How to design a tool for designers?

System-level feature-based modeling of cyber-physical systems

Shahab Pourtalebi

TU Delft

Shahab Pourtalebi

Experience

INSTRUCTOR 2018 - Present
Sheridan College

RESEARCHER / INSTRUCTOR 2012 - 2017
Delft University of Technology

FACULTY MEMBER 2006 – 2012
Tabriz IA University

HEAD OF DEPARTMENT 2008 – 2011
Tabriz IA University

DESIGNER / PROJECT MANAGER 2004 – 2006
NAK design studio

DESIGNER / USER RESEARCHER 2000 – 2004
PARS ZARASA household design and industrial company

Education

PhD 2012 – 2017

Design Engineering department, IDE, TU Delft

- Topic: System-level feature-based modeling of Cyber-Physical Systems

MASTER's DEGREE 2003 - 2006

Art University of Tehran

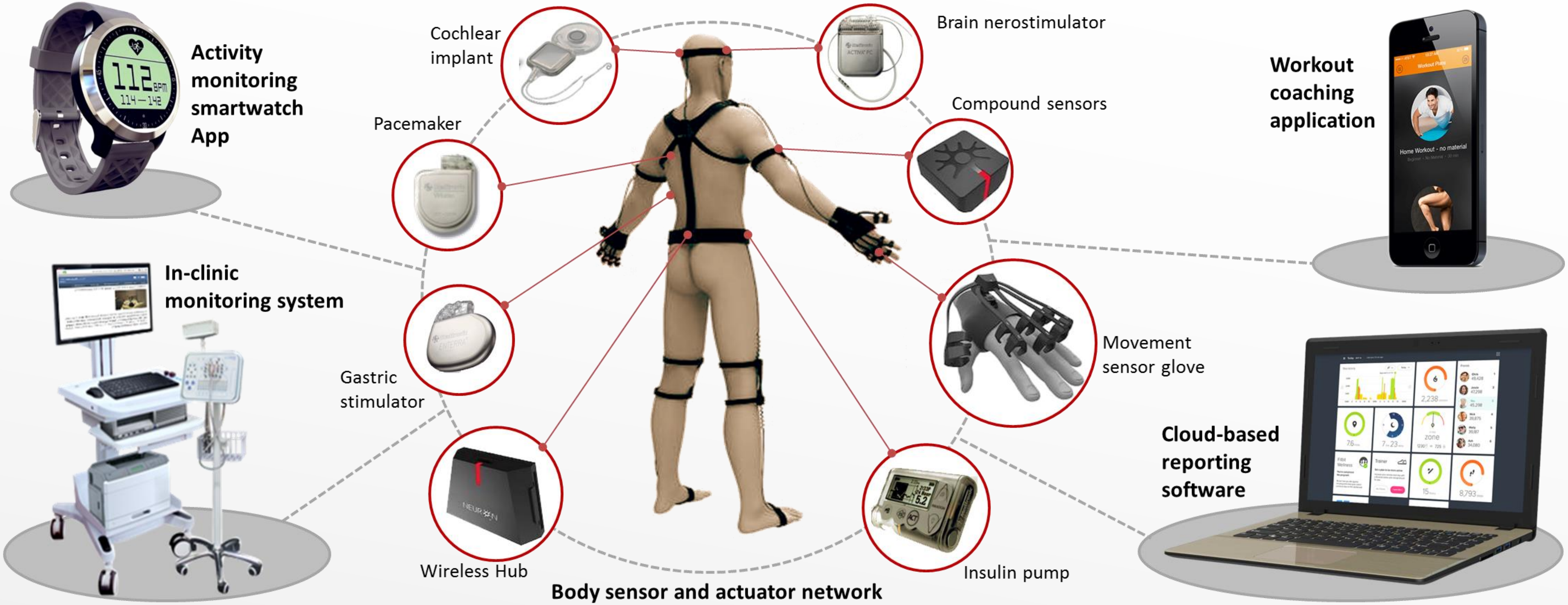
- Major: Industrial design engineering
- Summa cum laude

BACHELOR's DEGREE 1999 - 2003

Sahand University of Technology

- Major: Industrial design engineering
- Summa cum laude

Cyber-Physical System (CPS)



Characteristics of CPSs

Distributed integrity: distributed cooperative problem solving, in harmony with the techno-socio-economic environment

Openness: functional and structural openness (with blurred overall system boundaries)

Dynamicity: capability to dynamically change behaviors, and reconfigure internal structure

Source multiplicity: knowledge-intensive components are able to handle different information sources

Goal-oriented cooperation: components may operate according to different problem solving strategies while achieving the overall objective of the system

Autonomy: memorizing and learning from history in an unsupervised manner by smart agents

Proactivity: non-planned functional interactions, and acting proactively in emergent internal and external situations

Strategy variability, **Integrity,** **Heterogeneity,**

Real-time processing, **Situated reasoning,** **Reproductive intelligence**

Paradigmatic features	Ordinary systems	Low-end CPSs	High-end CPSs
System complexity	Ordinarily compound	Linearly complex	Non-linearly complex
Functional objectives	Providing means	Providing utilities	Providing services
Overall architecture	Closed system boundary	Extendable system boundary	Open system boundary
Construction topology	Predefined fix architecture	Predefined modifiable architecture	Runtime dynamically architected
Functional organization	Integrated structure	Distributed structure	Decentralized structure
Functional connectivity	Stand alone	Static networked	Dynamically networked
Operating principles	Execution and control	Sensing, computing and activating	Observing, reasoning, activating and motivating
Functional intelligence	Functionally determined	Functionally smart with limited awareness	Functionally autonomous and conscious
Readiness for operation	Condition activated	Alert in operation	Proactive in operation
Intelligent components	None or dedicated problem solvers	Individual agent-based reasoning	Cooperating agent swarm reasoning
System control	Manually controlled, Feedback controlled	Adaptive and self-adaptive control	Intelligent control according to goal driven operational scenarios
Placement in context	Largely independent of context	Context sensitive (situated) operation	Context driven operation
Adaptability	Partial external modifiability	Self-adapting capabilities	Self-evolving capabilities
Heterogeneity	Mono-disciplinary system	Multi-disciplinary system	Trans-disciplinary system

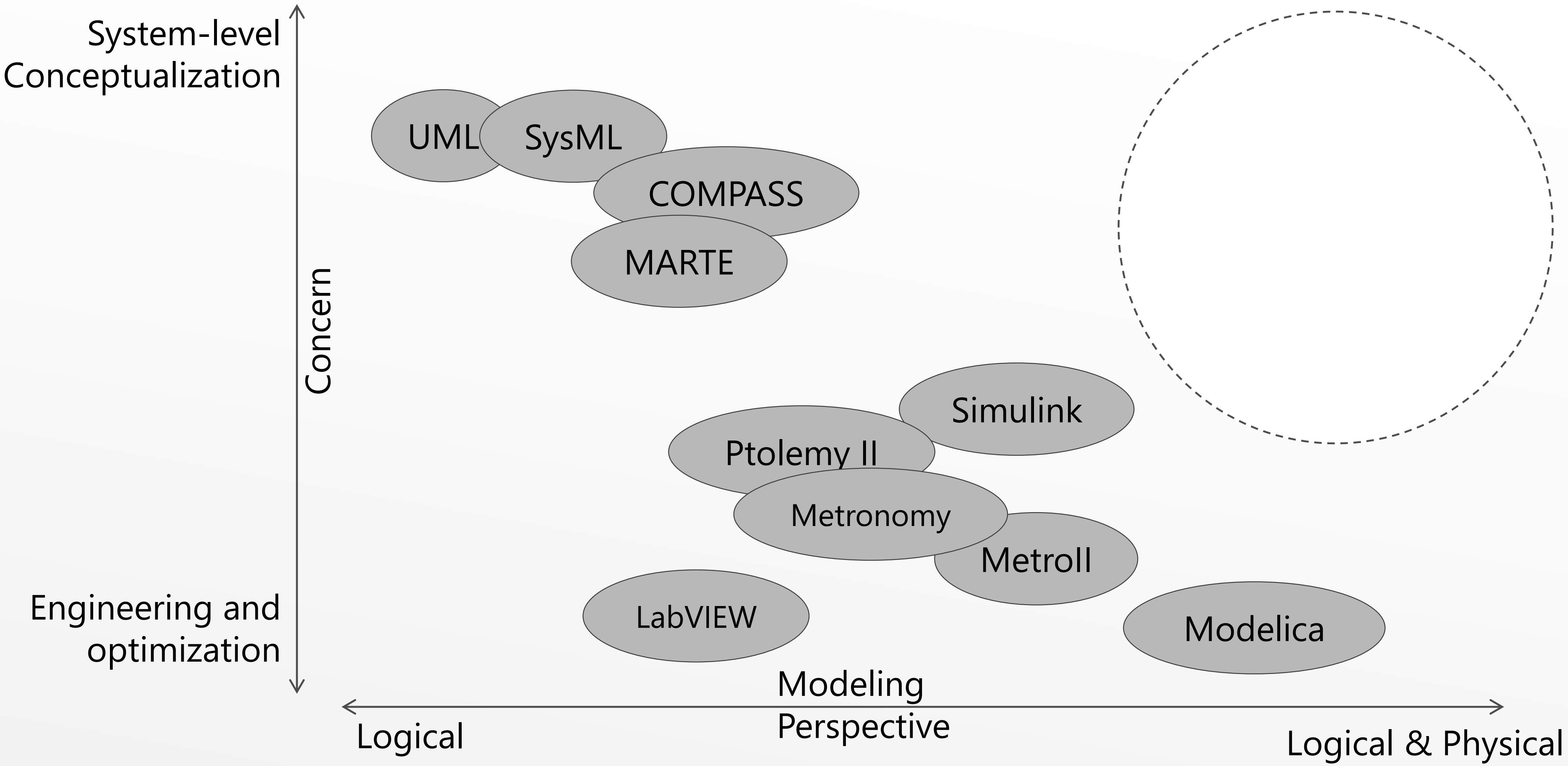
Shortcomings of the existing tools

- **Composability** concerns remained unsolved
- Are not suitable for **system-level design**
- Apply **too much simplification**
- Ignore **physical attributes**

Overview of the research

- **The research problem** — There is no suitable tool …
- **Initiating a solution** — A theoretical framework
- **Developing the idea** — Methodological fundamentals
- **Feasibility test** — Warehouses and procedures
- **Validation** — Benchmarking against other tools

Landscape of tools



Theoretical framework

Mereo-operandi theory

Architecture

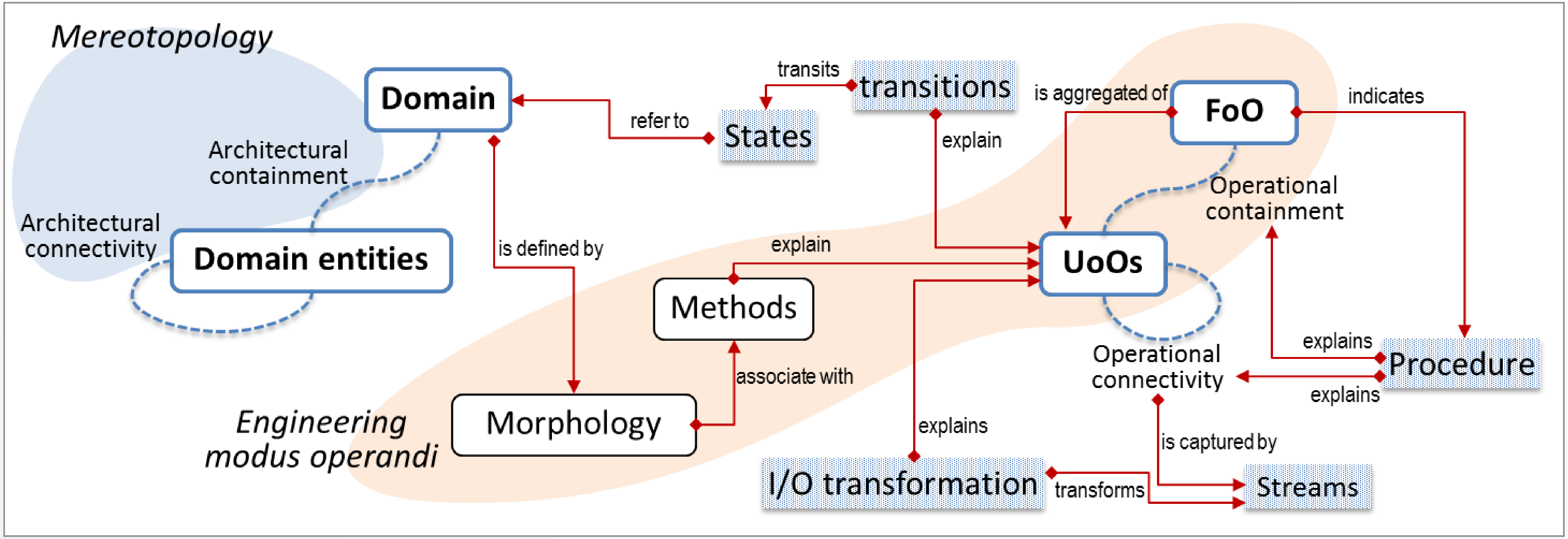
Mereotopology

Operations

Engineering modus operandi

Theoretical framework

Mereo-operandi theory

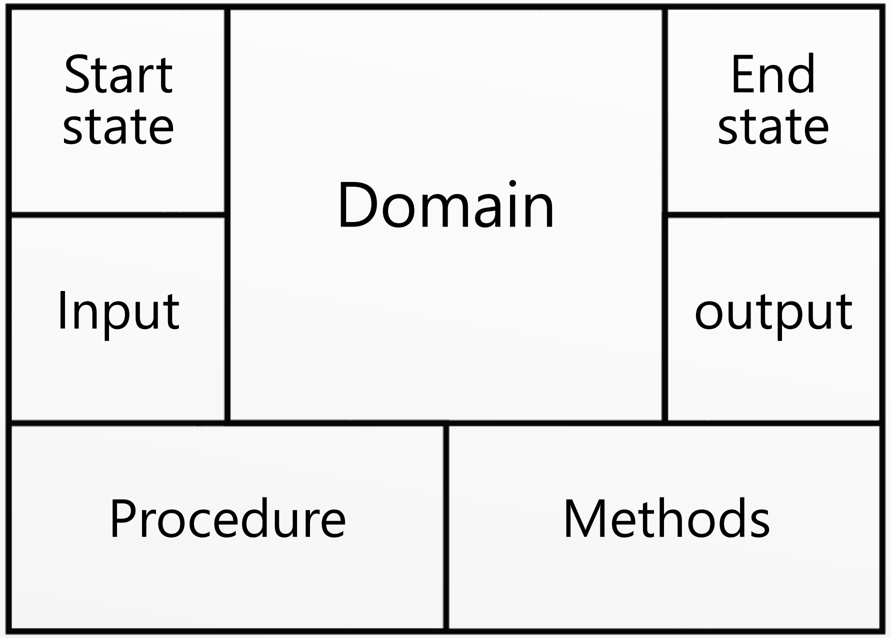


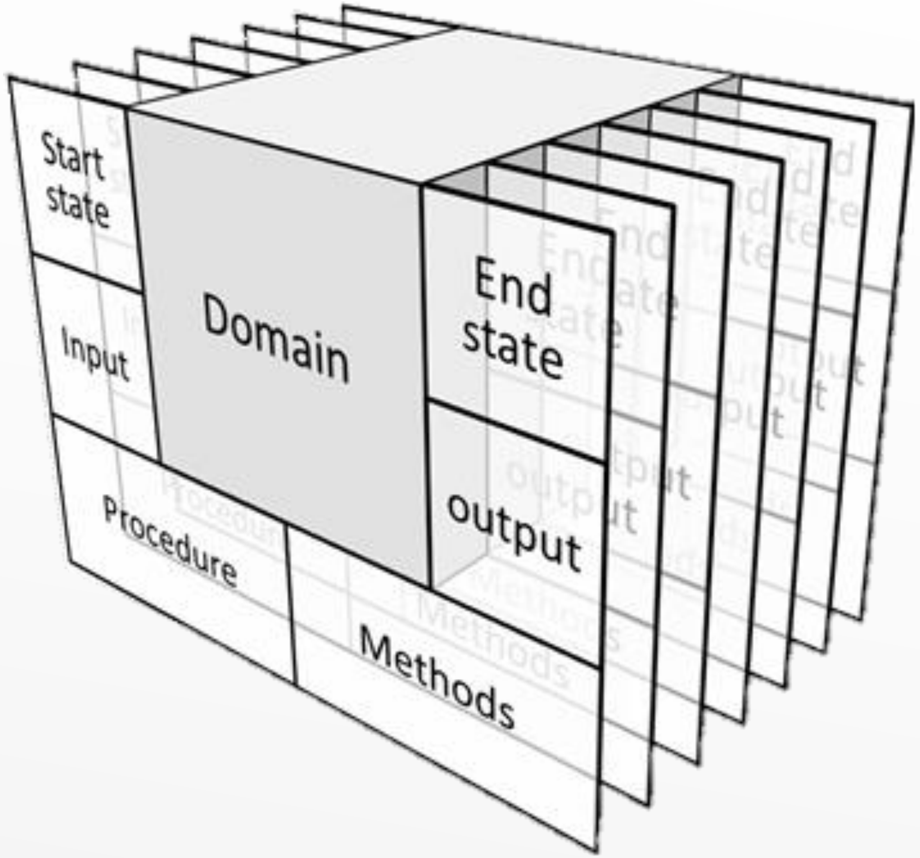
Theoretical framework

Mereo-operandi theory

1. System manifestation feature (SMF)
2. Operation and architecture concurrently and linked together
3. A uniform structure
4. Multi-granularity

System Manifestation Feature (SMF)

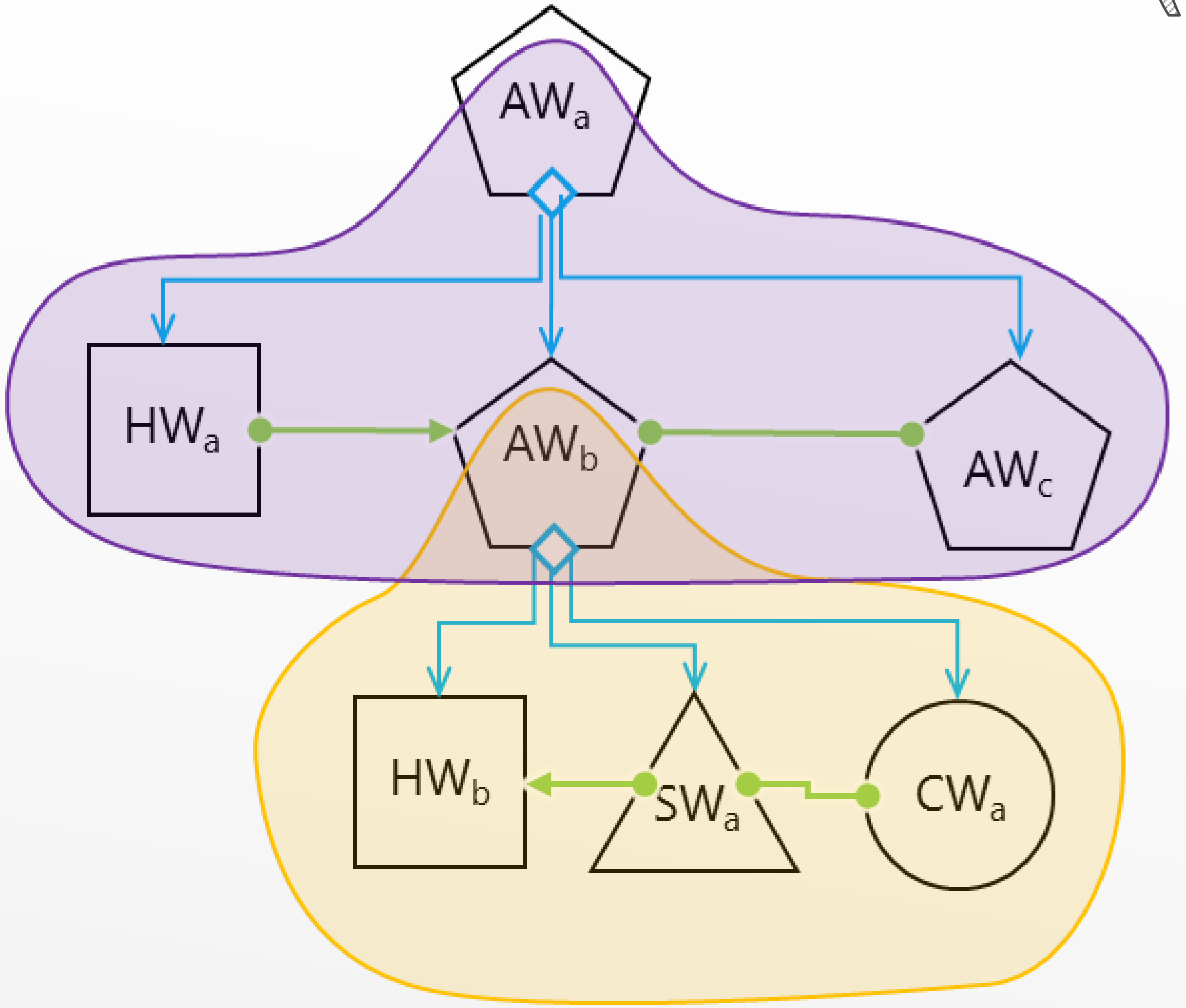




System Manifestation Feature (SMF)

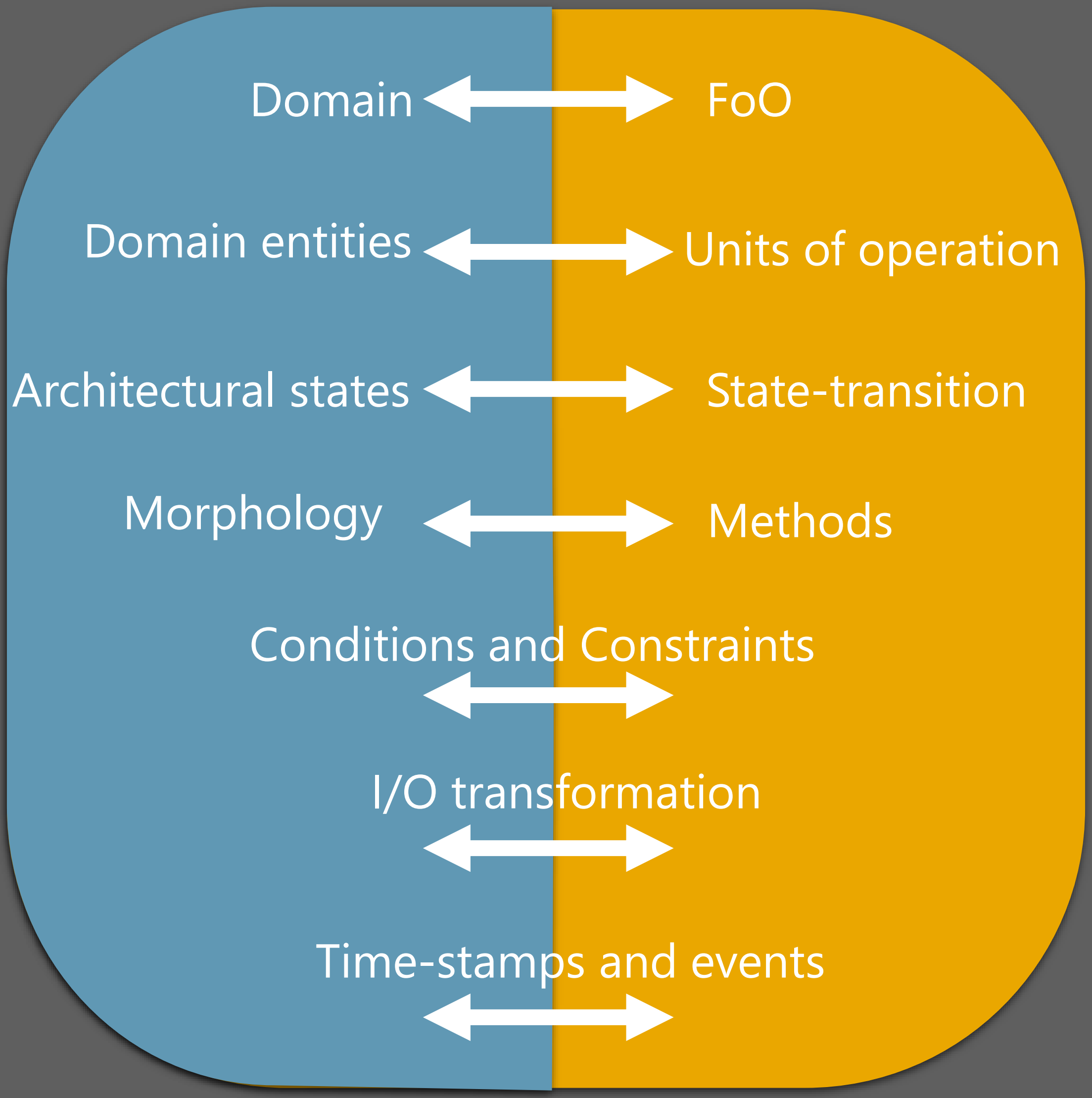
- A kind of system-level feature
- Reusable modeling entity
- Representing components (SW, HW, CW, AW) uniformly
- Captures both operation- and architecture- related info chunks, as well as their relations
- Supports multi-granularity
- Language independent
- Semantic-rich

System Manifestation Feature (SMF)



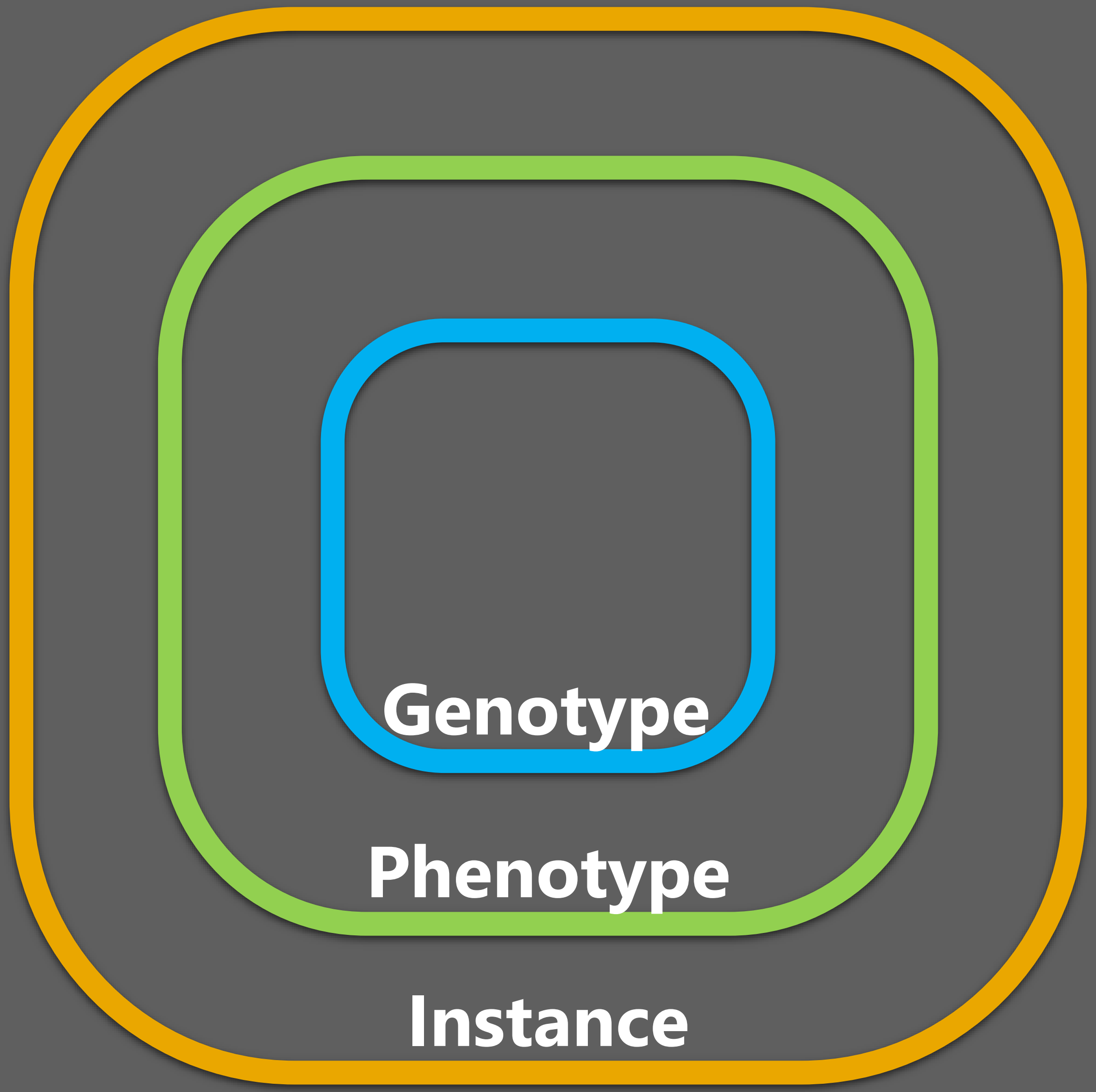
SMF

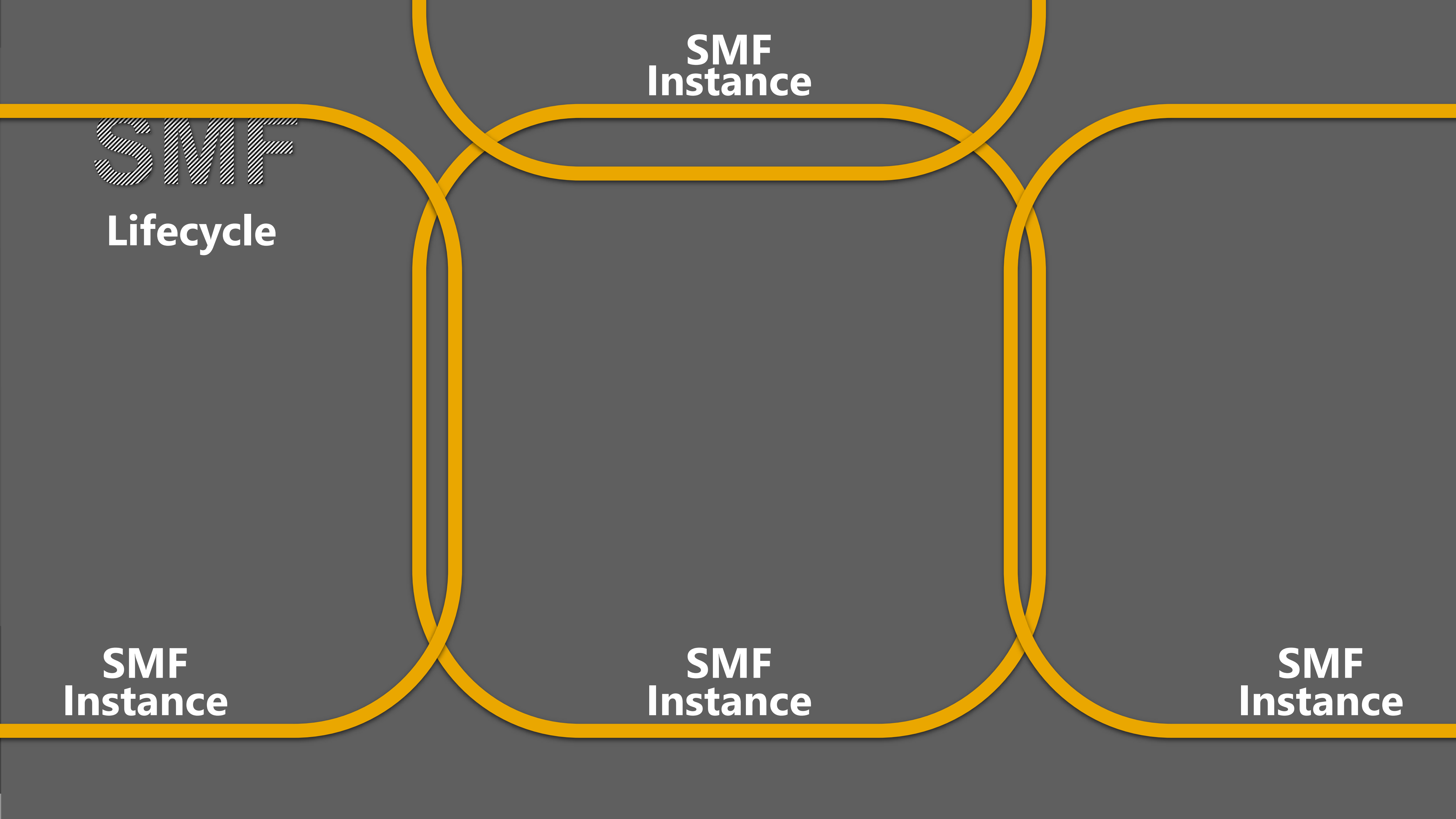
Architecture **Operation**



SMF

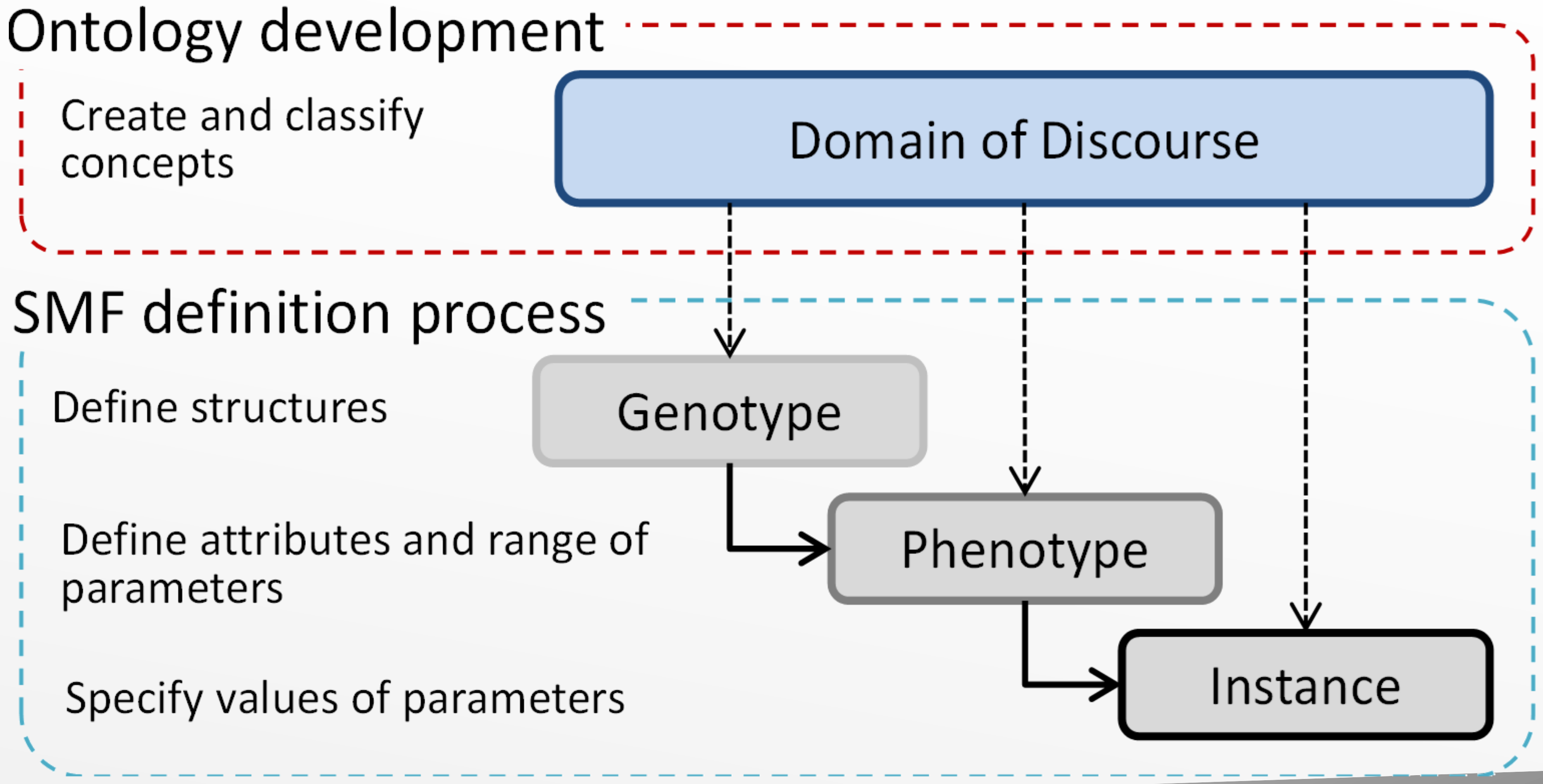
Lifecycle




SMF
Lifecycle
SMF Instance
SMF Instance
SMF Instance
SMF Instance

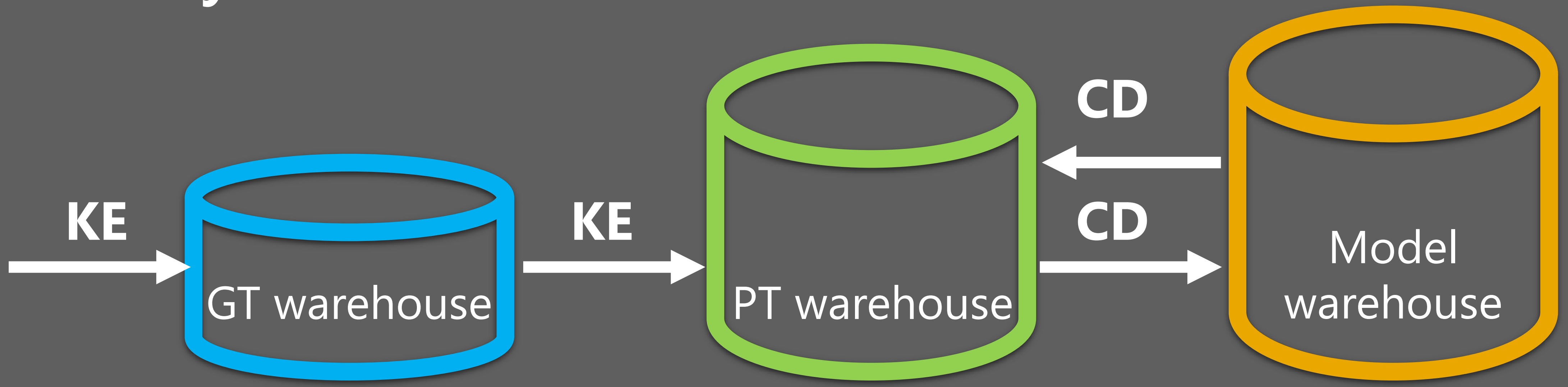
SMF

Lifecycle

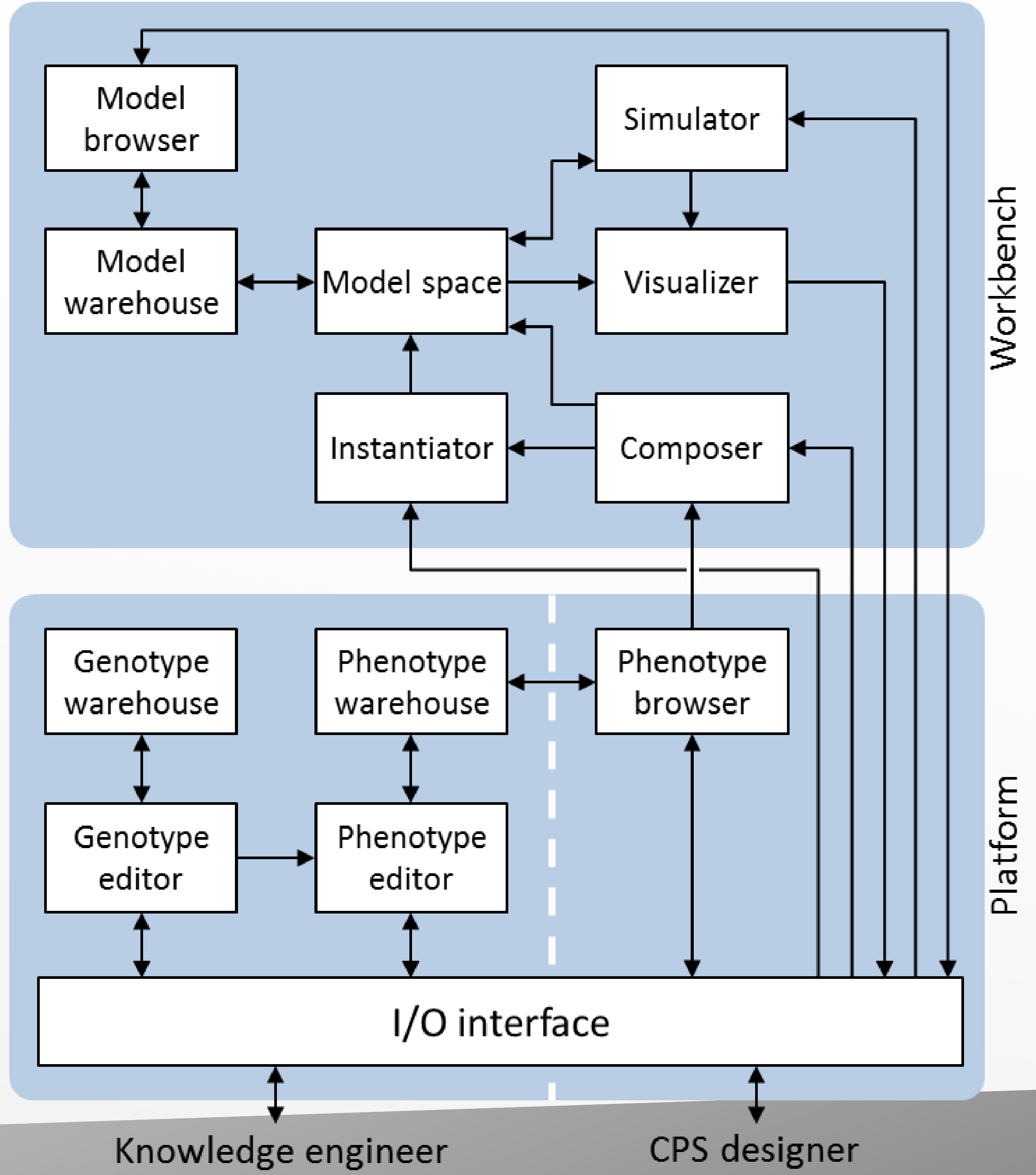


SMF

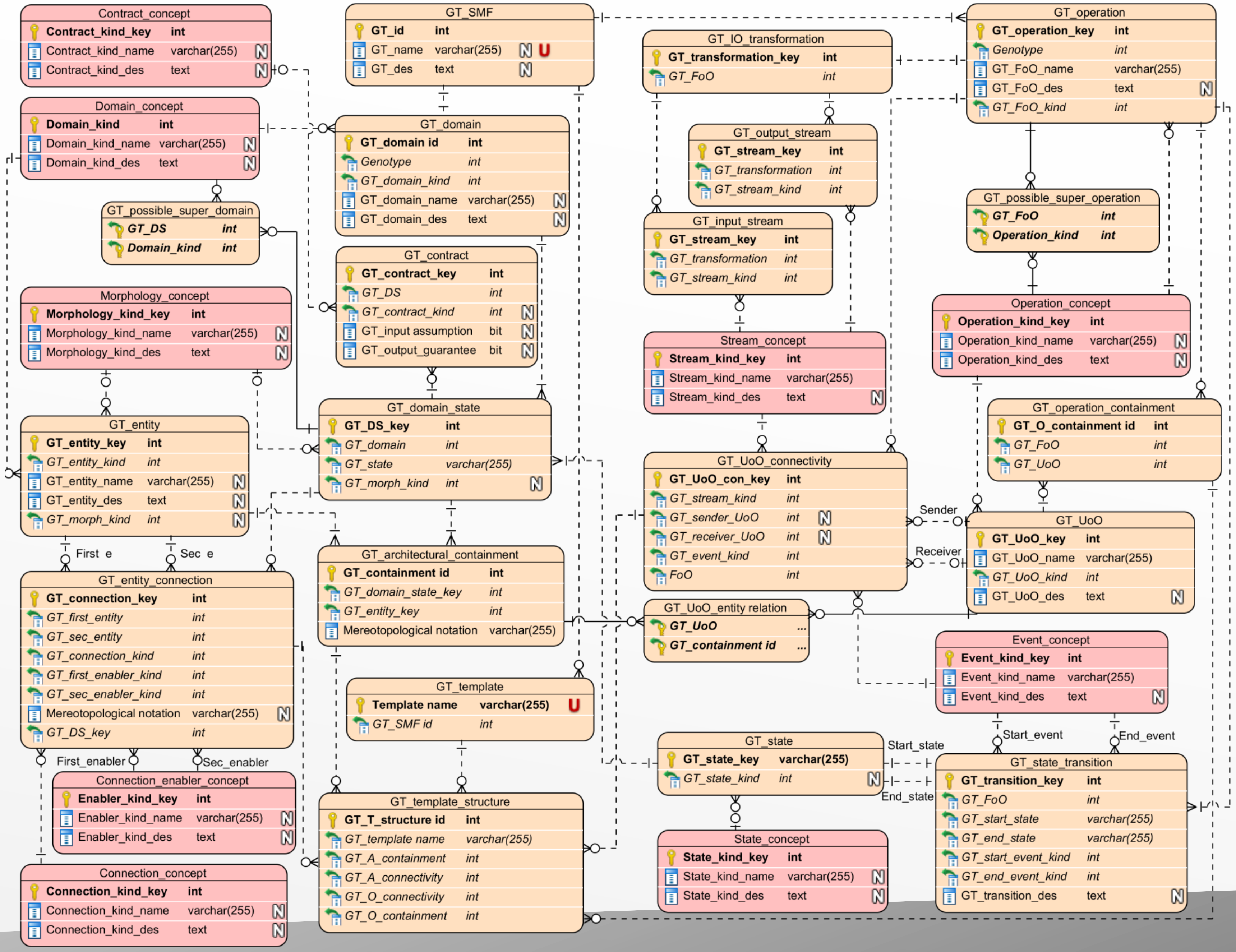
Lifecycle



SMF tool box

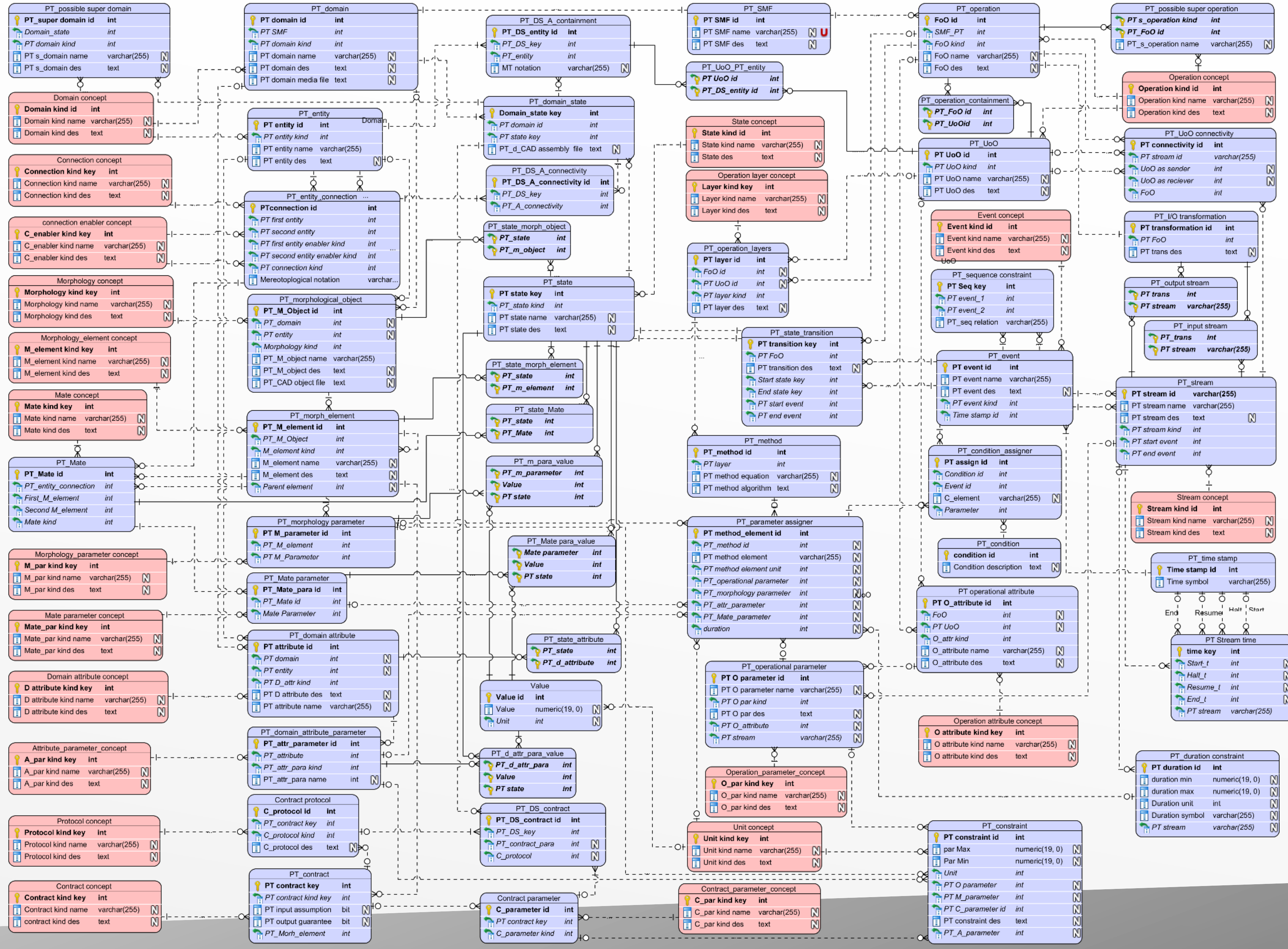

Model browser
Simulator
Model warehouse
Model space
Visualizer
Instantiator
Composer
Workbench
Genotype warehouse
Phenotype warehouse
Phenotype browser
Genotype editor
Phenotype editor
Platform
I/O interface
Knowledge engineer
CPS designer

GT database

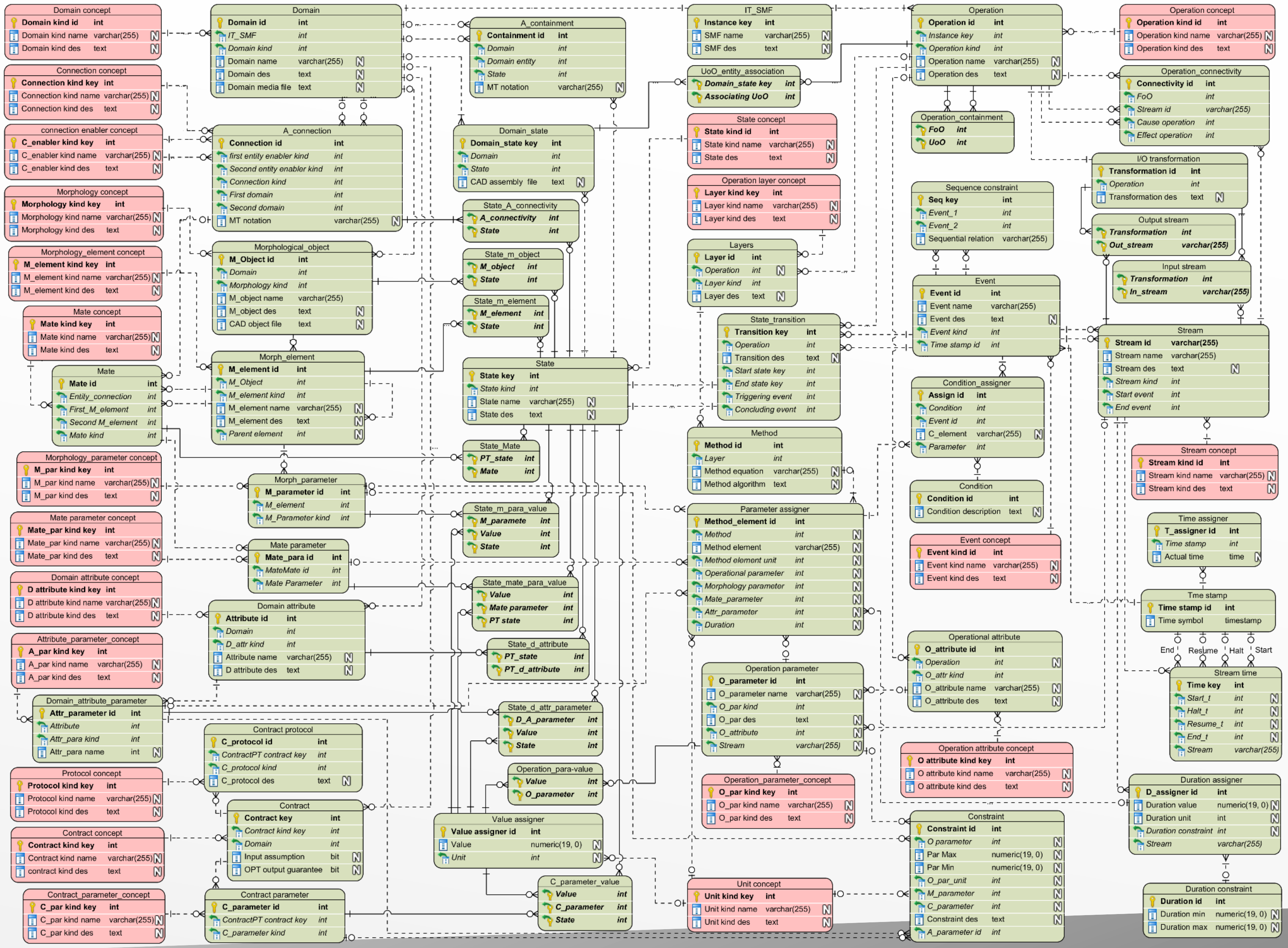


Contract_concept
Contract_kind_key int
Contract_kind_name varchar(255)
Contract_kind_des text
Domain_concept
Domain_kind int
Domain_kind_name varchar(255)
Domain_kind_des text
GT_possible_super_domain
GT_DS int
Domain_kind int
Morphology_concept
Morphology_kind_key int
Morphology_kind_name varchar(255)
Morphology_kind_des text
GT_entity
GT_entity_key int
GT_entity_kind int
GT_entity_name varchar(255)
GT_entity_des text
GT_morph_kind int
First_e
Sec_e
GT_entity_connection
GT_connection_key int
GT_first_entity int
GT_sec_entity int
GT_connection_kind int
GT_first_enabler_kind int
GT_sec_enabler_kind int
Mereotopological notation varchar(255)
GT_DS_key int
First_enabler
Sec_enabler
Connection_enabler_concept
Enabler_kind_key int
Enabler_kind_name varchar(255)
Enabler_kind_des text
Connection_concept
Connection_kind_key int
Connection_kind_name varchar(255)
Connection_kind_des text
GT_SMF
GT_id int
GT_name varchar(255)
GT_des text
GT_domain
GT_domain id int
Genotype int
GT_domain_kind int
GT_domain_name varchar(255)
GT_domain_des text
GT_contract
GT_contract_key int
GT_DS int
GT_contract_kind int
GT_input assumption bit
GT_output_guarantee bit
GT_domain_state
GT_DS_key int
GT_domain int
GT_state varchar(255)
GT_morph_kind int
GT_architectural_containment
GT_containment id int
GT_domain_state_key int
GT_entity_key int
Mereotopological notation varchar(255)
GT_template
Template name varchar(255)
GT_SMF id int
GT_template_structure
GT_T_structure id int
GT_template name varchar(255)
GT_A_containment int
GT_A_connectivity int
GT_O_connectivity int
GT_O_containment int
GT_IO_transformation
GT_transformation_key int
GT_FoO int
GT_output_stream
GT_stream_key int
GT_transformation int
GT_stream_kind int
GT_input_stream
GT_stream_key int
GT_transformation int
GT_stream_kind int
Stream_concept
Stream_kind_key int
Stream_kind_name varchar(255)
Stream_kind_des text
GT_UoO_connectivity
GT_UoO_con_key int
GT_stream_kind int
GT_sender_UoO int
GT_receiver_UoO int
GT_event_kind int
FoO int
GT_UoO_entity relation
GT_UoO ...
GT_containment id ...
GT_state
GT_state_key varchar(255)
GT_state_kind int
State_concept
State_kind_key int
State_kind_name varchar(255)
State_kind_des text
GT_operation
GT_operation_key int
Genotype int
GT_FoO_name varchar(255)
GT_FoO_des text
GT_FoO_kind int
GT_possible_super_operation
GT_FoO int
Operation_kind int
Operation_concept
Operation_kind_key int
Operation_kind_name varchar(255)
Operation_kind_des text
GT_operation_containment
GT_O_containment id int
GT_FoO int
GT_UoO int
Sender
Receiver
GT_UoO
GT_UoO_key int
GT_UoO_name varchar(255)
GT_UoO_kind int
GT_UoO_des text
Event_concept
Event_kind_key int
Event_kind_name varchar(255)
Event_kind_des text
Start_event
End_event
Start_state
End_state
GT_state_transition
GT_transition_key int
GT_FoO int
GT_start_state varchar(255)
GT_end_state varchar(255)
GT_start_event_kind int
GT_end_event_kind int
GT_transition_des text

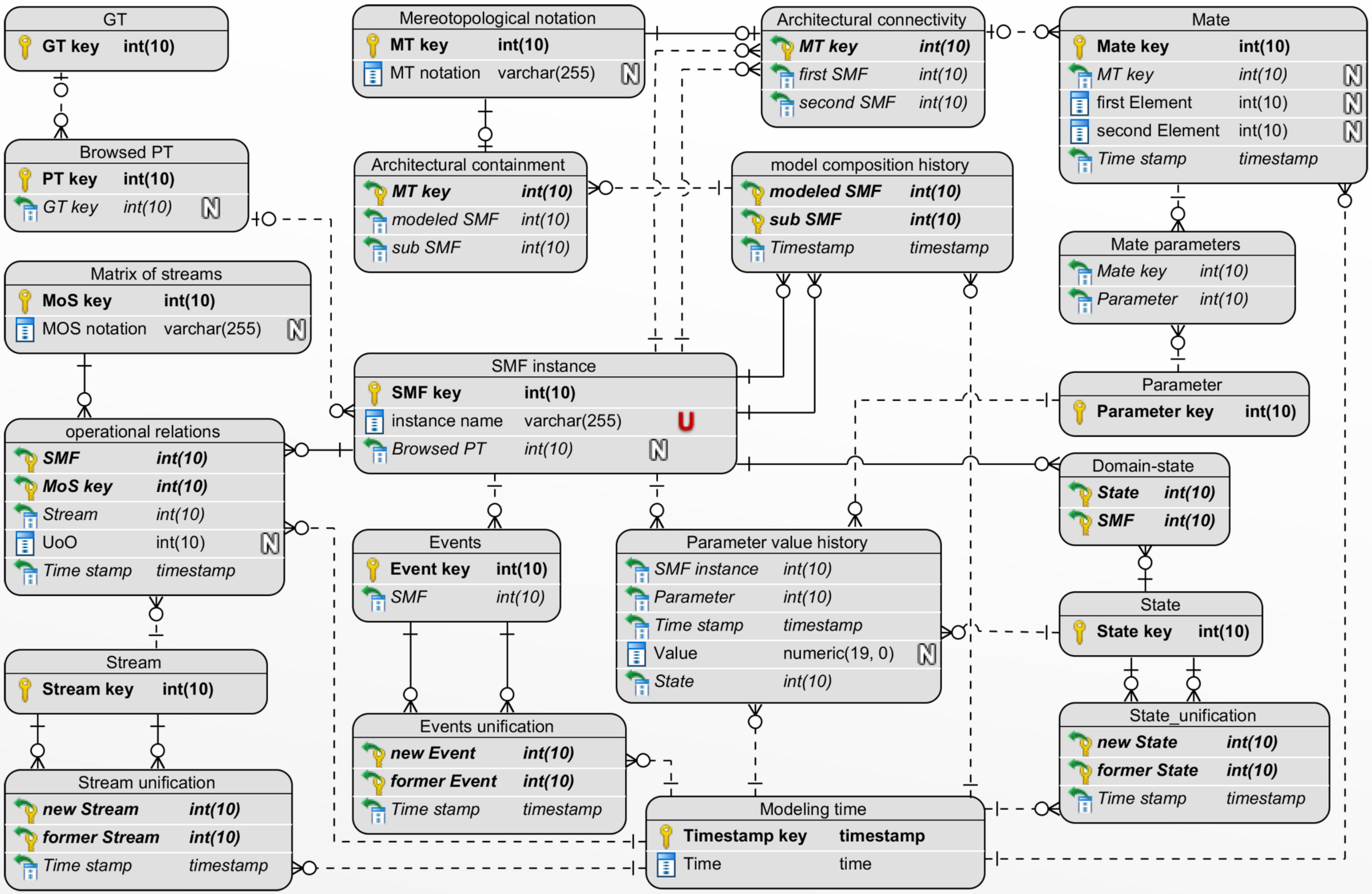
PT database


PT_possible super domain
PT_super domain id
Domain_state
PT domain kind
PT s_domain name
PT s_domain des
Domain concept
Domain kind id
Domain kind name
Domain kind des
Connection concept
Connection kind key
Connection kind name
Connection kind des
connection enabler concept
C_enabler kind key
C_enabler kind name
C_enabler kind des
Morphology concept
Morphology kind key
Morphology kind name
Morphology kind des
Morphology_element concept
M_element kind key
M_element kind name
M_element kind des
Mate concept
Mate kind key
Mate kind name
Mate kind des
PT_Mate
PT_Mate id
PT_entity_connection
First_M_element
Second M_element
Mate kind
Morphology_parameter concept
M_par kind key
M_par kind name
M_par kind des
Mate parameter concept
Mate_par kind key
Mate_par kind name
Mate_par kind des
Domain attribute concept
D attribute kind key
D attribute kind name
D attribute kind des
Attribute_parameter_concept
A_par kind key
A_par kind name
A_par kind des
Protocol concept
Protocol kind key
Protocol kind name
Protocol kind des
Contract concept
Contract kind key
Contract kind name
contract kind des
PT_domain
PT domain id
PT SMF
PT domain kind
PT domain name
PT domain des
PT domain media file
PT_entity
PT entity id
PT entity kind
PT entity name
PT entity des
PT_entity_connection
PTconnection id
PT first entity
PT second entity
PT first entity enabler kind
PT second entity enabler kind
PT connection kind
Mereotoplogical notation
PT_morphological_object
PT_M_Object id
PT_domain
PT entity
Morphology kind
PT_M_object name
PT_M_object des
PT_CAD object file
PT_morph_element
PT_M_element id
PT_M_Object
M_element kind
M_element name
M_element des
Parent element
PT_morphology parameter
PT M_parameter id
PT_M_element
PT M_Parameter
PT_Mate parameter
PT_Mate_para id
PT_Mate id
Mate Parameter
PT_domain attribute
PT attribute id
PT domain
PT entity
PT D_attr kind
PT D attribute des
PT attribute name
PT_domain_attribute_parameter
PT_attr_parameter id
PT_attribute
PT_attr_para kind
PT_attr_para name
Contract protocol
C_protocol id
PT_contract key
C_protocol kind
C_protocol des
PT_contract
PT contract key
PT contract kind key
PT input assumption
PT output guarantee
PT_Morh_element
PT_DS_A_containment
PT_DS_entity id
PT_DS_key
PT_entity
MT notation
PT_domain_state
Domain_state key
PT domain id
PT state key
PT_d_CAD assembly file
PT_DS_A_connectivity
PT_DS_A_connectivity id
PT_DS_key
PT_A_connectivity
PT_state_morph_object
PT_state
PT_m_object
PT_state
PT state key
PT_state kind
PT state name
PT state des
PT_state_morph_element
PT_state
PT_m_element
PT_state_Mate
PT_state
PT_Mate
PT_m_para_value
PT_m_parameter
Value
PT state
PT_Mate para_value
Mate parameter
Value
PT state
PT_state_attribute
PT_state
PT_d_attribute
Value
Value id
Value
Unit
PT_d_attr_para_value
PT_d_attr_para
Value
PT state
PT_DS_contract
PT_DS_contract id
PT_DS_key
PT_contract_para
C_protocol
Contract parameter
C_parameter id
PT contract key
C_parameter kind
PT_SMF
PT SMF id
PT SMF name
PT SMF des
PT_UoO_PT_entity
PT UoO id
PT_DS_entity id
State concept
State kind id
State kind name
State des
Operation layer concept
Layer kind key
Layer kind name
Layer kind des
PT_operation_layers
PT layer id
FoO id
PT UoO id
PT layer kind
PT layer des
PT_state_transition
PT transition key
PT FoO
PT transition des
Start state key
End state key
PT start event
PT end event
PT_method
PT_method id
PT layer
PT method equation
PT method algorithm
PT_parameter assigner
PT_method_element id
PT_method id
PT method element
PT_method element unit
PT_operational parameter
PT_morphology parameter
PT_attr_parameter
PT_Mate_parameter
duration
PT_operational_parameter
PT O parameter id
PT O parameter name
PT O par kind
PT O par des
PT O_attribute
PT stream
Operation_parameter_concept
O_par kind key
O_par kind name
O_par kind des
Unit concept
Unit kind key
Unit kind name
Unit kind des
Contract_parameter_concept
C_par kind key
C_par kind name
C_par kind des
PT_operation
FoO id
SMF_PT
FoO kind
FoO name
FoO des
PT_operation_containment
PT_FoO id
PT_UoOid
PT_UoO
PT UoO id
PT UoO kind
PT UoO name
PT UoO des
Event concept
Event kind id
Event kind name
Event kind des
PT_sequence constraint
PT Seq key
PT event_1
PT event_2
PT_seq relation
PT_event
PT event id
PT event name
PT event des
PT event kind
Time stamp id
PT_condition_assigner
PT assign id
Condition id
Event id
C_element
Parameter
PT_condition
condition id
Condition description
PT operational attribute
PT O_attribute id
FoO
PT UoO
O_attr kind
O_attribute name
O_attribute des
Operation attribute concept
O attribute kind key
O attribute kind name
O attribute kind des
PT_constraint
PT constraint id
par Max
Par Min
Unit
PT O parameter
PT M_parameter
PT C_parameter id
PT constraint des
PT_A_parameter
PT_possible super operation
PT s_operation kind
PT_FoO id
PT_s_operation name
Operation concept
Operation kind id
Operation kind name
Operation kind des
PT_UoO connectivity
PT connectivity id
PT stream id
UoO as sender
UoO as reciever
FoO
PT_I/O transformation
PT transformation id
PT FoO
PT trans des
PT_output stream
PT trans
PT stream
PT_input stream
PT_trans
PT stream
PT_stream
PT stream id
PT stream name
PT stream des
PT stream kind
PT start event
PT end event
Stream concept
Stream kind id
Stream kind name
Stream kind des
PT_time stamp
Time stamp id
Time symbol
End
Resume
Halt
Start
PT Stream time
time key
Start_t
Half_t
Resume_t
End_t
PT stream
PT_duration constraint
PT duration id
duration min
duration max
Duration unit
Duration symbol
PT stream

Model database



Meta-Model database



Affordances of SMF-TB

- Imposing strictly physical view
- Concurrent consideration of architecture and operation
- Uniform information structures for heterogeneous components
- Multi-level multi-granularity
- Multi-aspect coupling among SMFs
- Multi-stage model composition
- Multi-purpose system-level features
- Multiple application context
- Multi-component warehouses
- Active ontologies

Aspects of benchmarking

A1: Addressing heterogeneity of CPSs

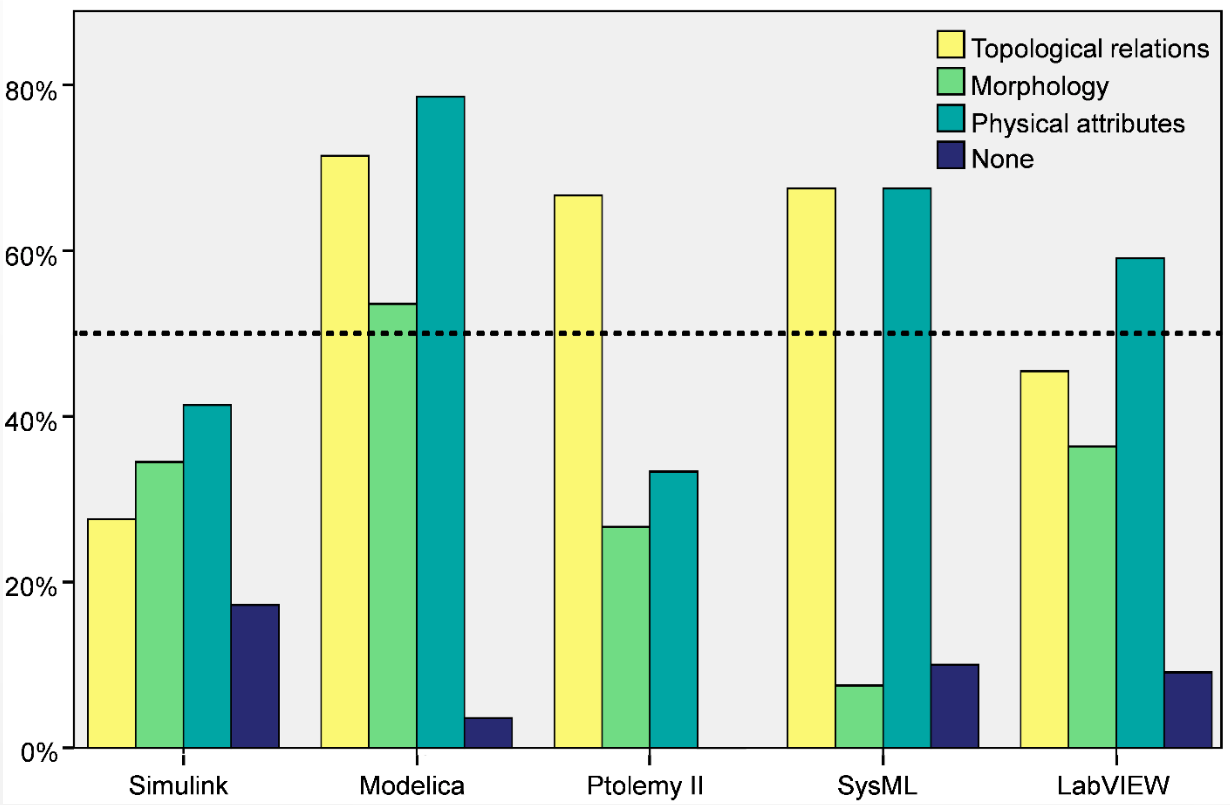
A2: Addressing integrity of CPSs:

A3: Addressing complexity of CPSs:

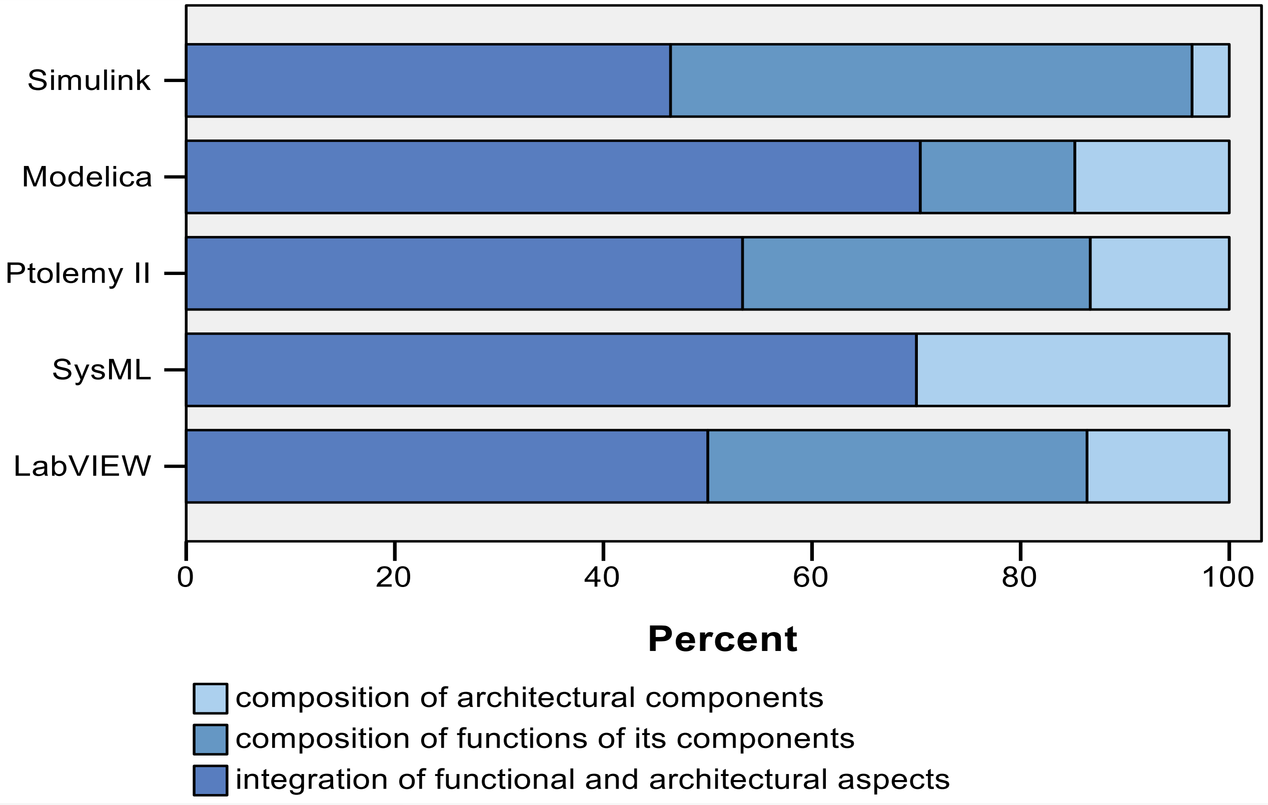
A4: Addressing information comprehensiveness:

A5: Addressing tool implementation:

Results

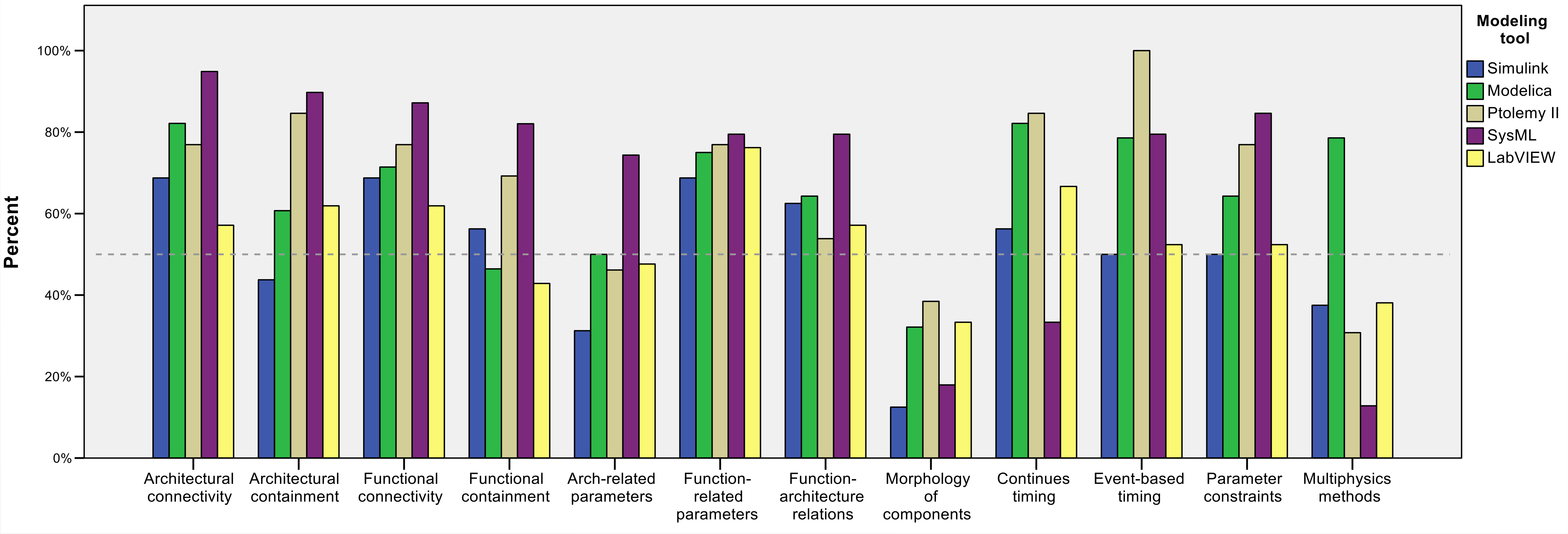


Topological relations
Morphology
Physical attributes
None
80%
60%
40%
20%
0%
Simulink
Modelica
Ptolemy II
SysML
LabVIEW



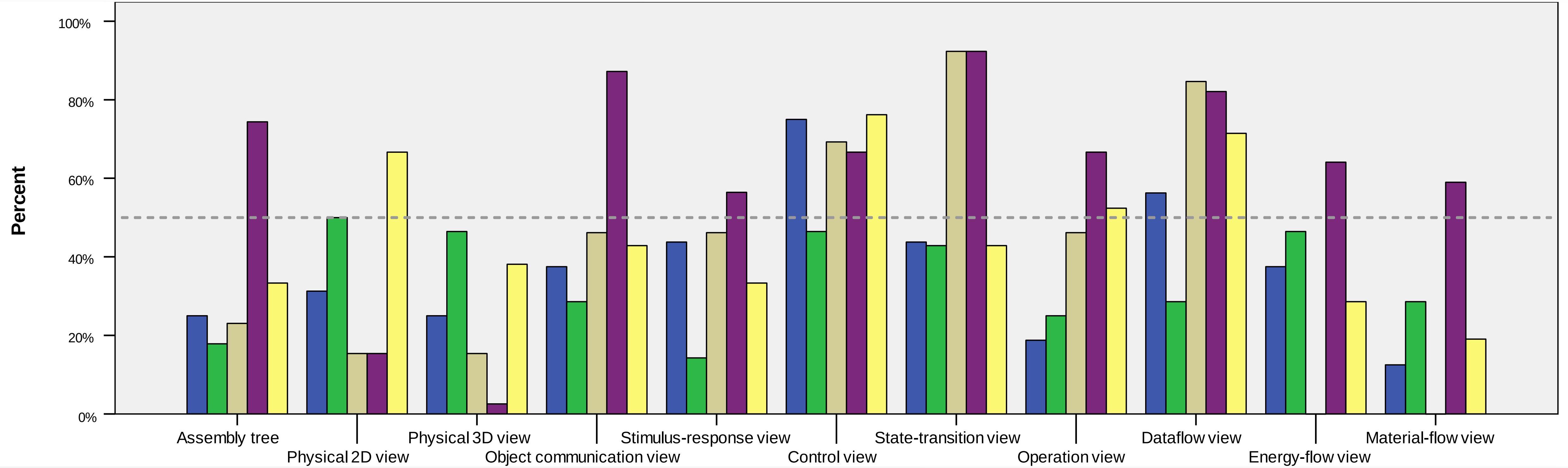
Simulink
Modelica
Ptolemy II
SysML
LabVIEW
0
20
40
60
80
100
Percent
composition of architectural components
composition of functions of its components
integration of functional and architectural aspects

Results



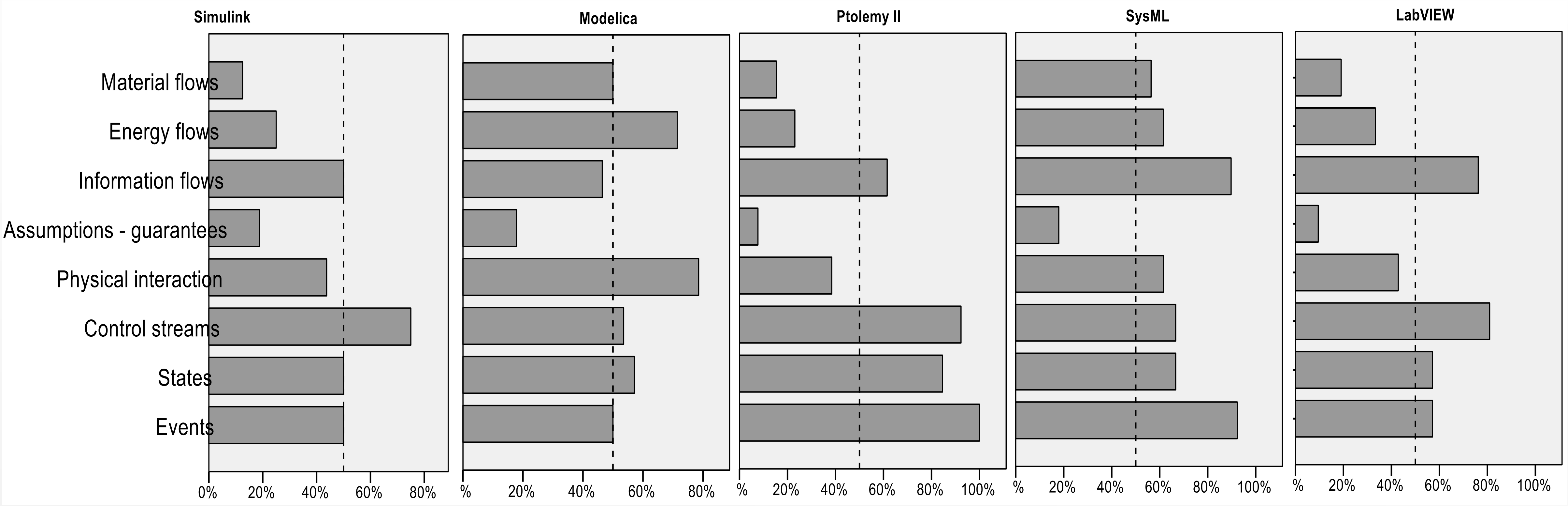
Modeling tool
Simulink
Modelica
Ptolemy II
SysML
LabVIEW
Percent
100%
80%
60%
40%
20%
0%
Architectural connectivity
Architectural containment
Functional connectivity
Functional containment
Arch-related parameters
Function-related parameters
Function-architecture relations
Morphology of components
Continues timing
Event-based timing
Parameter constraints
Multiphysics methods

Results



100%
80%
60%
40%
20%
0%
Percent
Assembly tree
Physical 2D view
Physical 3D view
Object communication view
Stimulus-response view
Control view
State-transition view
Operation view
Dataflow view
Energy-flow view
Material-flow view

Results



Simulink
Modelica
Ptolemy II
SysML
LabVIEW
Material flows
Energy flows
Information flows
Assumptions - guarantees
Physical interaction
Control streams
States
Events
0%
20%
40%
60%
80%
100%

Publications

- Pourtalebi, S., Horváth, I., & Opiyo, E. (2013). **Multi-aspect study of mass customization in the context of cyber-physical consumer durables**. In ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers, pp. V004T05A006-V004T05A006.
- Pourtalebi, S., Horváth, I., & Opiyo, E. (2014). **New features imply new principles? deriving design principles for mass customization of cyber-physical consumer durables**. Proceedings of the TMCE, Budapest, Hungary, 95-108.
- Pourtalebi, S., Horváth, I., & Opiyo, E. Z. (2014a). **First steps towards a mereo-operandi theory for a system feature-based architecting of cyber-physical systems**. In GI-Jahrestagung, pp. 2001-2006.
- Pourtalebi, S., Horváth, I., & Opiyo, E. Z. (2014b). **New features imply new principles? Deriving design principles for mass customization of cyber-physical consumer durables**. In Proceedings of the 10th international symposium on tools and methods of competitive engineering TMCE 2014, Budapest, Hungary, May 19-23, 2014, Delft University of Technology.
- Horváth, I., & Pourtalebi, S. (2015). **Fundamentals of a mereo-operandi theory to support transdisciplinary modeling and co-design of cyber-physical systems**. In ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers, pp. V01AT02A005-V01AT02A005.
- Pourtalebi, S., & Horváth, I. (2016a). **Towards a methodology of system manifestation features-based pre-embodiment design**. Journal of Engineering Design, 1-37.
- Pourtalebi, S., & Horváth, I. (2016b). **Information schema constructs for defining warehouse databases of genotypes and phenotypes of system manifestation features**. Frontiers of Information Technology & Electronic Engineering, vol. x, no. x, pp. x-x.
- Pourtalebi, S., & Horváth, I. (2017). **Information schema constructs for instantiation and composition of system manifestation features**. Frontiers of Information Technology & Electronic Engineering, vol. x, no. x, pp. x-x.
- Pourtalebi, S., & Horváth, I. (2018). **Benchmarking the conceptual framework of a system-level manifestation features-based toolbox**, (in preparation for journal publication), pp. x-x.