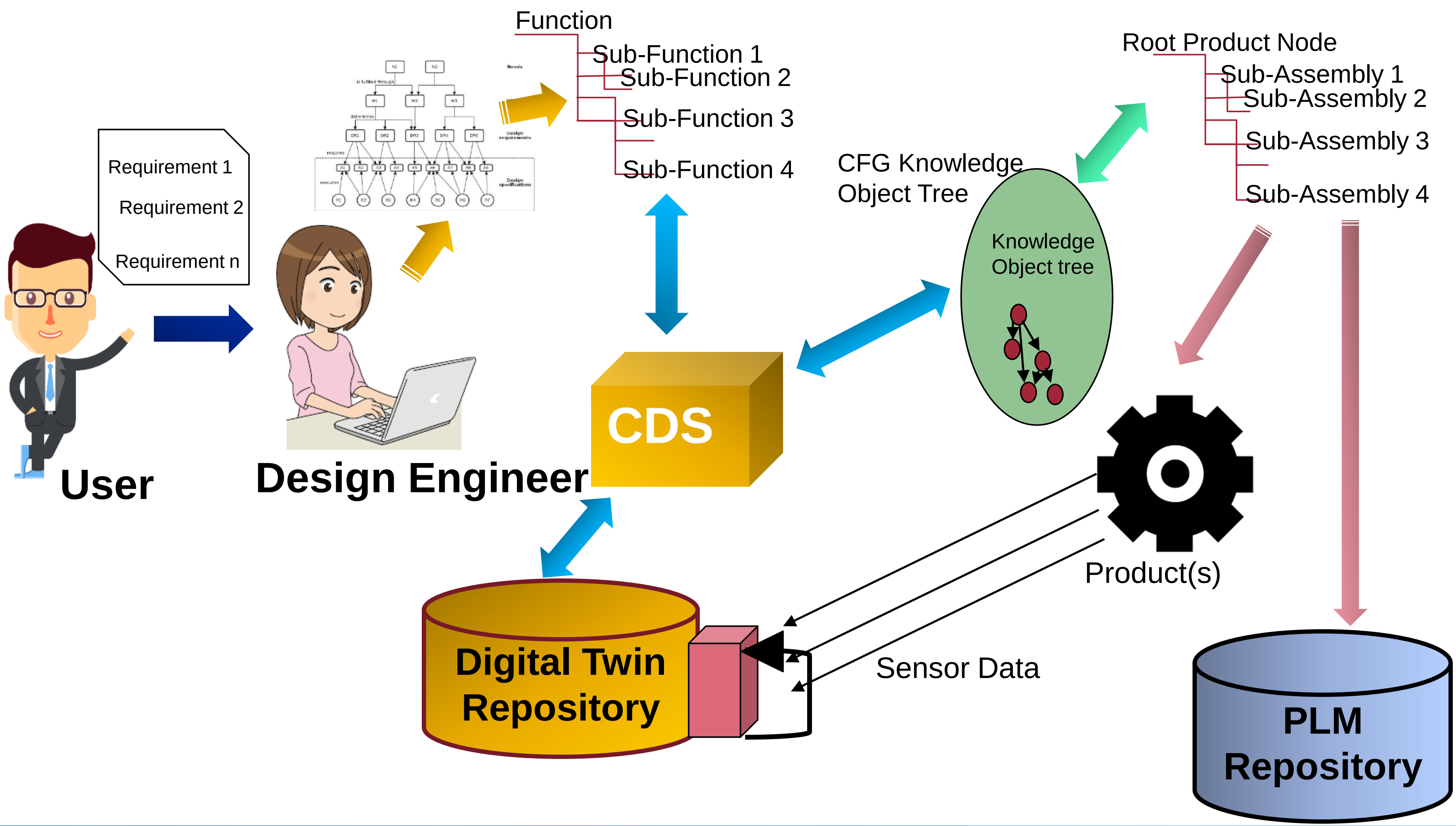
Global Product Data Interoperability Summit | **2018**



Ref: [1]

Proposed Workflow for CDS with Digital Twin

Global Product Data Interoperability Summit | **2018**



Questions?

Global Product Data Interoperability Summit | **2018**

References

1. "Introduction to Model-Based Systems Engineering (MBSE) and SYSML," Laura E. Hart, Delaware Valley INCOSE Chapter Meeting, July 30, 2015
2. "Knowledge Based Systems Engineering," Sunil Elanayar, BTEC 2011, 2011.
3. "The evolution, challenges, and future of knowledge representation in product design systems," S.K. Chandrasegaran et al., Computer Aided Design, 2013.
4. "Mathematical modelling and simulation quarter car vehicle suspension," P. Sathishkumar et al., International Journal of Innovative Research in Science, Engineering and Technology, *Volume 3, Special Issue 1, February 2014.*
5. "Radial Basis Function Neural Network for Approximation and Estimation of Nonlinear Stochastic Dynamic Systems," Sunil Elanayar V.T. and Yung C. Shin, IEEE Transactions on Neural Networks, Vol. 5, No. 4, July 1994.
6. "Dynamic modeling of production networks of autonomous work systems with local capacity control," N.A. Duffie et al., CIRP Annals - Manufacturing Technology 57, 2008.
7. "Object Oriented Graph Grammars for Computational Design Synthesis," Bergen Helms, Ph.D. Dissertation, TECHNISCHE UNIVERSITÄT MÜNCHEN, 2012.
8. "Deep Learning," Yann LeCunn et al., Nature, Vol 521, May 2015.
9. "A Framework for Computational Design Synthesis based on Graph-Grammars and Function-Behavior-Structure," Bergen Helms, Kristina Shea, and Frank Hoisl, ASME DETC, San Diego, 2009.
10. "Design Repositories: Engineering Design's New Knowledge Base," Simon Szykman et al., IEEE Intelligent Systems, 2000.
11. "A framework for computer-aided conceptual design and its application to systems architecting of mechatronic products," Hitoshi Komoto and Tetsuo Tomiyama, Computer-Aided Design, Vol 44, 2012.
12. "Computer-Based Design Synthesis Research: An Overview," Amaresh Chakrabarti et al., Journal of Computing and Information Science in Engineering, Vol 11, 2011.
13. "A Framework for Computational Design Synthesis: Model and Applications", Cagan, J. et al., Journal of Computing and Information Science in Engineering, Vol. 5, 2005.
14. "A Graph Grammar Based Framework for Automated Concept Generation," Kurtoglu, T.; Campbell, M.I., 9th International Design Conference, DESIGN 2006.
15. "Engineering Design: A Systematic Approach," Pahl, G.; Beitz, W.; Feldhusen, J.; Grote, K.-H, 3rd edition, London, 2007.
16. "Digital Twin: Manufacturing Excellence through Virtual Factory Replication," Michael Grieves, Digital Twin White Paper, 2014.
17. "Modelling, analysis and control of linear systems using state space representations," Olivier Sename, Grenoble INP / GIPSA-lab, February 2018.