A Foundation for Inconsistency Management in Model-Based Systems Engineering

Istvan David

istvan.david@uantwerpen.be

PhD Defense – Antwerp, Belgium – July 1, 2019.

Supervisor: Prof. Dr. Hans Vangheluwe

MSDL
Modelling, Simulation and Design Lab

FLANDERS MAKE
MANUFACTURING INNOVATION NETWORK

Ansymo
Antwerp Systems & Software Modelling
University of Antwerp

CORRECTNESS

Correctness: satisfies the required properties.

CORRECTNESS

Correctness: satisfies the required properties.

COMPLEXITY

CORRECTNESS

Correctness: satisfies the required properties.

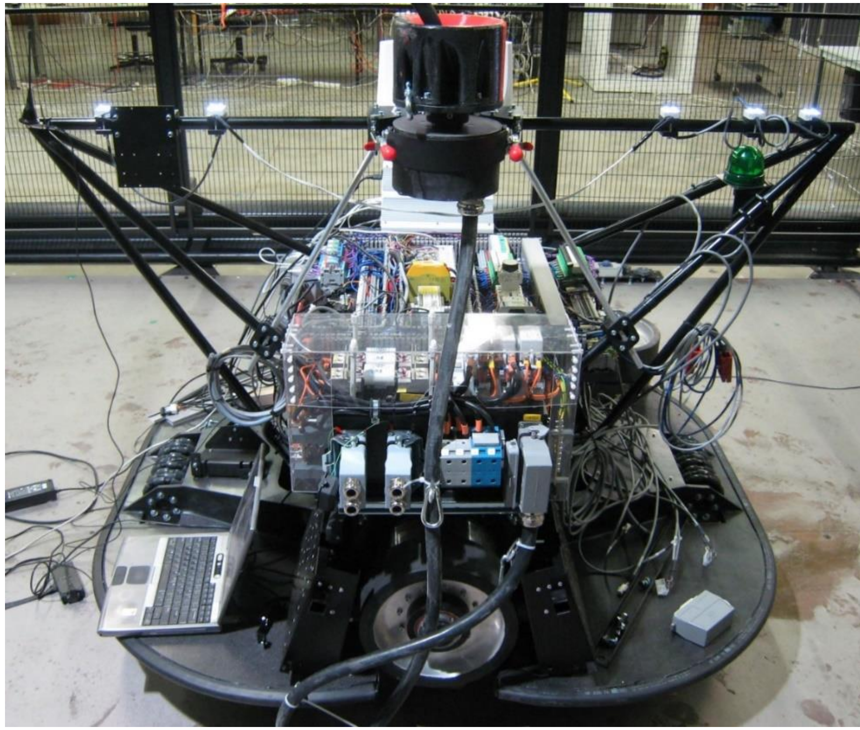
Steam engine

COMPLEXITY

CORRECTNESS

Correctness: satisfies the required properties.

Steam engine



Automated Guided Vehicle (AGV)

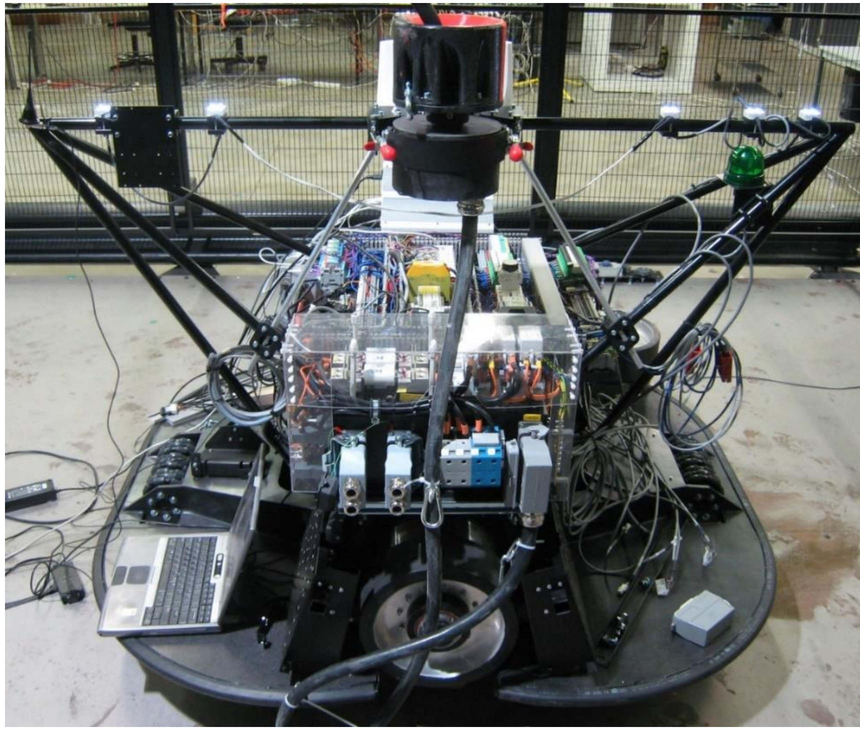
COMPLEXITY

CORRECTNESS

Correctness: satisfies the required properties.

Steam engine

	Number
Components	<
Concerns	<



Automated Guided Vehicle (AGV)

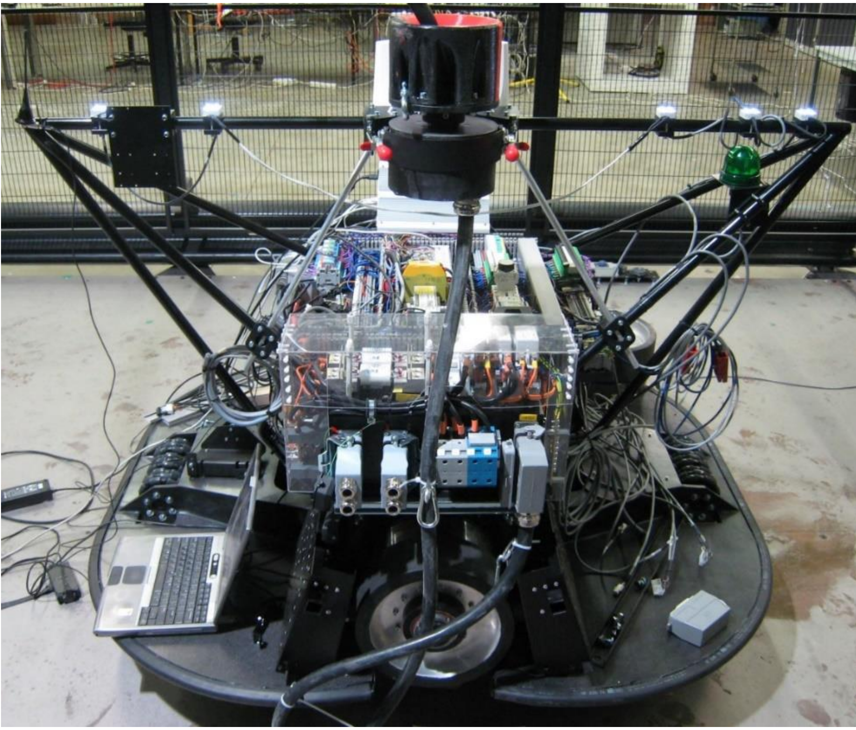
COMPLEXITY

CORRECTNESS

Correctness: satisfies the required properties.

Steam engine

	Number	Heterogeneity
Components	<	≪
Concerns	<	≪



Automated Guided Vehicle (AGV)

COMPLEXITY

Tackling complexity: Multi-paradigm modeling

MODEL
EVERYTHING!

…explicitly
at the most appropriate level(s) of abstraction
using the most appropriate formalism(s),
with processes modelled explicitly.

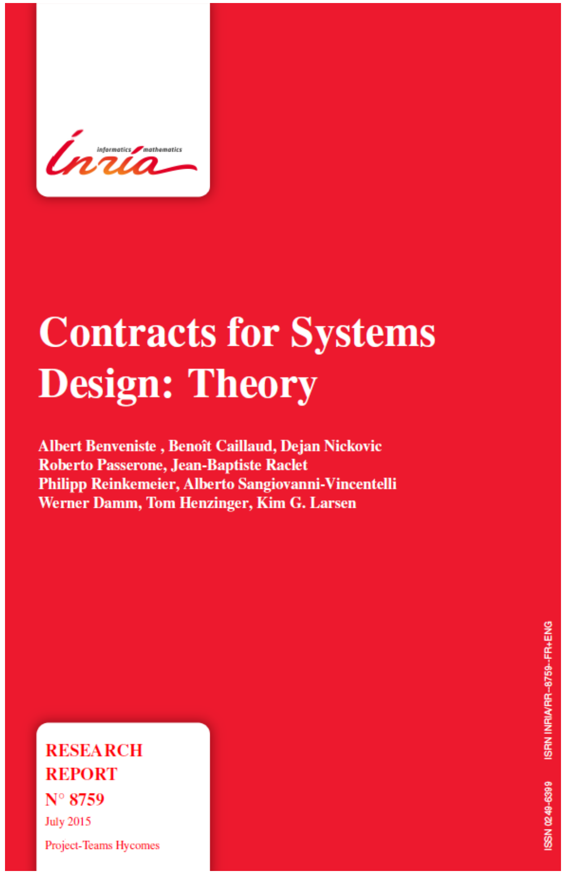
Tackling complexity: Multi-paradigm modeling

MODEL EVERYTHING!

...explicitly
at the most appropriate level(s) of abstraction
using the most appropriate formalism(s),
with processes modelled explicitly.

Tackle complexity by

- architectural decomposition
  - ...and designing individual components
- view decomposition
  - ...and work in individual views
- abstraction/refinement



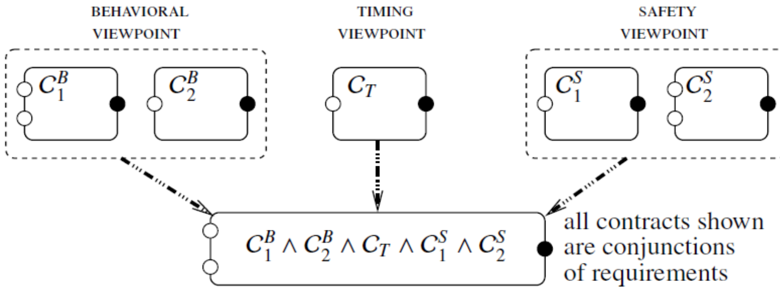
Inria informatics mathematics

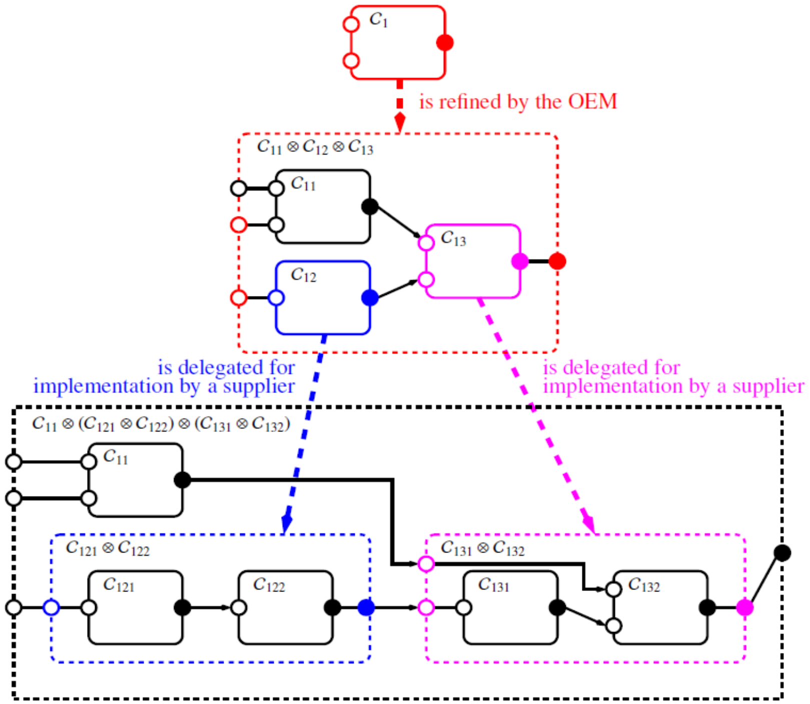
Contracts for Systems Design: Theory

Albert Benveniste , Benoît Caillaud, Dejan Nickovic
Roberto Passerone, Jean-Baptiste Raclet
Philipp Reinkemeier, Alberto Sangiovanni-Vincentelli
Werner Damm, Tom Henzinger, Kim G. Larsen

RESEARCH REPORT
N° 8759
July 2015
Project-Teams Hycomes

ISSN 0249-6399 ISRN INRIA/RR--8759--FR+ENG


BEHAVIORAL VIEWPOINT
TIMING VIEWPOINT
SAFETY VIEWPOINT
C_1^B
C_2^B
C_T
C_1^S
C_2^S
$C_1^B \wedge C_2^B \wedge C_T \wedge C_1^S \wedge C_2^S$
all contracts shown are conjunctions of requirements


C_1
is refined by the OEM
$C_{11} \otimes C_{12} \otimes C_{13}$
C_{11}
C_{13}
C_{12}
is delegated for implementation by a supplier
is delegated for implementation by a supplier
$C_{11} \otimes (C_{121} \otimes C_{122}) \otimes (C_{131} \otimes C_{132})$
C_{11}
$C_{121} \otimes C_{122}$
C_{121}
C_{122}
$C_{131} \otimes C_{132}$
C_{131}
C_{132}

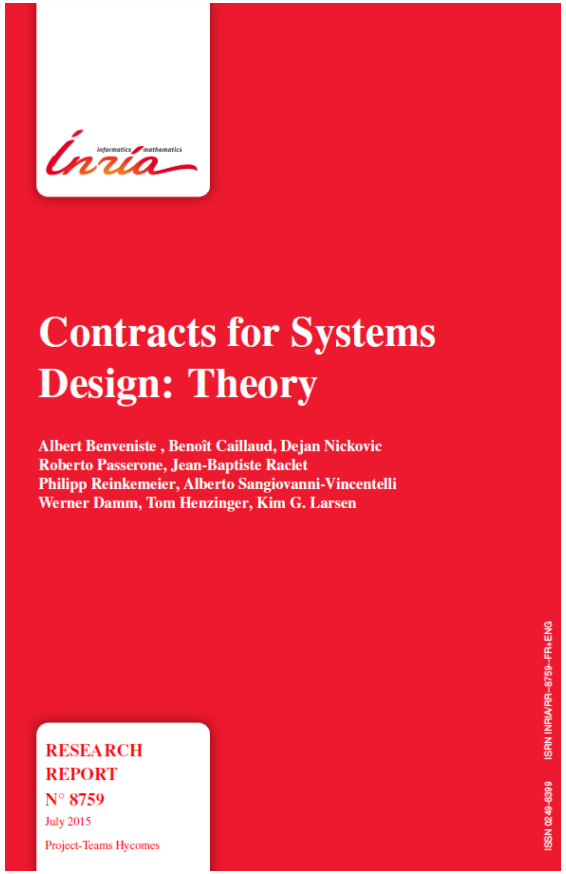
Tackling complexity: Multi-paradigm modeling

MODEL
EVERYTHING!

...explicitly
at the most appropriate level(s) of abstraction
using the most appropriate formalism(s),
with processes modelled explicitly.

Tackle complexity by

- architectural decomposition
  - ...and designing individual components
- view decomposition
  - ...and work in individual views
- abstraction/refinement

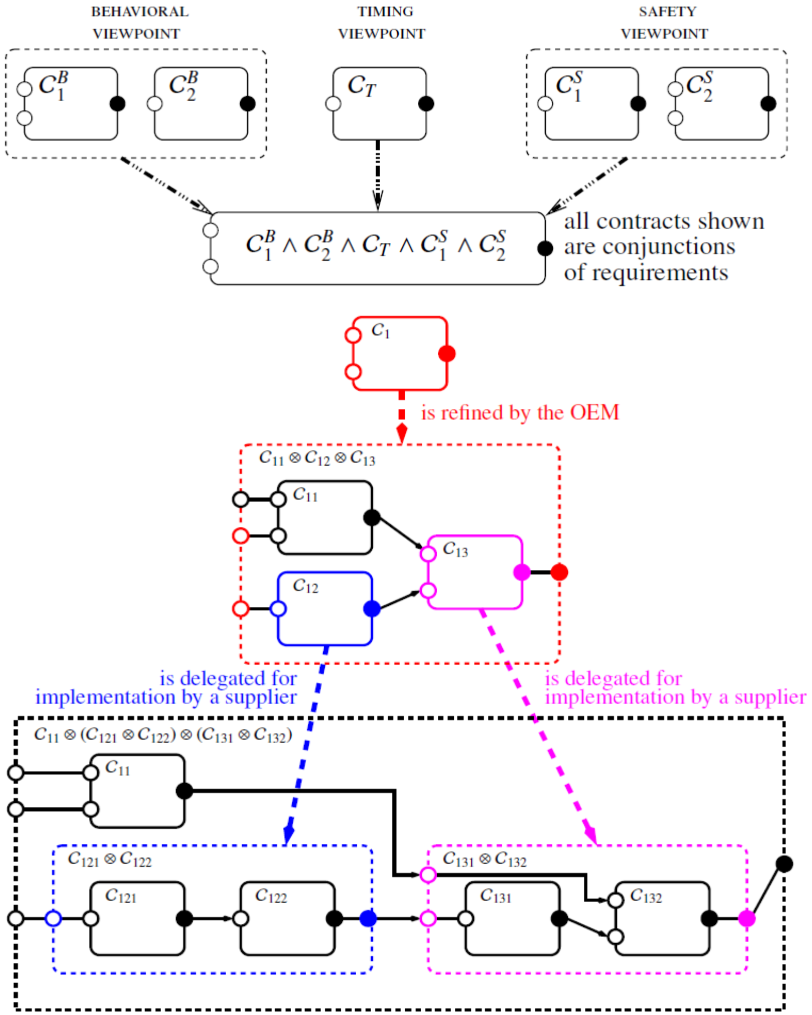


Contracts for Systems Design: Theory

Albert Benveniste , Benoît Caillaud, Dejan Nickovic
Roberto Passerone, Jean-Baptiste Raclet
Philipp Reinkemeier, Alberto Sangiovanni-Vincentelli
Werner Damm, Tom Henzinger, Kim G. Larsen

RESEARCH
REPORT
N° 8759
July 2015
Project-Teams Hycomes

ISRN INRIA/RR--8759--FR+ENG
ISSN 0249-6399


BEHAVIORAL VIEWPOINT
TIMING VIEWPOINT
SAFETY VIEWPOINT
C_1^B
C_2^B
C_T
C_1^S
C_2^S
$C_1^B \wedge C_2^B \wedge C_T \wedge C_1^S \wedge C_2^S$
all contracts shown are conjunctions of requirements
C_1
is refined by the OEM
$C_{11} \otimes C_{12} \otimes C_{13}$
C_{11}
C_{13}
C_{12}
is delegated for implementation by a supplier
is delegated for implementation by a supplier
$C_{11} \otimes (C_{121} \otimes C_{122}) \otimes (C_{131} \otimes C_{132})$
C_{11}
$C_{121} \otimes C_{122}$
C_{121}
C_{122}
$C_{131} \otimes C_{132}$
C_{131}
C_{132}

Parallel branches give rise to inconsistencies!

(In?)consistency

An inconsistency is present if two or more statements are made that are not jointly satisfiable [such as a] *failure of an equivalence test, non-conformance to a standard or constraint and the violation of physical or mathematical principles.*

S. Herzig, C. Paredis: „A conceptual basis for inconsistency management in model-based systems engineering.” *Procedia CIRP* 21 (2014): 52-57.

(In?)consistency

An inconsistency is present if two or more statements are made that are not jointly satisfiable [such as a] *failure of an equivalence test, non-conformance to a standard or constraint and the violation of physical or mathematical principles.*

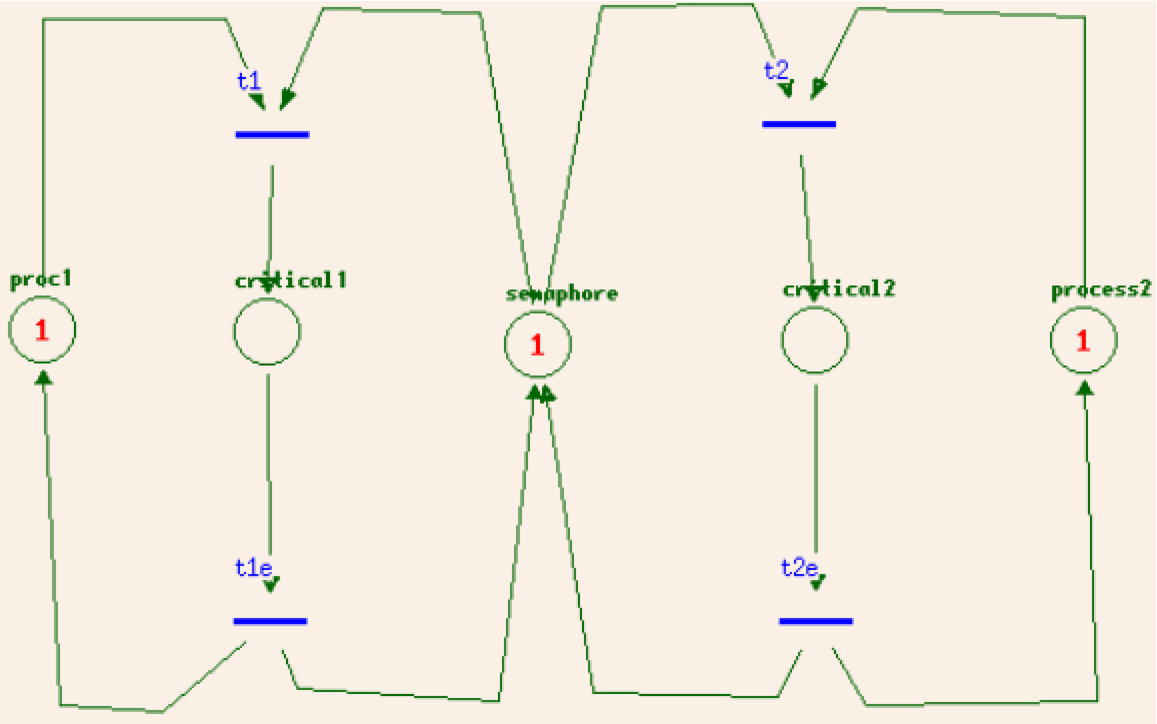
S. Herzig, C. Paredis: „A conceptual basis for inconsistency management in model-based systems engineering.” *Procedia CIRP* 21 (2014): 52-57.

Problem: (un)satisfiability often remains hidden in the semantic domain.

Properties in the semantic domain

Property: ***„The system is safe”.***

Petri net

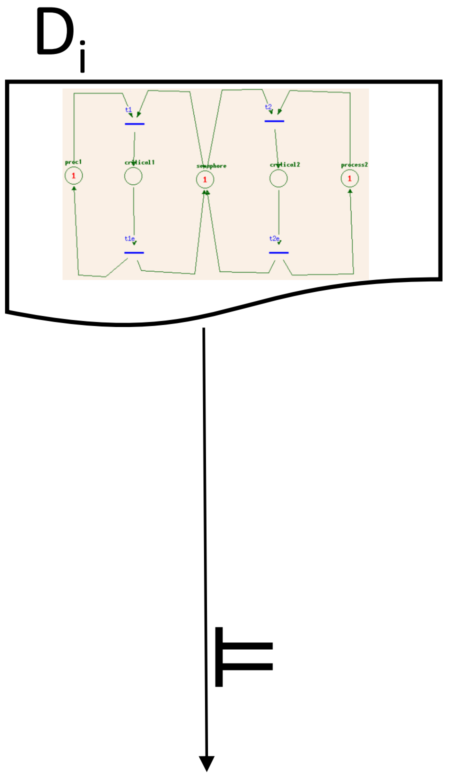


The two processes are never simultaneously situated in their respective critical sections.

Properties in the semantic domain

Property: ***„The system is safe”***.

Petri net

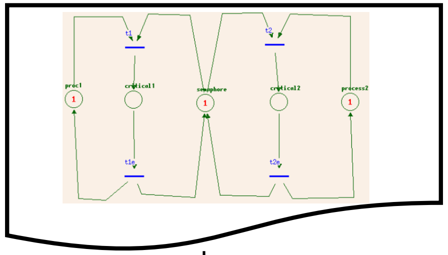


The two processes are never simultaneously situated in their respective critical sections.

Properties in the semantic domain

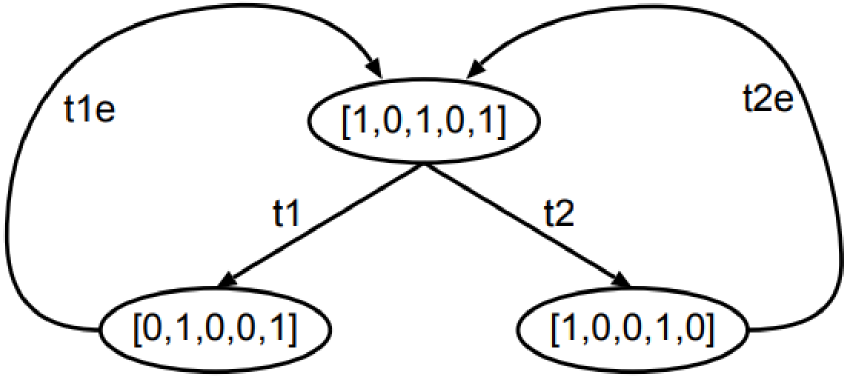
Property: ***„The system is safe”.***

D_i



[[.]]

Semantic domain



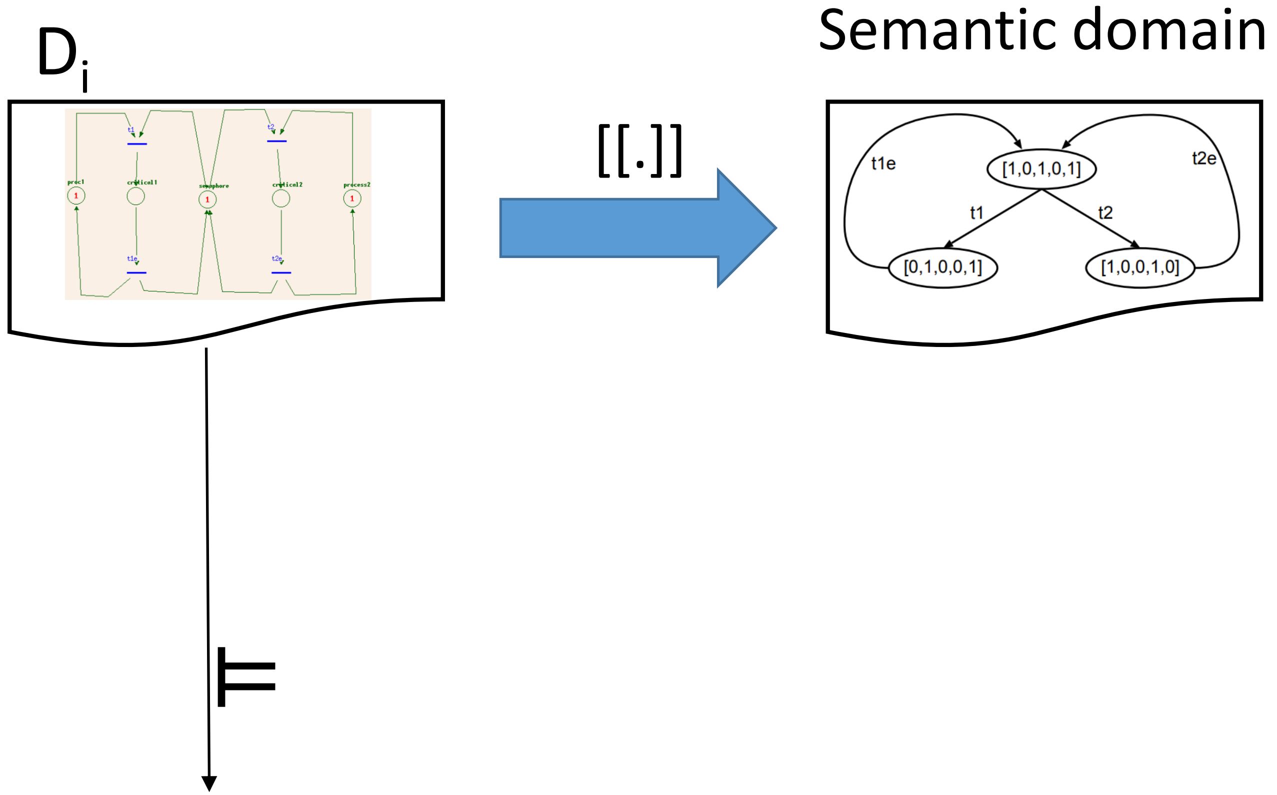
True/False

$\models$

The two processes are never simultaneously situated in their respective critical sections.

Properties in the semantic domain

Property: ***„The system is safe”***.



D_i

Semantic domain

[[.]]

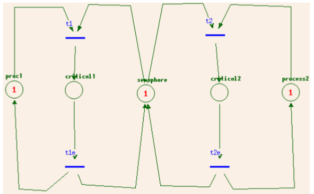
$\models$

The two processes are never simultaneously situated in their respective critical sections.

Properties in the semantic domain

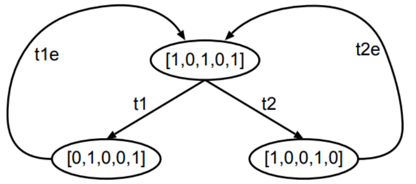
Property: ***„The system is safe”***.

D_i



[[.]]

Semantic domain



$\models$

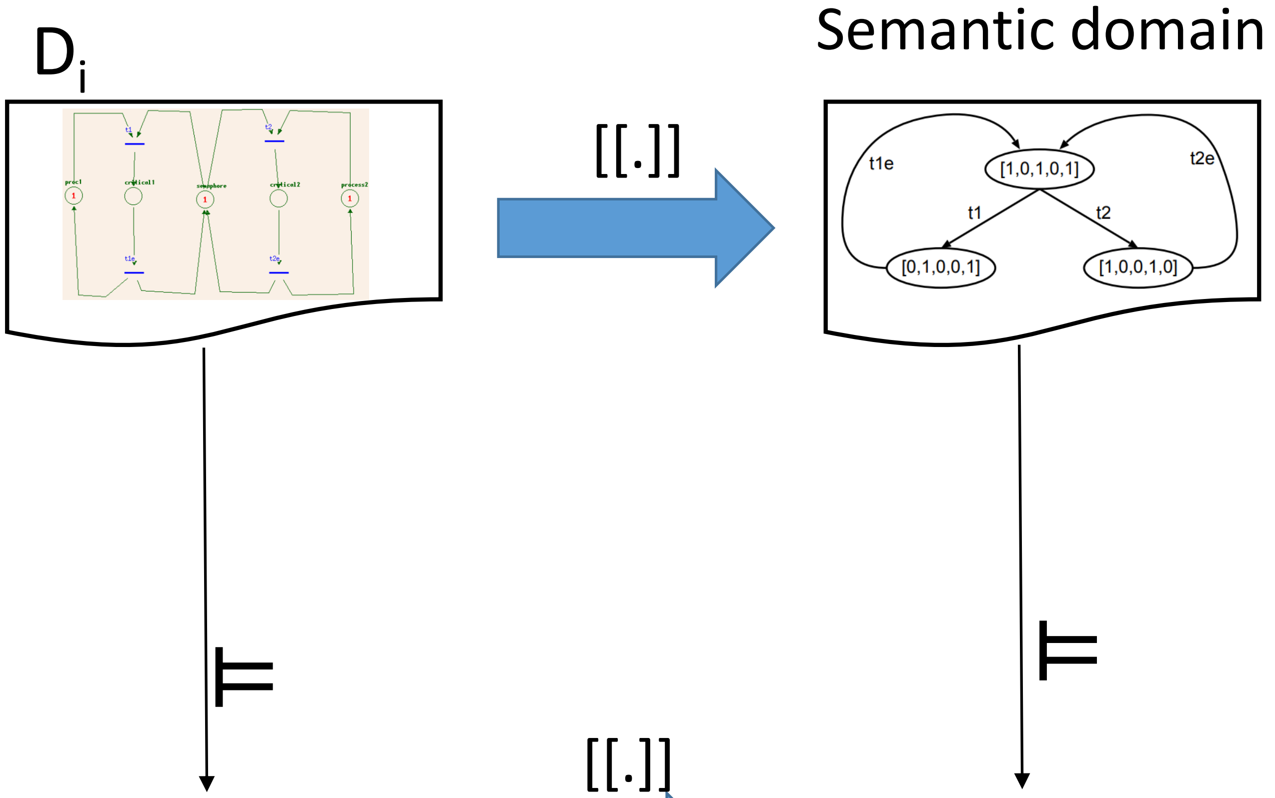
The two processes are never simultaneously situated in their respective critical sections.

[[.]]

∄[_, 1, _, 1, _]

Properties in the semantic domain

Property: ***„The system is safe”***.



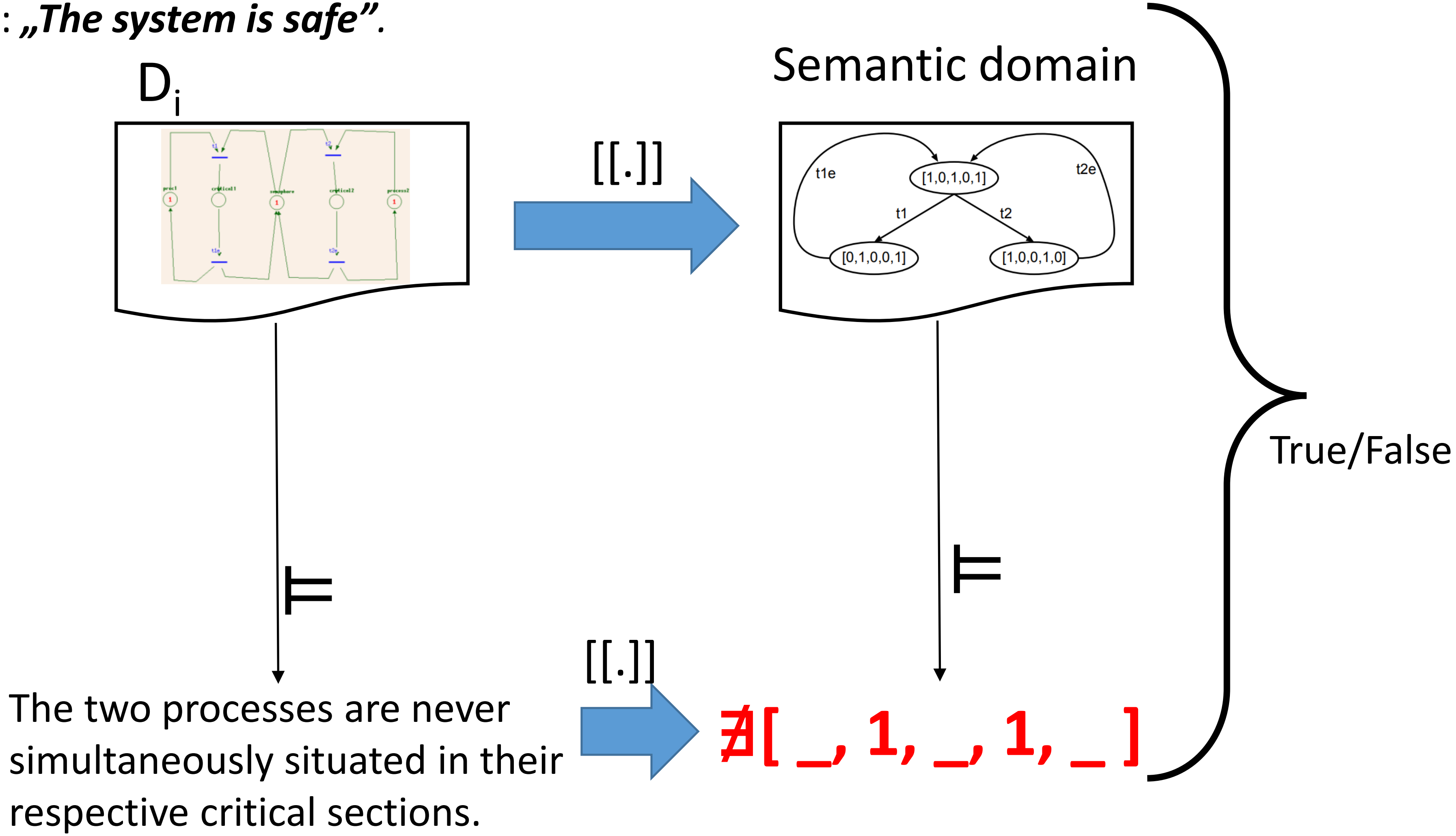
The two processes are never simultaneously situated in their respective critical sections.

[[.]]

$\nexists [\ \_, 1, \_, 1, \_\ ]$

Properties in the semantic domain

Property: ***„The system is safe”***.



D_i

Semantic domain

[[.]]

⊨

⊨

True/False

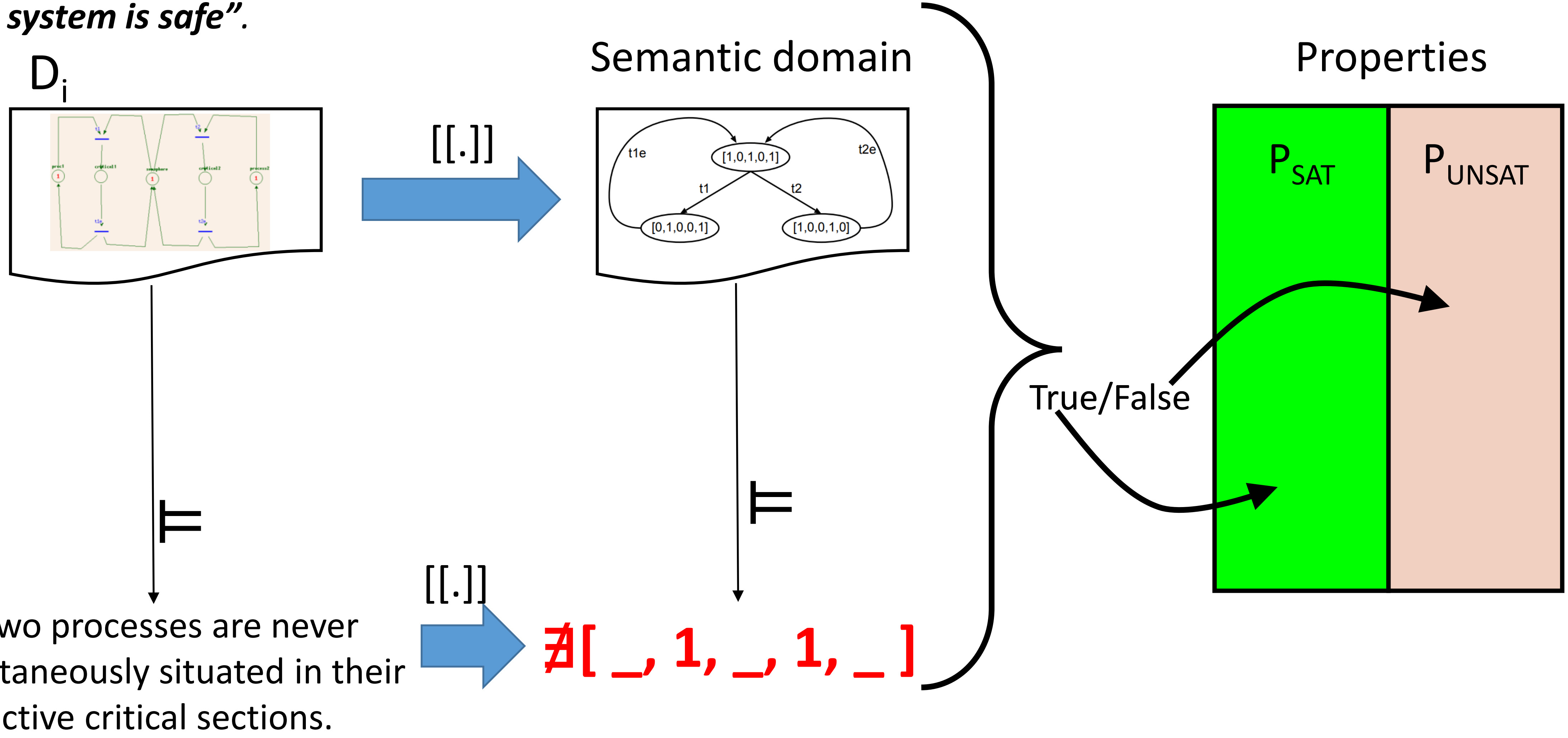
The two processes are never simultaneously situated in their respective critical sections.

[[.]]

∄[_, 1, _, 1, _]

Properties in the semantic domain

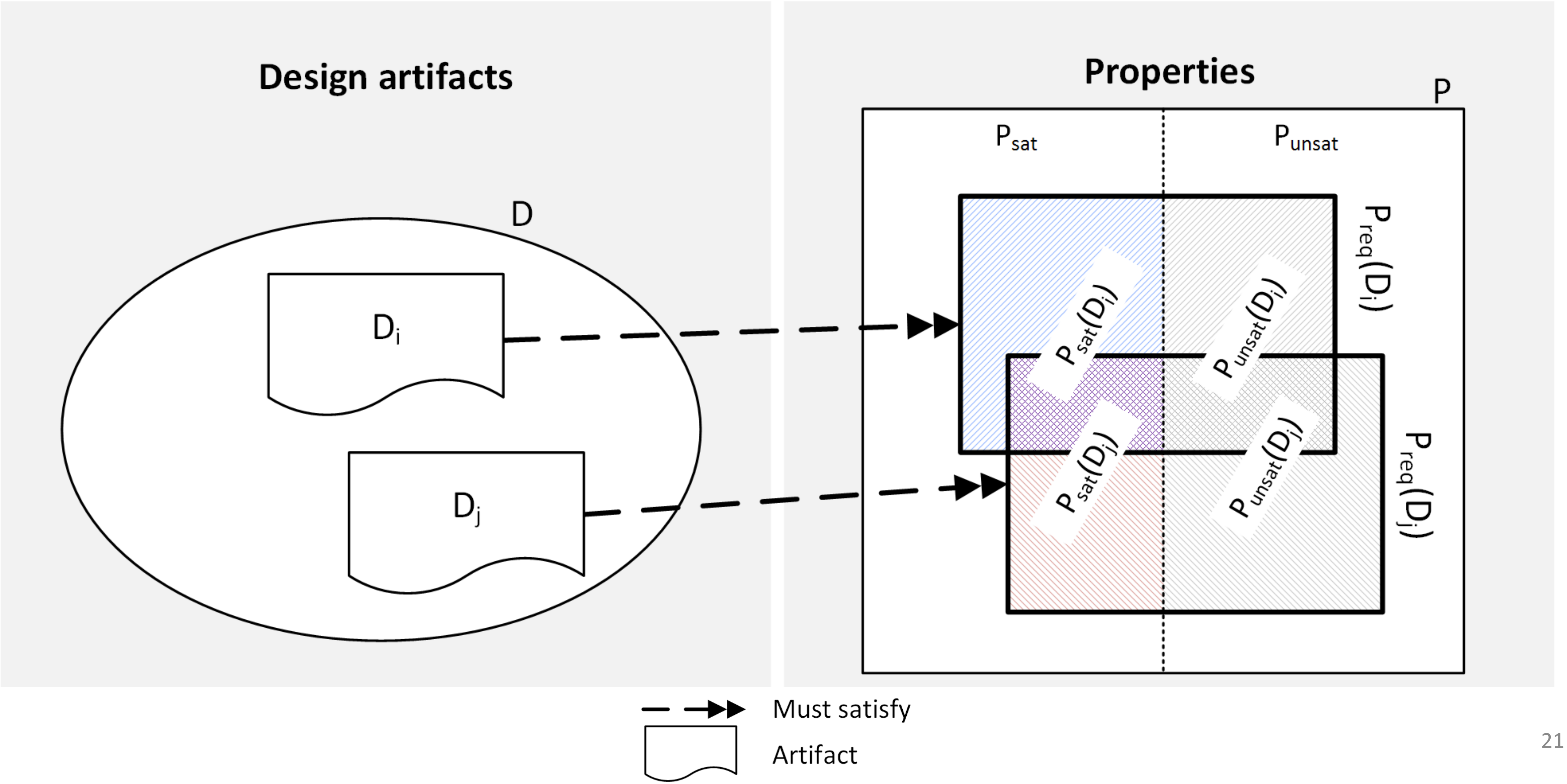
Property: ***„The system is safe”.***



The two processes are never simultaneously situated in their respective critical sections.

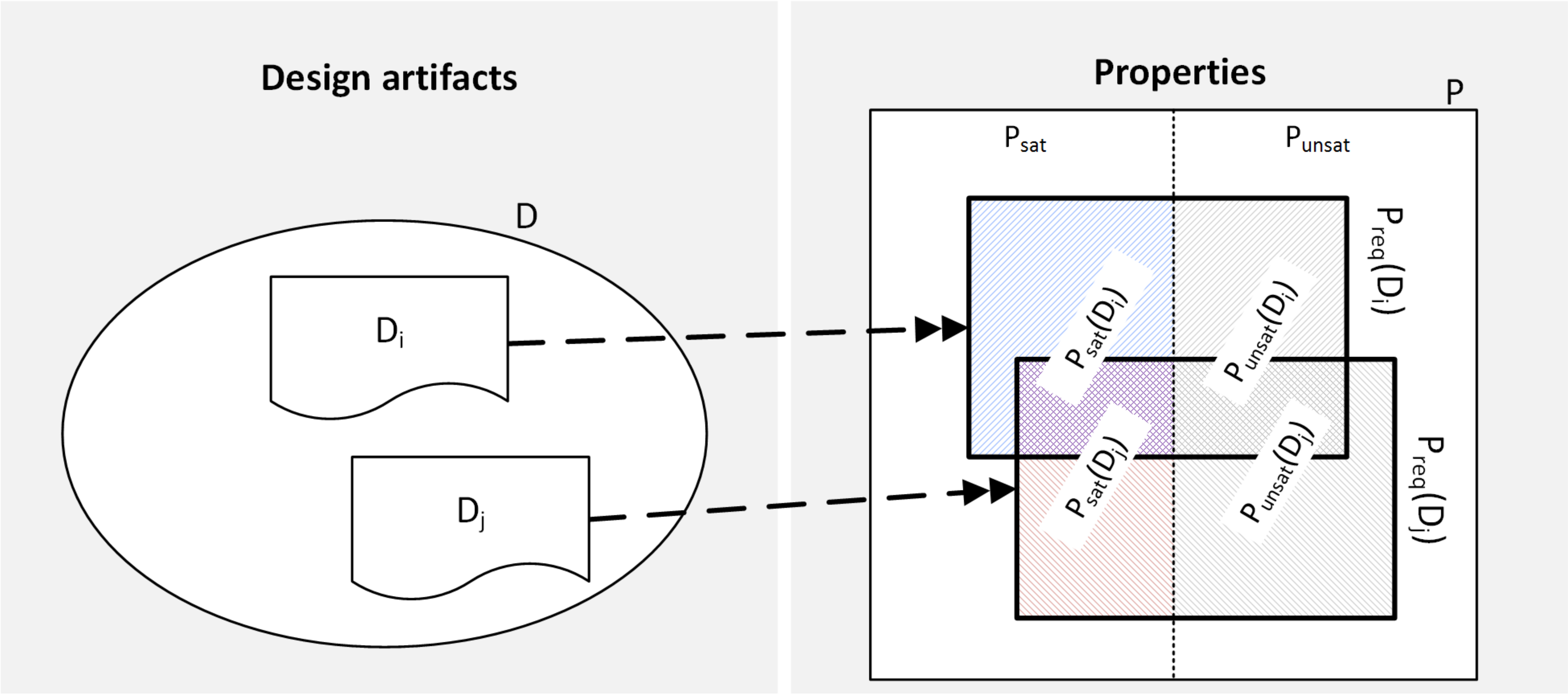
$\nexists$[_, 1, _, 1, _]

Consistency



The design artifacts D_i and D_j are said to be consistent w.r.t. the set of properties $P' \subseteq P \equiv P_{req}(D_i) \bigcap P_{req}(D_j)$ iff $\forall p \in P' : D_i \vDash p \Leftrightarrow D_j \vDash p$.

Consistency

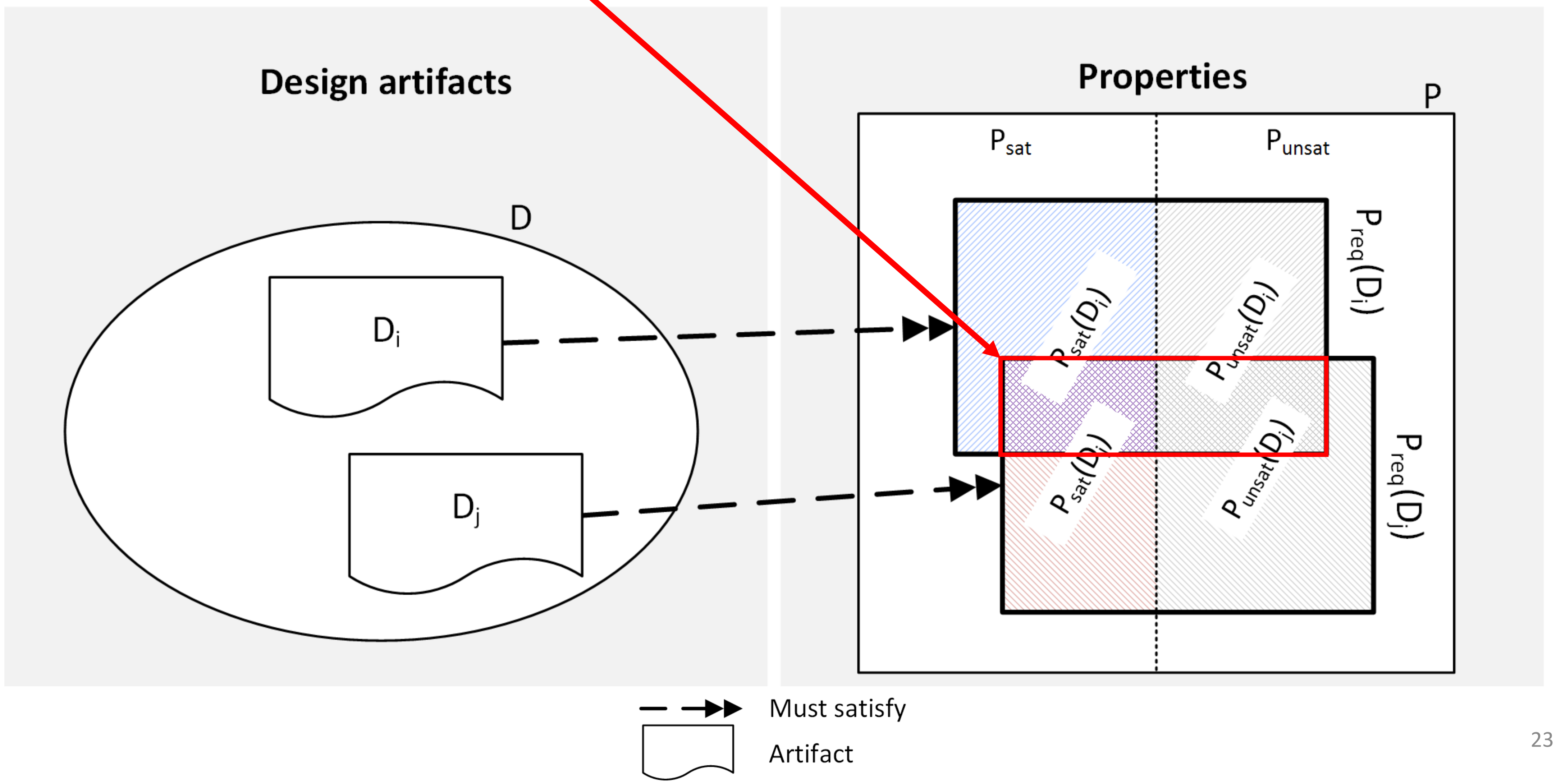


– –▶▶ Must satisfy

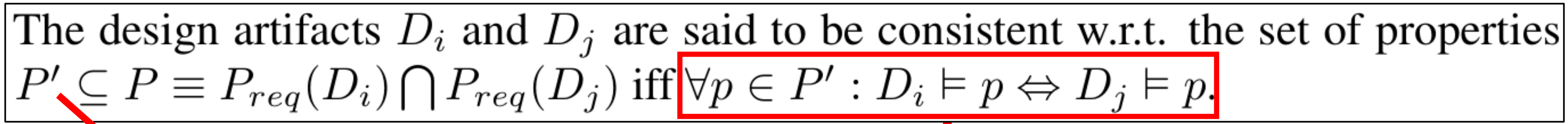
Artifact

Consistency

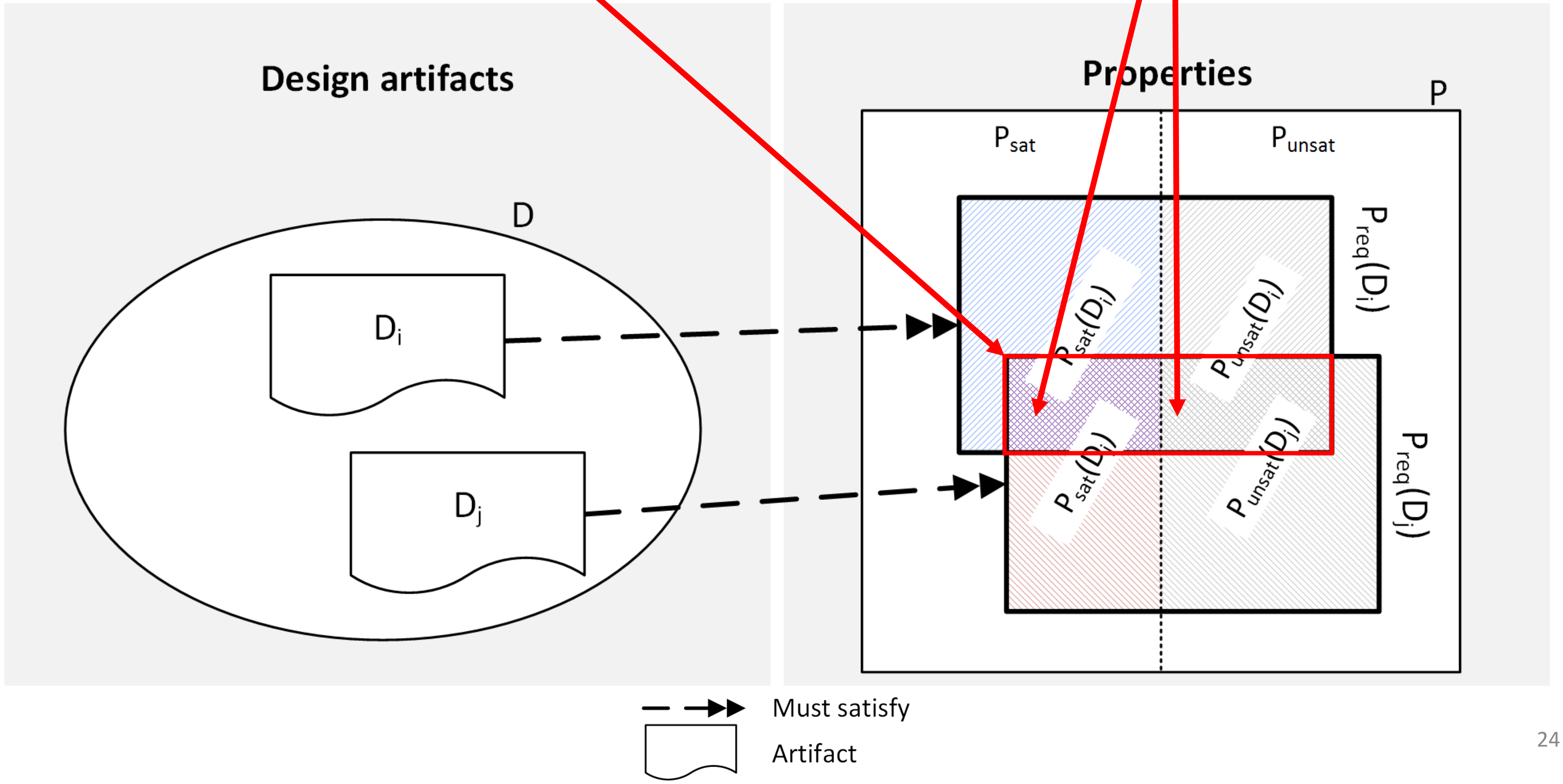
The design artifacts D_i and D_j are said to be consistent w.r.t. the set of properties $P' \subseteq P \equiv P_{req}(D_i) \bigcap P_{req}(D_j)$ iff $\forall p \in P' : D_i \models p \Leftrightarrow D_j \models p$.



Consistency



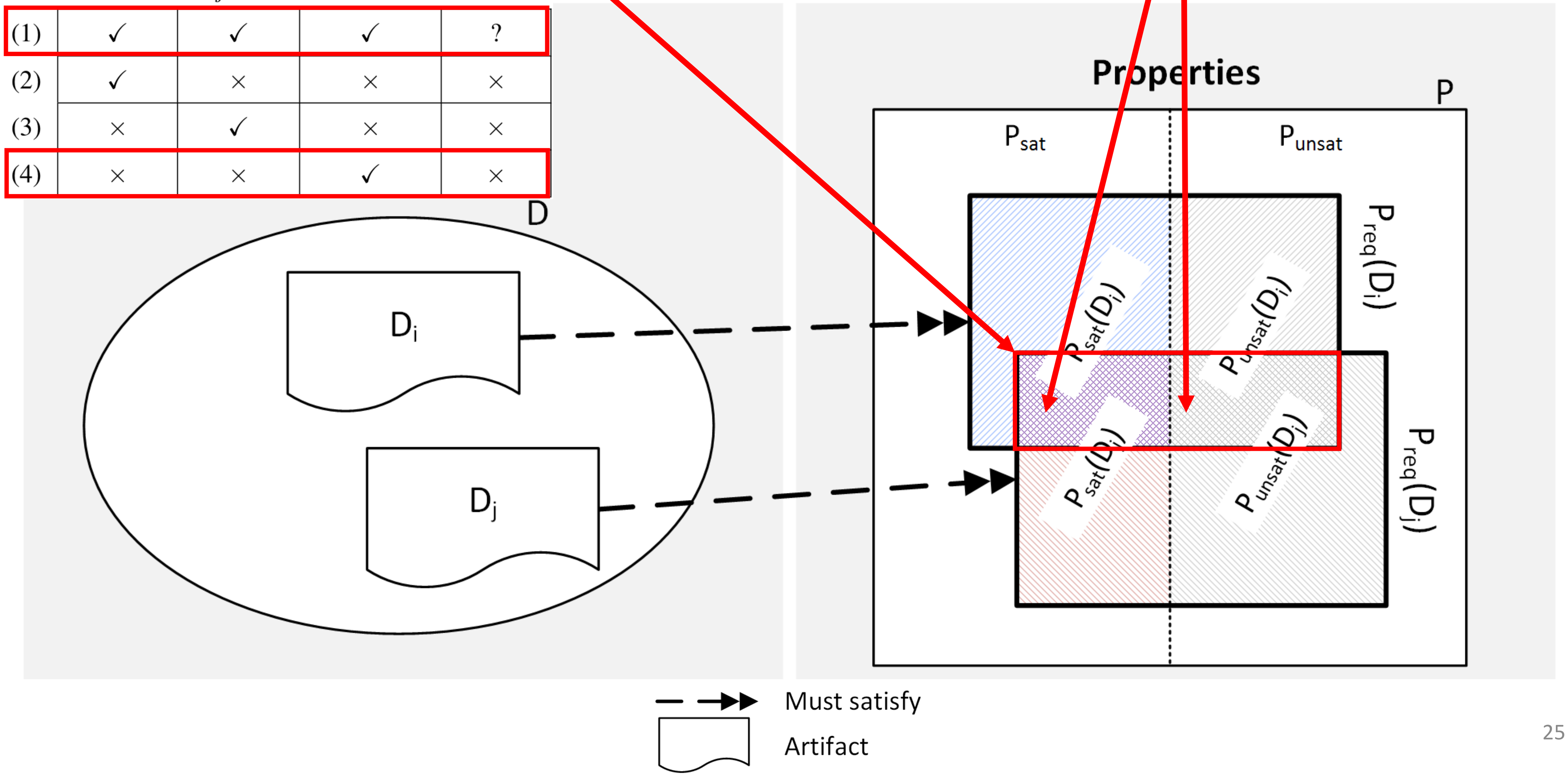
The design artifacts D_i and D_j are said to be consistent w.r.t. the set of properties $P' \subseteq P \equiv P_{req}(D_i) \bigcap P_{req}(D_j)$ iff $\forall p \in P' : D_i \models p \Leftrightarrow D_j \models p$.



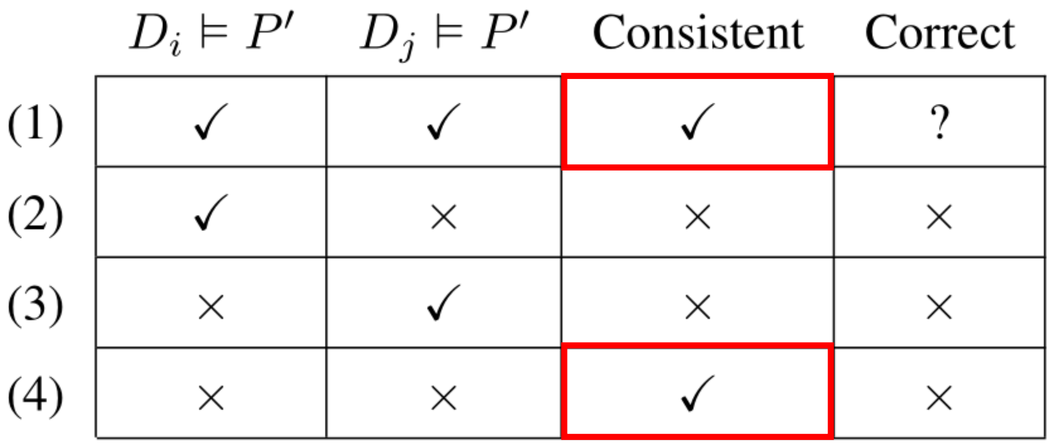
The design artifacts D_i and D_j are said to be consistent w.r.t. the set of properties $P' \subseteq P \equiv P_{req}(D_i) \bigcap P_{req}(D_j)$ iff $\forall p \in P' : D_i \models p \Leftrightarrow D_j \models p$.

Consistency

	$D_i \models P'$	$D_j \models P'$	Consistent	Correct
(1)	✓	✓	✓	?
(2)	✓	×	×	×
(3)	×	✓	×	×
(4)	×	×	✓	×

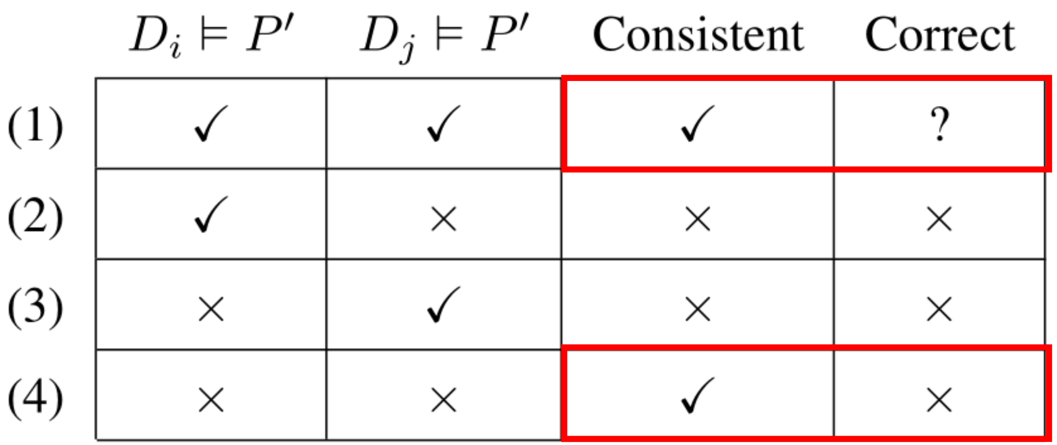


(In)correctness and (in)consistency



	$D_i \vDash P'$	$D_j \vDash P'$	Consistent	Correct
(1)	✓	✓	✓	?
(2)	✓	×	×	×
(3)	×	✓	×	×
(4)	×	×	✓	×

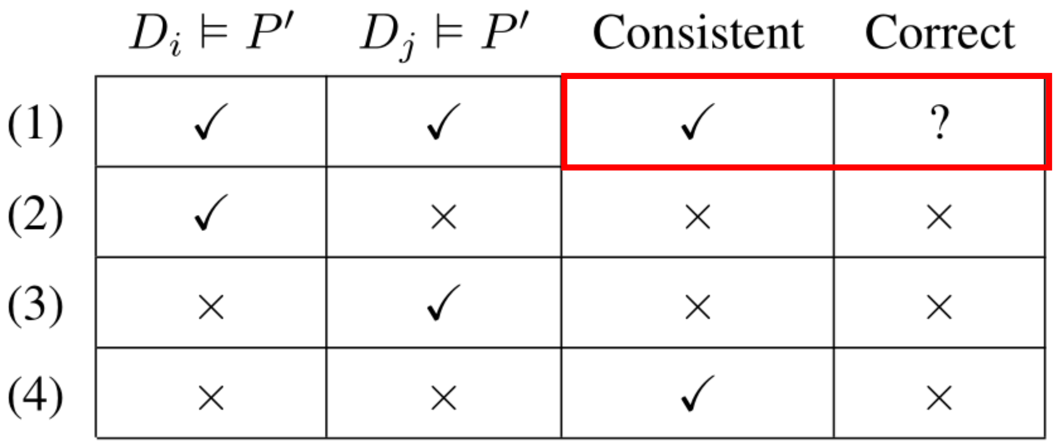
(In)correctness and (in)consistency



	$D_i \models P'$	$D_j \models P'$	Consistent	Correct
(1)	✓	✓	✓	?
(2)	✓	×	×	×
(3)	×	✓	×	×
(4)	×	×	✓	×

$$consistent(D_i, D_j) \not\Rightarrow correct(D)$$

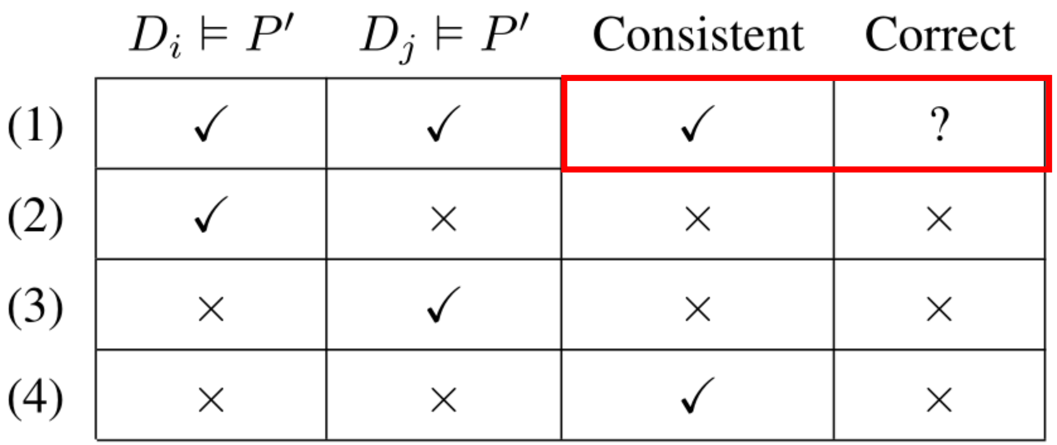
(In)correctness and (in)consistency



	$D_i \vDash P'$	$D_j \vDash P'$	Consistent	Correct
(1)	✓	✓	✓	?
(2)	✓	×	×	×
(3)	×	✓	×	×
(4)	×	×	✓	×

$$consistent(D_i, D_j) \not\Rightarrow correct(D)$$

(In)correctness and (in)consistency



	$D_i \models P'$	$D_j \models P'$	Consistent	Correct
(1)	✓	✓	✓	?
(2)	✓	×	×	×
(3)	×	✓	×	×
(4)	×	×	✓	×

$$consistent(D_i, D_j) \nRightarrow correct(D)$$

$$correct(D) \Rightarrow consistent(D_i, D_j)$$

(In)correctness and (in)consistency

	$D_i \models P'$	$D_j \models P'$	Consistent	Correct
(1)	✓	✓	✓	?
(2)	✓	×	×	×
(3)	×	✓	×	×
(4)	×	×	✓	×

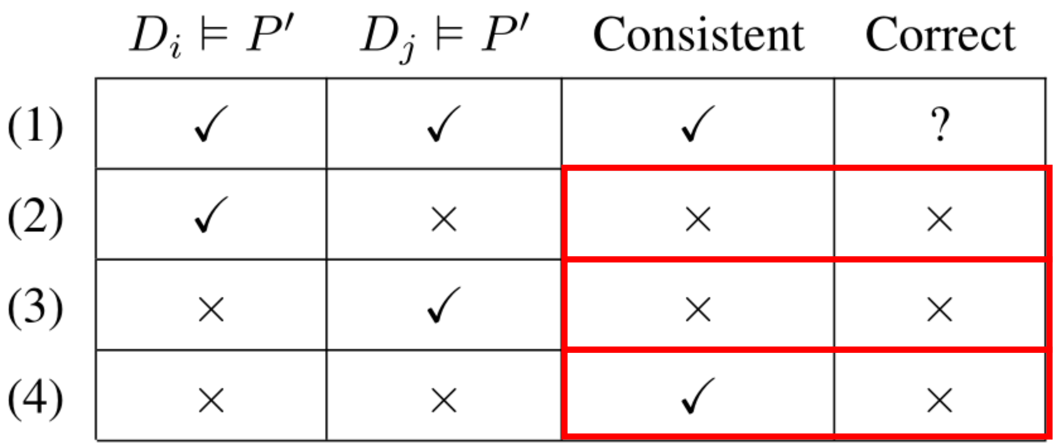
$$consistent(D_i, D_j) \nRightarrow correct(D)$$

$$correct(D) \Rightarrow consistent(D_i, D_j)$$

$$inconsistent(D_i, D_j) \Rightarrow incorrect(D)$$

$$incorrect(D) \nRightarrow inconsistent(D_i, D_j)$$

(In)correctness and (in)consistency



	$D_i \vDash P'$	$D_j \vDash P'$	Consistent	Correct
(1)	✓	✓	✓	?
(2)	✓	×	×	×
(3)	×	✓	×	×
(4)	×	×	✓	×

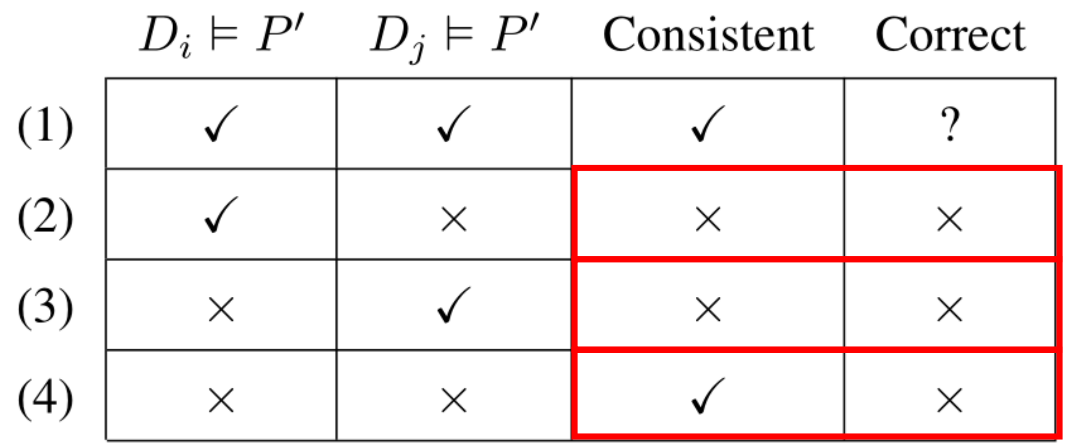
$$consistent(D_i, D_j) \nRightarrow correct(D)$$

$$correct(D) \Rightarrow consistent(D_i, D_j)$$

$$inconsistent(D_i, D_j) \Rightarrow incorrect(D)$$

$$incorrect(D) \nRightarrow inconsistent(D_i, D_j)$$

(In)correctness and (in)consistency



	$D_i \vDash P'$	$D_j \vDash P'$	Consistent	Correct
(1)	✓	✓	✓	?
(2)	✓	×	×	×
(3)	×	✓	×	×
(4)	×	×	✓	×

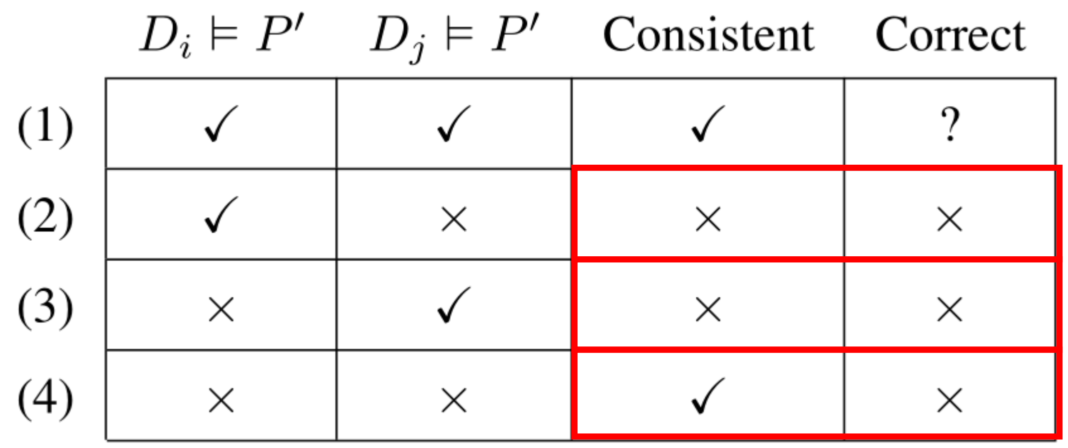
$$consistent(D_i, D_j) \not\Rightarrow correct(D)$$

$$correct(D) \Rightarrow consistent(D_i, D_j)$$

$$inconsistent(D_i, D_j) \Rightarrow incorrect(D)$$

$$incorrect(D) \not\Rightarrow inconsistent(D_i, D_j)$$

(In)correctness and (in)consistency



	$D_i \models P'$	$D_j \models P'$	Consistent	Correct
(1)	✓	✓	✓	?
(2)	✓	×	×	×
(3)	×	✓	×	×
(4)	×	×	✓	×

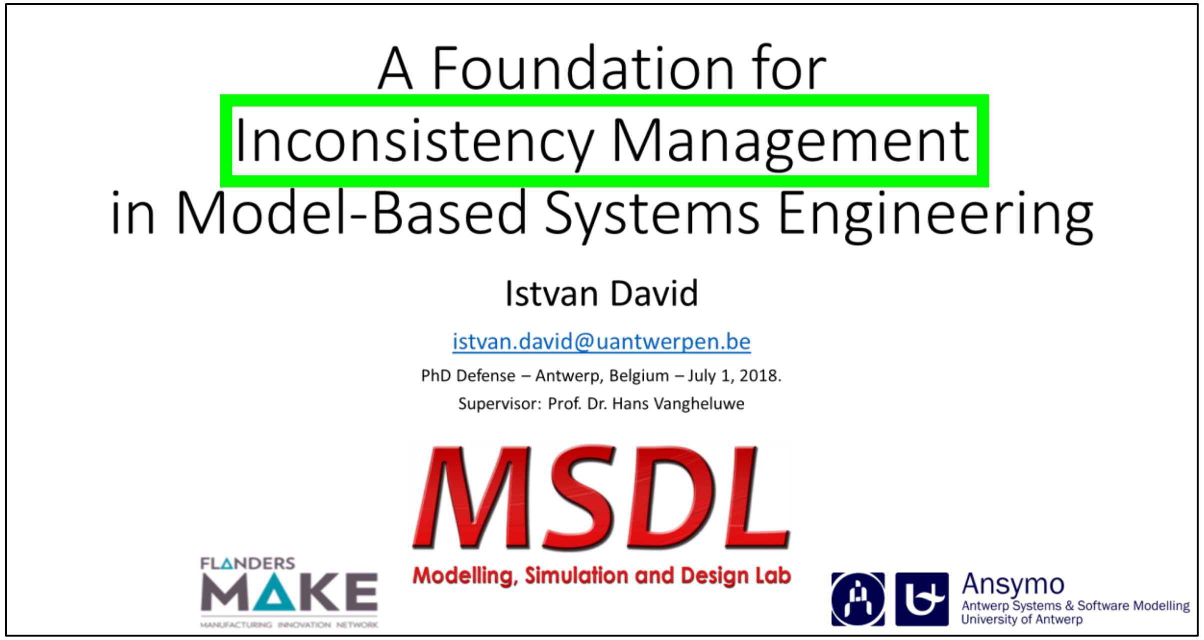
$$consistent(D_i, D_j) \nRightarrow correct(D)$$

$$correct(D) \Rightarrow consistent(D_i, D_j)$$

$$inconsistent(D_i, D_j) \Rightarrow incorrect(D)$$

$$incorrect(D) \nRightarrow inconsistent(D_i, D_j)$$

(In)correctness and (in)consistency



A Foundation for
Inconsistency Management
in Model-Based Systems Engineering

Istvan David

istvan.david@uantwerpen.be

PhD Defense – Antwerp, Belgium – July 1, 2018.

Supervisor: Prof. Dr. Hans Vangheluwe

MSDL
Modelling, Simulation and Design Lab
FLANDERS MAKE
MANUFACTURING INNOVATION NETWORK
Ansymo
Antwerp Systems & Software Modelling
University of Antwerp

$$consistent(D_i, D_j) \not\Rightarrow correct(D)$$

$$correct(D) \Rightarrow consistent(D_i, D_j)$$

$$inconsistent(D_i, D_j) \Rightarrow incorrect(D)$$

$$incorrect(D) \not\Rightarrow inconsistent(D_i, D_j)$$

Managing inconsistencies

Rather than thinking about removing inconsistency, we need to think about ***managing consistency****. – Finkelstein, 2000*

$$consistent(D_i, D_j) \nRightarrow correct(D)$$
$$inconsistent(D_i, D_j) \Rightarrow incorrect(D)$$

A. Finkelstein: „*A foolish consistency: Technical challenges in consistency management*“. In *International Conference on Database and Expert Systems Applications*, pp. 1-5. Springer, Berlin, Heidelberg, 2000.

Managing inconsistencies

Rather than thinking about removing inconsistency, we need to think about ***managing*** ~~***consistency**. – Finkelstein, 2000*~~ ***inconsistency**.* (Vangheluwe, Denil, Vanherpen, David, et.al., 2013-2019)

$$consistent(D_i, D_j) \not\Rightarrow correct(D)$$
$$inconsistent(D_i, D_j) \Rightarrow incorrect(D)$$

A. Finkelstein: „*A foolish consistency: Technical challenges in consistency management*“. In *International Conference on Database and Expert Systems Applications*, pp. 1-5. Springer, Berlin, Heidelberg, 2000.

Managing inconsistencies

Rather than thinking about removing inconsistency, we need to think about ***managing*** ~~***consistency**. – Finkelstein, 2000*~~ ***inconsistency**.* (Vangheluwe, Denil, Vanherpen, David, et.al., 2013-2019)

- K. Vanherpen: „*A contract-based approach for multi-viewpoint consistency in the concurrent design of cyber-physical systems*”, PhD Thesis, 2018.
- I. Dávid, J. Denil, H. Vangheluwe: „*Process-oriented Inconsistency Management in Collaborative Systems Modeling*”, 16th Industrial Simulation Conference, 2018.
- I. Dávid, B. Meyers, K. Vanherpen, Y. Van Tendeloo, K. Berx, H. Vangheluwe: „*Modeling and Enactment Support for Early Detection of Inconsistencies in Engineering Processes*”, 2nd International Workshop on Collaborative Modelling in MDE, 2017.
- I. Dávid, J. Denil, K. Gadeyne, H. Vangheluwe: „*Engineering Process Transformation to Manage (In)consistency*”, 1st International Workshop on Collaborative Modelling in MDE, 2016.
- I. Dávid, E. Syriani, C. Verbrugge, D. Buchs, D. Blouin, A. Cicchetti, K. Vanherpen: „*Towards Inconsistency Tolerance by Quantification of Semantic Inconsistencies*”, 1st International Workshop on Collaborative Modelling in MDE, 2016.
- I. Dávid: „*A Multi-Paradigm Modeling Foundation for Collaborative Multi-view Model/System Development*”, ACM Student Research Competition MoDELS, 2016.
- K. Vanherpen, J. Denil, I. Dávid, P. De Meulenaere, P. Mosterman, M. Törngren, A. Qamar, H. Vangheluwe: ”*Ontological Reasoning for Consistency in the Design of Cyber-Physical Systems*”, In *2016 1st International Workshop on Cyber-Physical Production Systems (CPPS)*, (April 2016).
- K. Vanherpen, J. Denil, P. De Meulenaere, H. Vangheluwe: „*Ontological Reasoning as an Enabler of Contract-Based Co-Design*”, In *2016 6th International Workshop on Cyber Physical Systems. Design, Modeling, and Evaluation (CyPhy)*, (October 2016).
- I. Dávid, J. Denil, H. Vangheluwe: „*Patterns of inconsistency management in mechatronics - A survey*”, Technical report, 2015.
- I. Dávid, J. Denil, H. Vangheluwe: „*Towards Inconsistency Management by Process-Oriented Dependency Modeling*”, 9th International Workshop on Multi-Paradigm Modeling, 2015.

A. Finkelstein: „*A foolish consistency: Technical challenges in consistency management*“. In *International Conference on Database and Expert Systems Applications*, pp. 1-5. Springer, Berlin, Heidelberg, 2000.

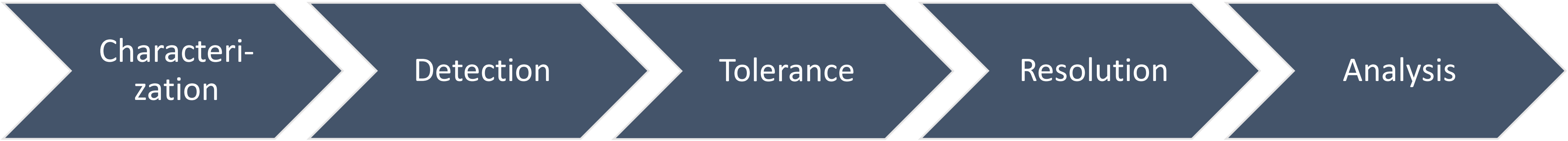
Managing inconsistencies

*Rather than thinking about removing inconsistency, we need to think about **managing** ~~**consistency**. – Finkelstein, 2000~~* ***inconsistency***. (Vangheluwe, Denil, Vanherpen, David, et.al., 2013-2019)

- Prevention (e.g. contract-based design)
- Detect-and-Resolve

VANHERPEN, 2018

DAVID, 2019



K. Vanherpen: „*A contract-based approach for multi-viewpoint consistency in the concurrent design of cyber-physical systems*”. PhD Thesis, 2018.

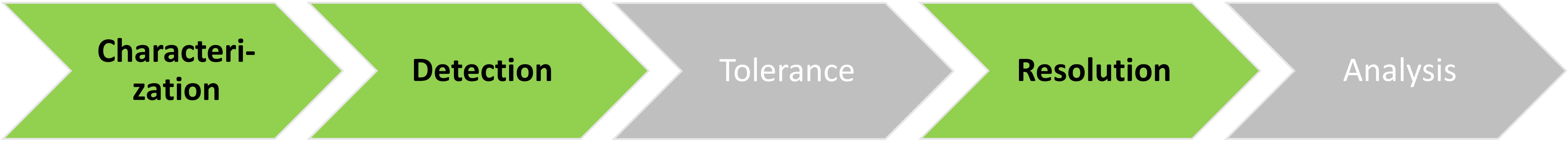
Managing inconsistencies

*Rather than thinking about removing inconsistency, we need to think about **managing** ~~**consistency**. – Finkelstein, 2000~~ **inconsistency**.* (Vangheluwe, Denil, Vanherpen, David, et.al., 2013-2019)

- Prevention (e.g. contract-based design)
- Detect-and-Resolve

VANHERPEN, 2018

DAVID, 2019



K. Vanherpen: „*A contract-based approach for multi-viewpoint consistency in the concurrent design of cyber-physical systems*”. PhD Thesis, 2018.

Managing inconsistencies

Rather than thinking about removing inconsistency, we ne *to thin* *out* ***aging*** ~~***consistency.***~~ ~~*– Finkelstein, 2000*~~ ***inconsistency***. (Va avid et.al., 2013-2019)

- Prevention (e.g. contract-based design)
- Detect-and-Resolve

VANHE

DAVID,

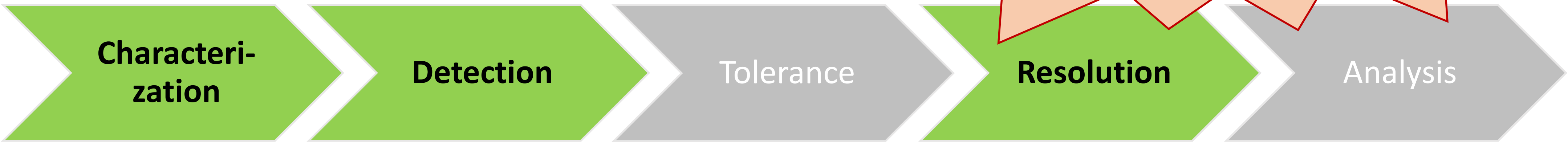
Which one to employ?

When to initiate?

What scope?

....

?



Managing inconsistencies

Rather than thinking about removing inconsistency, we ne[...] *to thin*[...] *out* [...]***aging*** ~~***consistency****. – Finkelstein, 2000*~~ ***inconsistency****.* (Va [...] David et.al., 2013-2019)

- Prevention (e.g. contract-based design)
- Detect-and-Resolve

VANHE

DAVID,

EFFICIENCY!

…of the overall engineering endeavor.

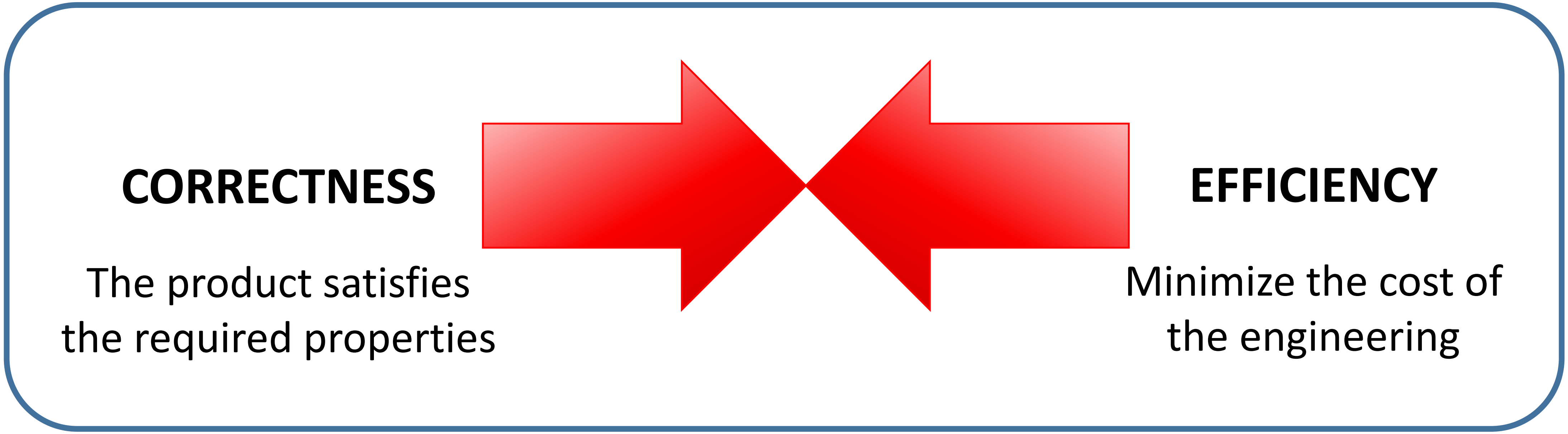
Characteri-zation

De

Resolution

Analysis

„...of the overall engineering endeavor”



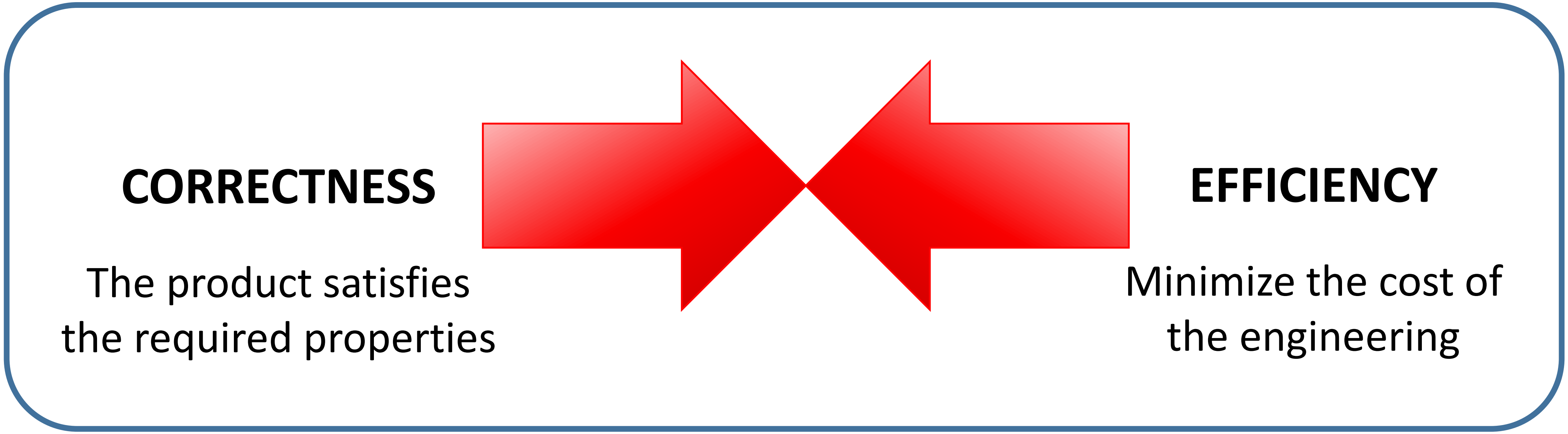
CORRECTNESS

The product satisfies
the required properties

EFFICIENCY

Minimize the cost of
the engineering

PROCESS



CORRECTNESS

The product satisfies the required properties

EFFICIENCY

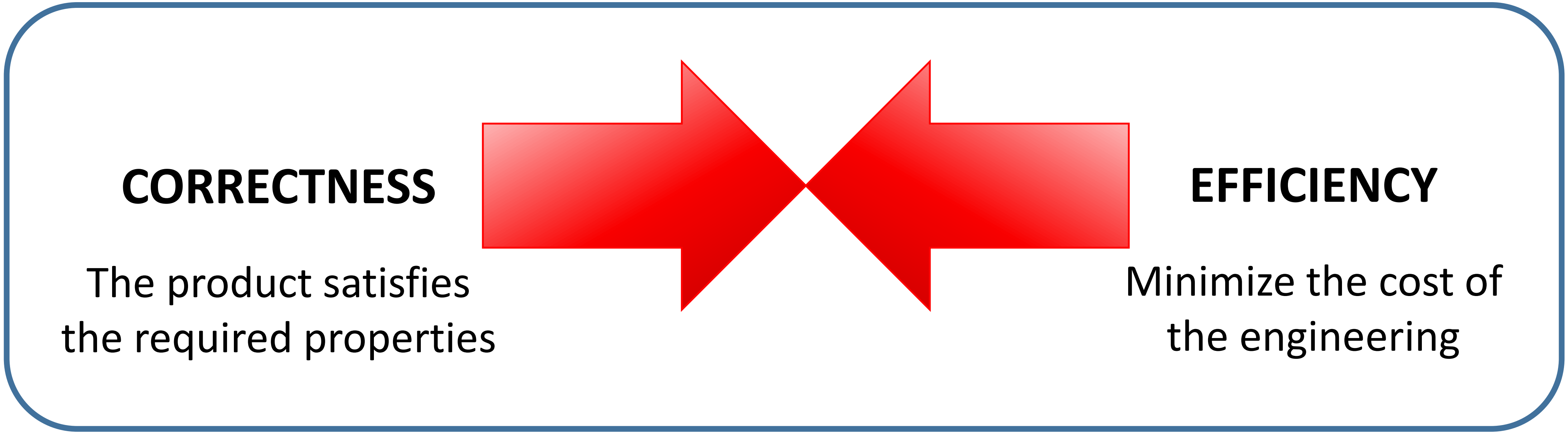
Minimize the cost of the engineering

Partially ordered set of activities,

manipulating artifacts (models)

Activity: manual vs automated

PROCESS



CORRECTNESS

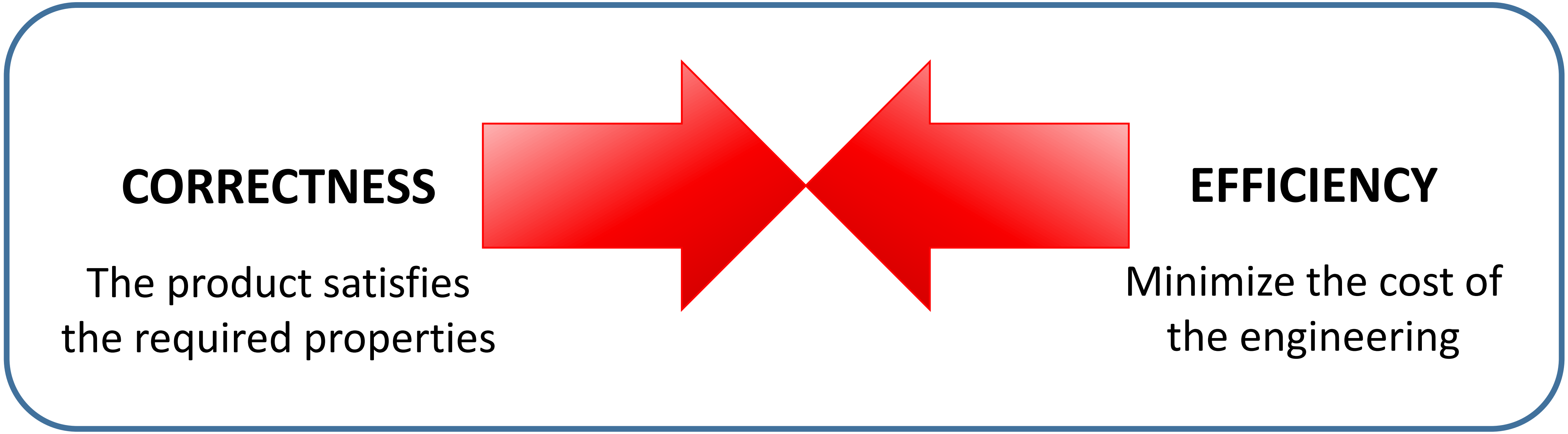
The product satisfies
the required properties

EFFICIENCY

Minimize the cost of
the engineering

PROCESS

Optimize so that (i) a correct product is delivered; and (ii) the process is as efficient as possible.



CORRECTNESS

The product satisfies the required properties

EFFICIENCY

Minimize the cost of the engineering

$$inconsistent(D_i, D_j) \Rightarrow incorrect(D)$$

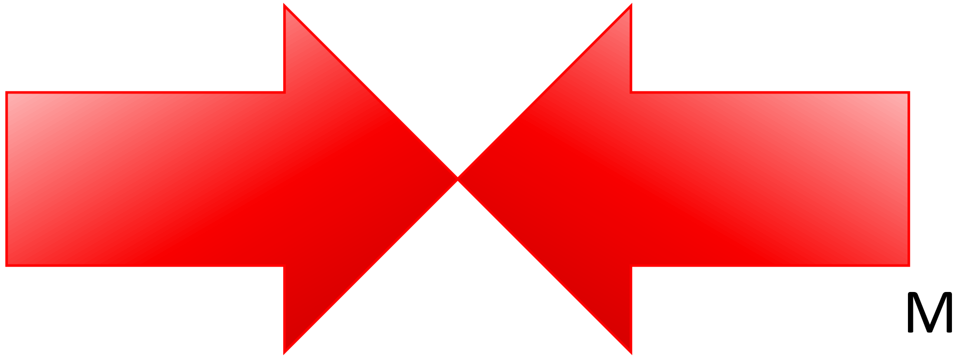
$$consistent(D_i, D_j) \nRightarrow correct(D)$$

PROCESS

Optimize so that (i) a correct product is delivered; and (ii) the process is as efficient as possible.

CORRECTNESS

The product satisfies the required properties



EFFICIENCY

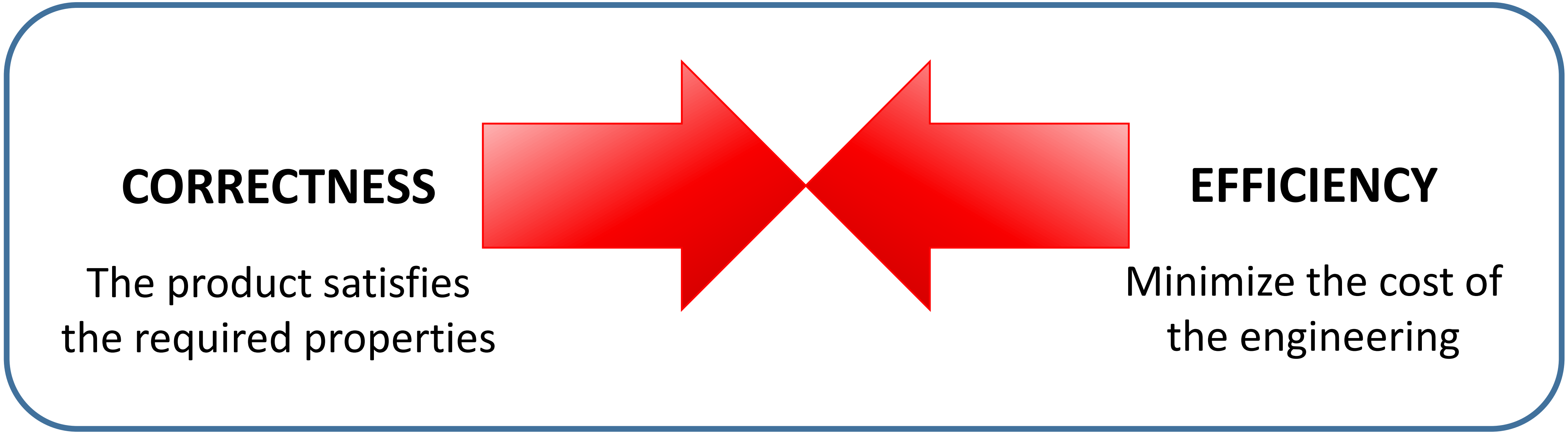
Minimize the cost of the engineering

$$inconsistent(D_i, D_j) \Rightarrow incorrect(D)$$

$$consistent(D_i, D_j) \not\Rightarrow correct(D)$$

PROCESS

Optimize so that (i) a correct product is delivered; and (ii) the process is as efficient as possible.



CORRECTNESS

The product satisfies the required properties

EFFICIENCY

Minimize the cost of the engineering

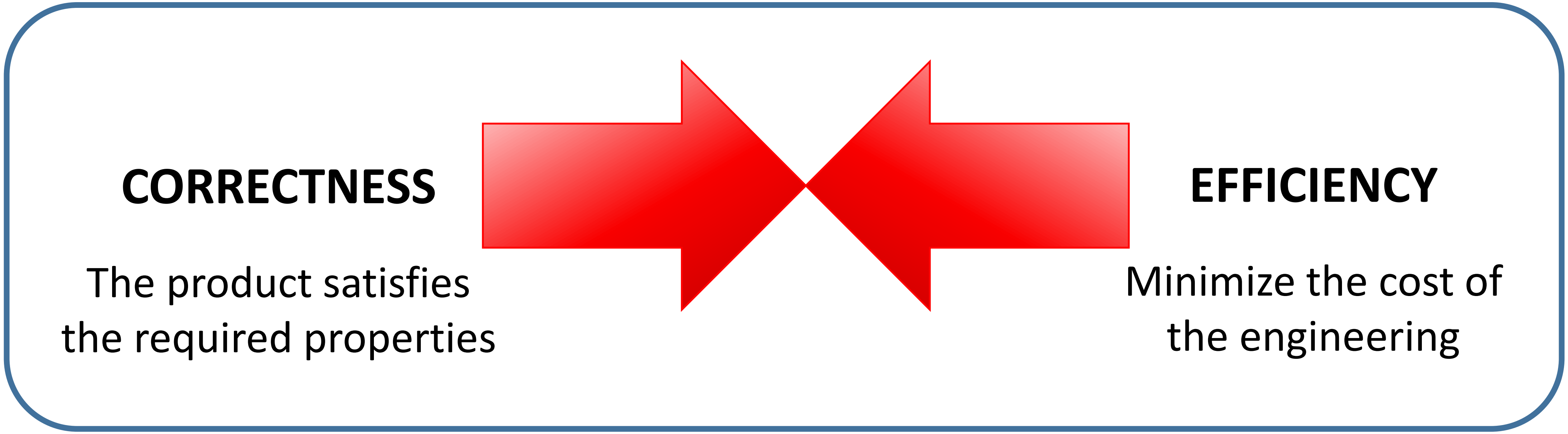
Heuristic (Romanycia and Pelletier). Any device, be it a program, rule, piece of knowledge, etc., which one is not entirely confident will be useful in providing a practical solution, but which one has reason to believe will be useful, and which is added to a problem-solving system in expectation that on average the performance will improve.

$$\mathit{inconsistent}(D_i, D_j) \Rightarrow \mathit{incorrect}(D)$$

$$\mathit{consistent}(D_i, D_j) \not\Rightarrow \mathit{correct}(D)$$

PROCESS

Optimize so that (i) a correct product is delivered; and (ii) the process is as efficient as possible.



CORRECTNESS

The product satisfies the required properties

EFFICIENCY

Minimize the cost of the engineering

MODEL (IN)CONSISTENCY

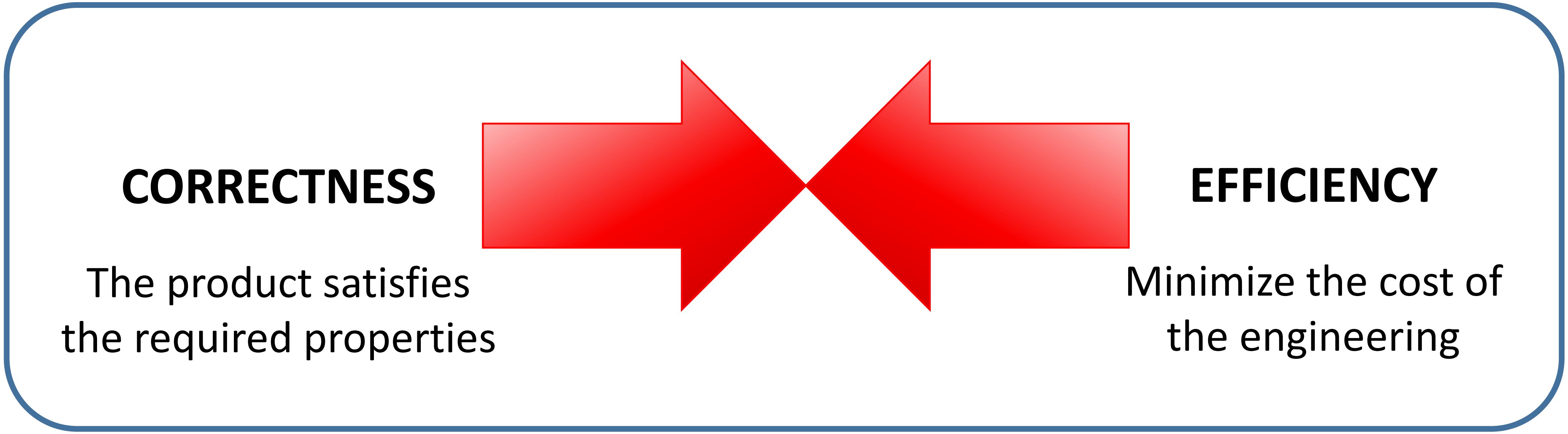
Heuristic (Romanycia and Pelletier). Any device, be it a program, rule, piece of knowledge, etc., which one is not entirely confident will be useful in providing a practical solution, but which one has reason to believe will be useful, and which is added to a problem-solving system in expectation that on average the performance will improve.

$$inconsistent(D_i, D_j) \Rightarrow incorrect(D)$$

$$consistent(D_i, D_j) \not\Rightarrow correct(D)$$

PROCESS

Optimize so that (i) a correct product is delivered; and (ii) the process is as efficient as possible.



CORRECTNESS

The product satisfies the required properties

EFFICIENCY

Minimize the cost of the engineering

MODEL (IN)CONSISTENCY

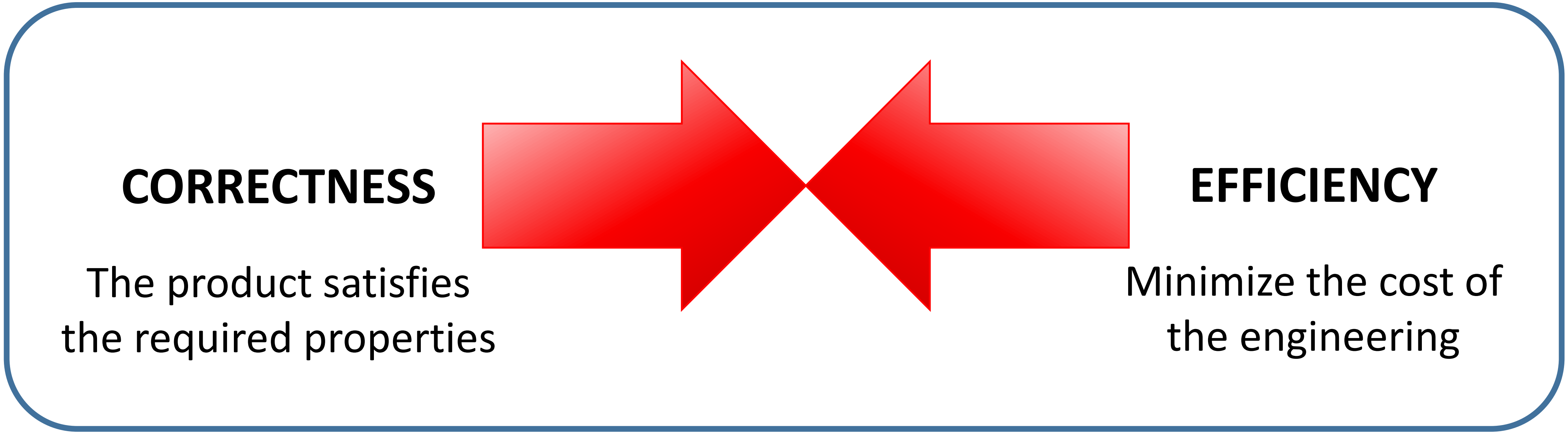
Heuristic (Romanycia and Pelletier). Any **device**, be it a program, rule, piece of knowledge, etc., which one is not entirely confident will be useful in providing a practical solution, but which one has reason to believe will be useful, and which is added to a problem-solving system in expectation that on average the performance will improve.

$$inconsistent(D_i, D_j) \Rightarrow incorrect(D)$$

$$consistent(D_i, D_j) \not\Rightarrow correct(D)$$

PROCESS

Optimize so that (i) a correct product is delivered; and (ii) the process is as efficient as possible.



CORRECTNESS

The product satisfies the required properties

EFFICIENCY

Minimize the cost of the engineering

MODEL (IN)CONSISTENCY

Heuristic (Romanycia and Pelletier). Any **device**, be it a program, rule, piece of knowledge, etc., which one is not entirely confident will be useful in providing a **practical solution**, but which one has reason to believe will be useful, and which is added to a problem-solving system in expectation that on average the performance will improve.

$$inconsistent(D_i, D_j) \Rightarrow incorrect(D)$$

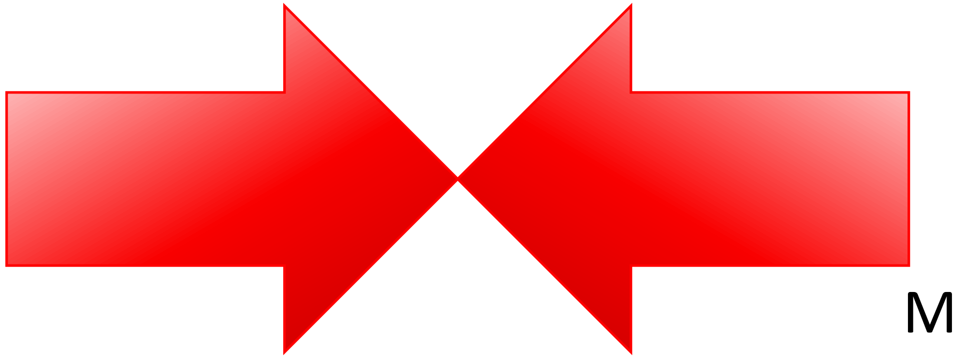
$$consistent(D_i, D_j) \nRightarrow correct(D)$$

PROCESS

Optimize so that (i) a correct product is delivered; and (ii) the process is as efficient as possible.

CORRECTNESS

The product satisfies the required properties



EFFICIENCY

Minimize the cost of the engineering

MODEL (IN)CONSISTENCY

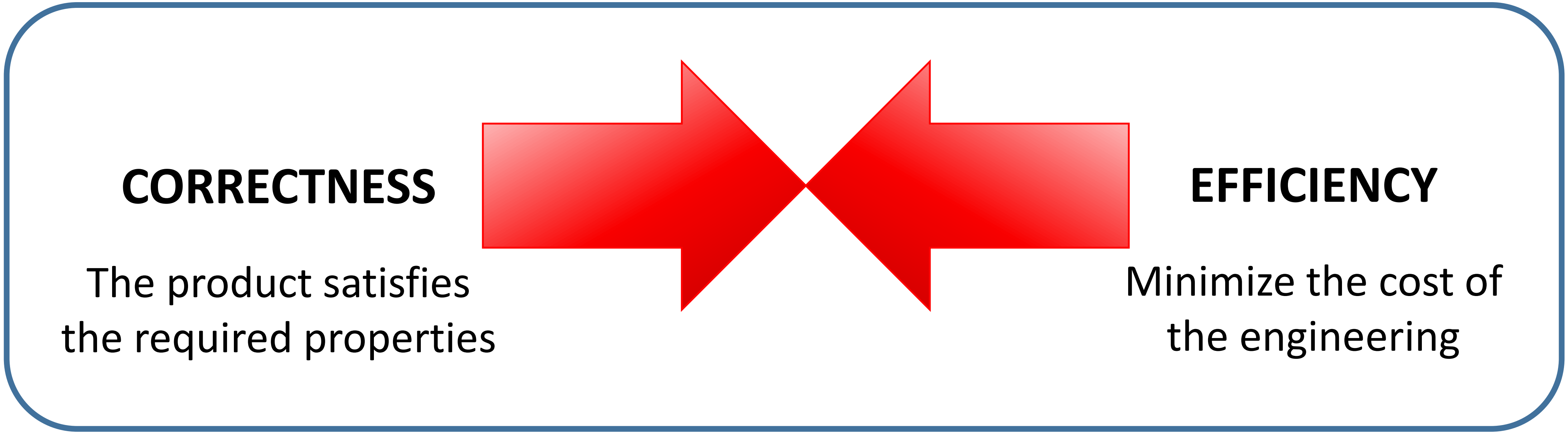
Heuristic (Romanycia and Pelletier). Any **device**, be it a program, rule, piece of knowledge, etc., which one is **not entirely confident** will be useful in providing a **practical solution**, but which one has reason to believe will be useful, and which is added to a problem-solving system in expectation that on average the performance will improve.

$$inconsistent(D_i, D_j) \Rightarrow incorrect(D)$$

$$consistent(D_i, D_j) \nRightarrow correct(D)$$

PROCESS

Optimize so that (i) a correct product is delivered; and (ii) the process is as efficient as possible.



CORRECTNESS

The product satisfies the required properties

EFFICIENCY

Minimize the cost of the engineering

MODEL (IN)CONSISTENCY

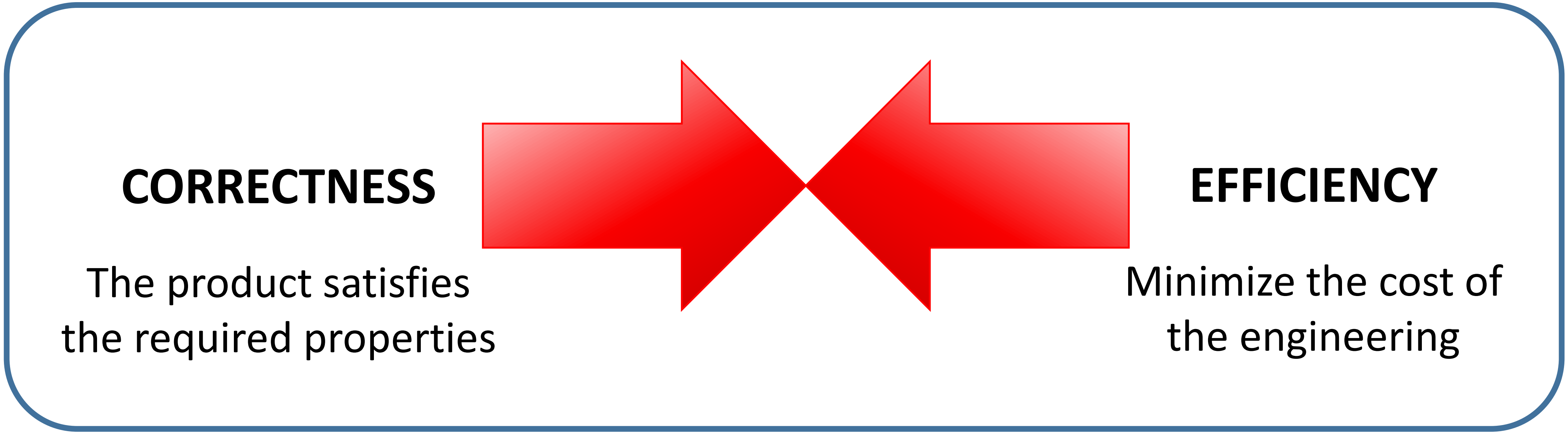
Heuristic (Romanycia and Pelletier). Any **device**, be it a program, rule, piece of knowledge, etc., which one is **not entirely confident** will be useful in providing a **practical solution**, but which one has reason to believe will be useful, and which is added to a **problem-solving system** in expectation that on average the performance will improve.

$$inconsistent(D_i, D_j) \Rightarrow incorrect(D)$$

$$consistent(D_i, D_j) \not\Rightarrow correct(D)$$

PROCESS

Optimize so that (i) a correct product is delivered; and (ii) the process is as efficient as possible.



CORRECTNESS

The product satisfies the required properties

EFFICIENCY

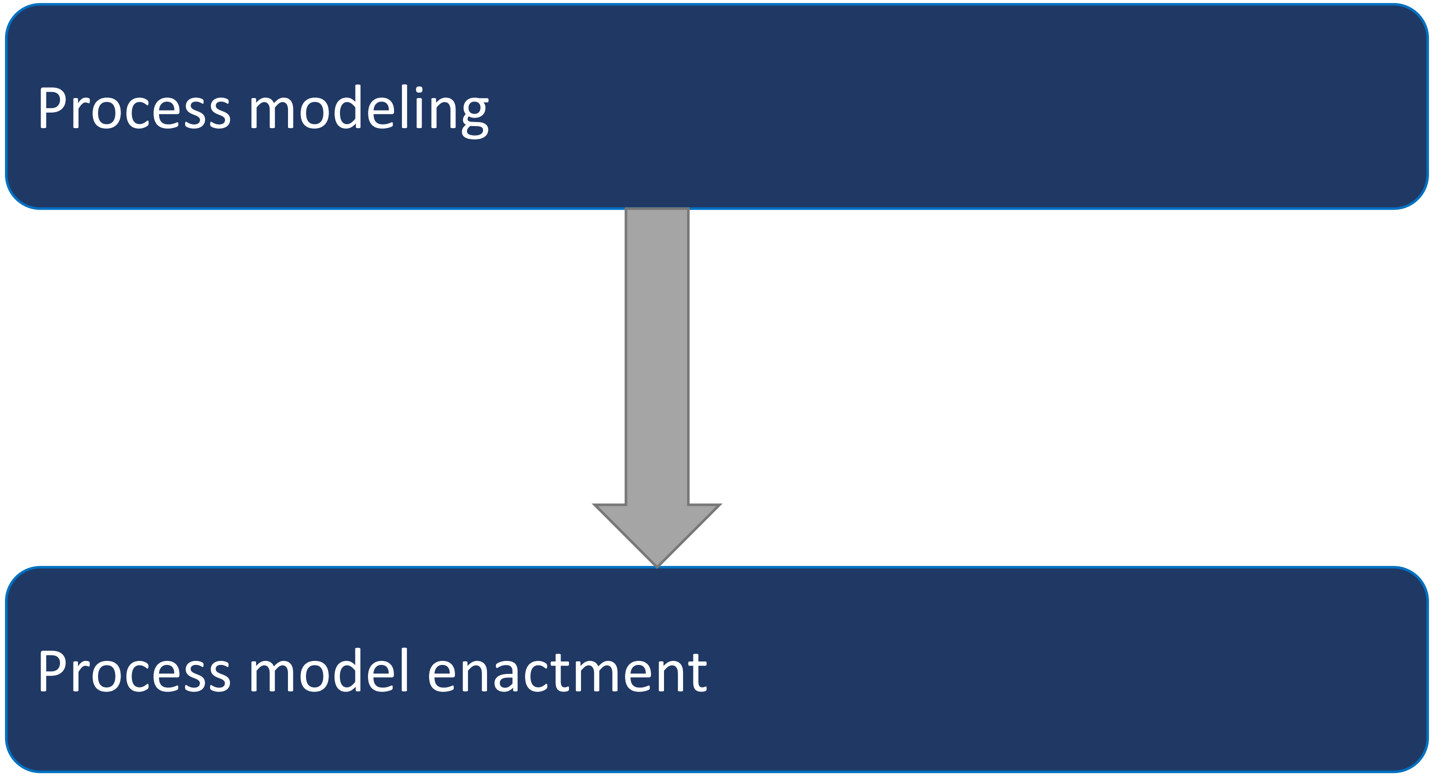
Minimize the cost of the engineering

MODEL (IN)CONSISTENCY

Heuristic (Romanycia and Pelletier). Any **device**, be it a program, rule, piece of knowledge, etc., which one is **not entirely confident** will be useful in providing a **practical solution**, but which one has reason to believe will be useful, and which is added to a **problem-solving system** in expectation that on average **the performance will improve**.

Approach

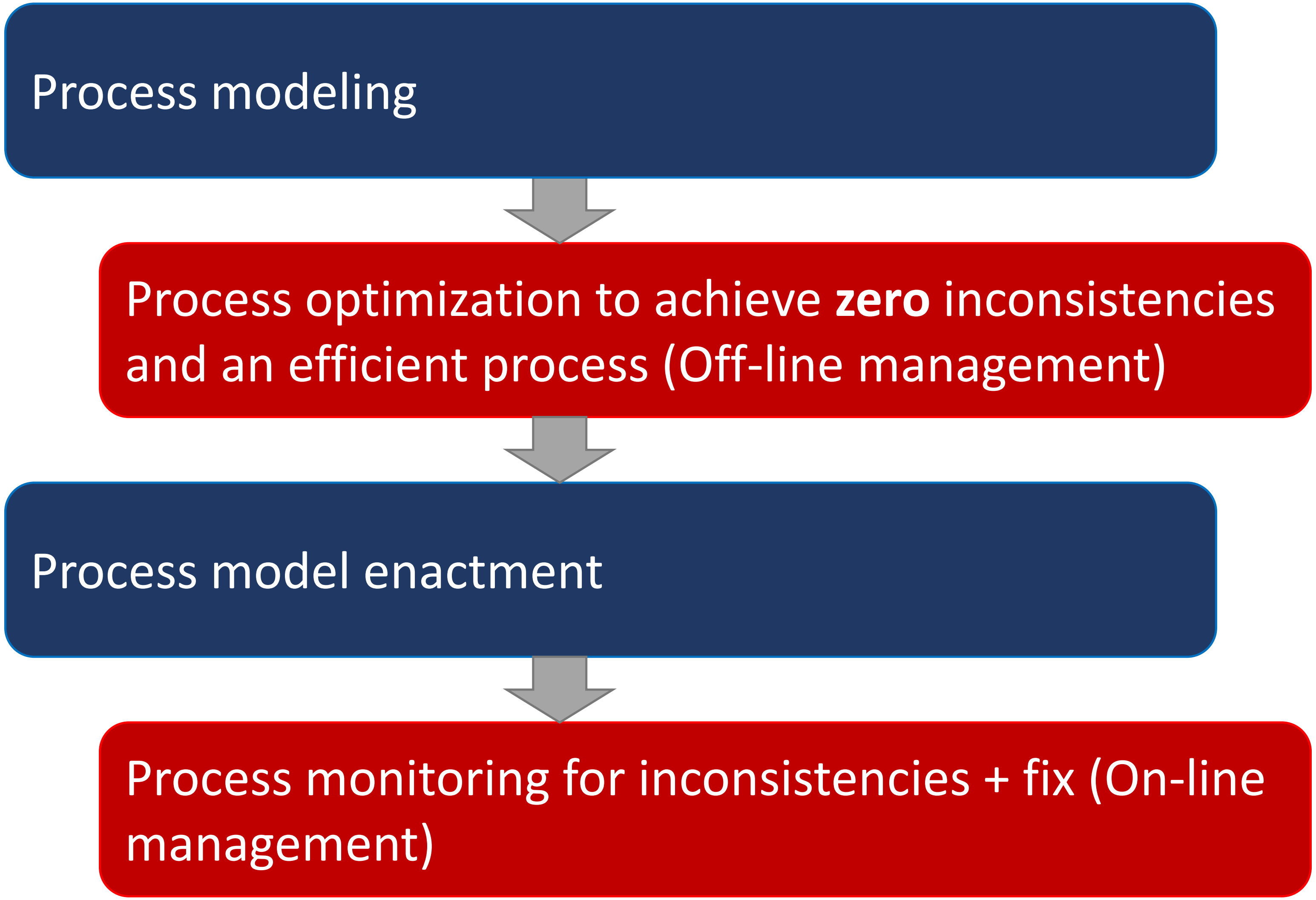
Approach



Process modeling

Process model enactment

Approach



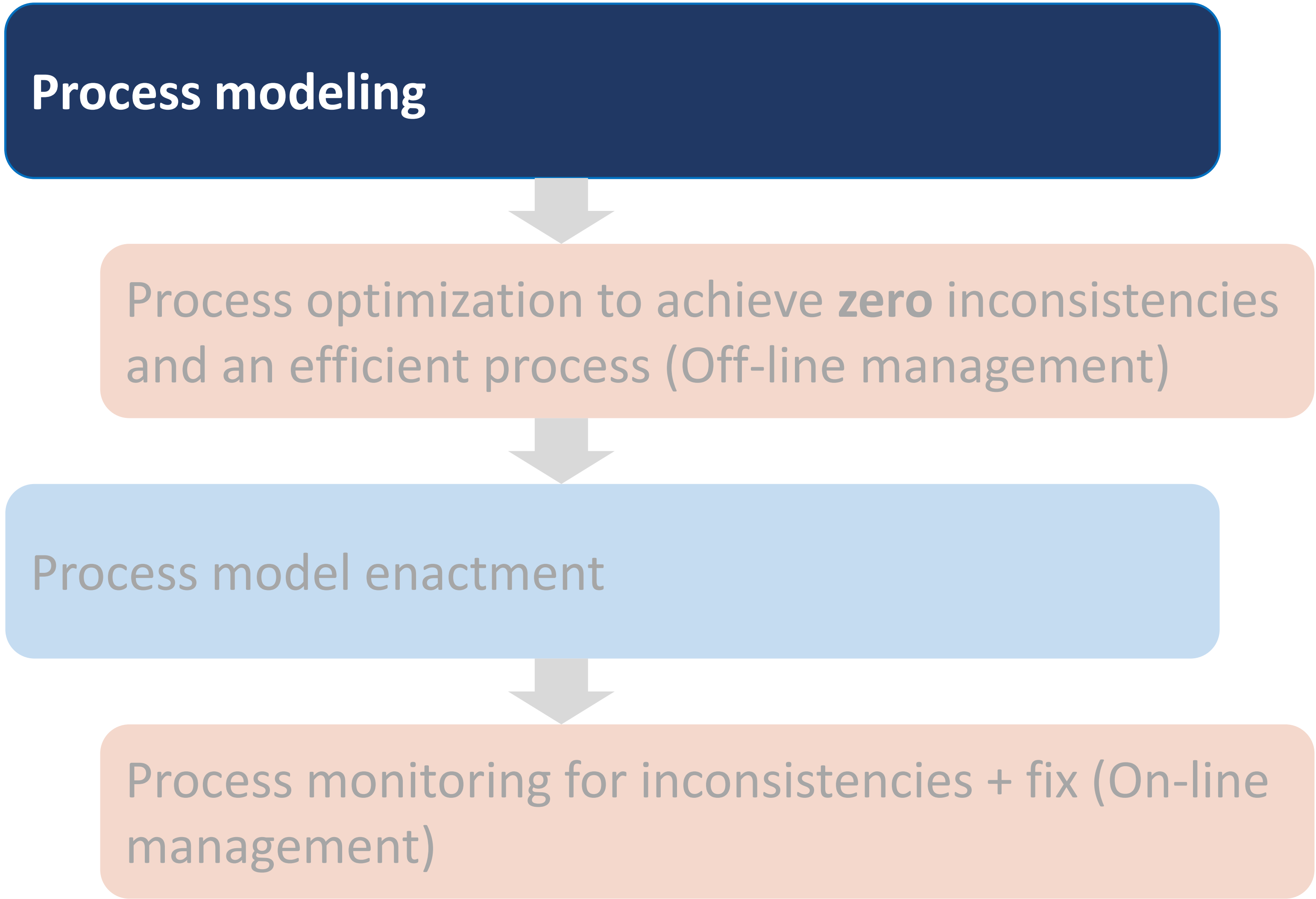
Process modeling

Process optimization to achieve **zero** inconsistencies and an efficient process (Off-line management)

Process model enactment

Process monitoring for inconsistencies + fix (On-line management)

Approach

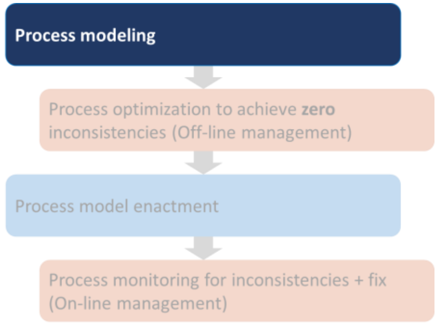


Process modeling

Process optimization to achieve **zero** inconsistencies and an efficient process (Off-line management)

Process model enactment

Process monitoring for inconsistencies + fix (On-line management)



Process modeling

Process optimization to achieve **zero** inconsistencies (Off-line management)

Process model enactment

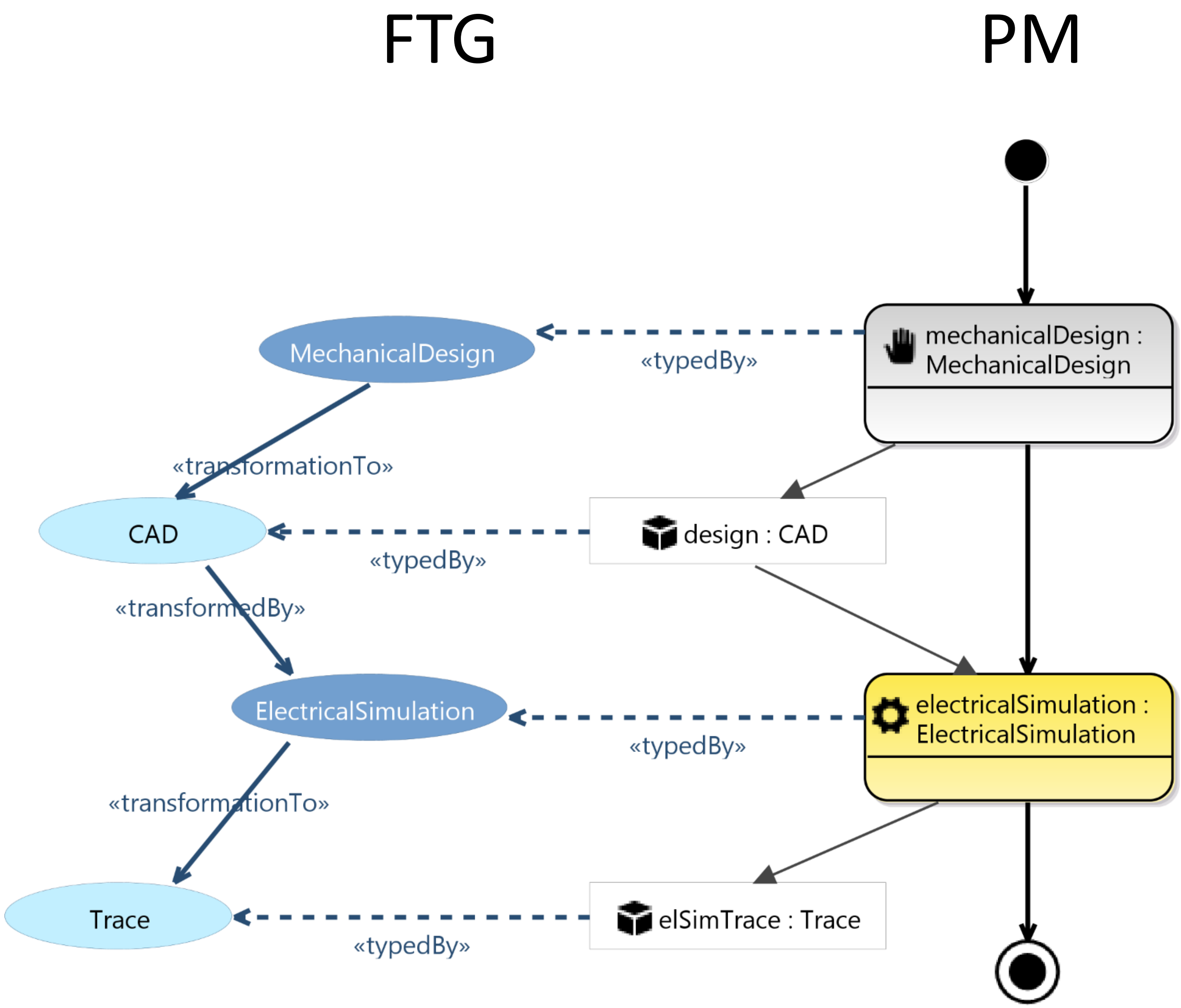
Process monitoring for inconsistencies + fix (On-line management)

Process modeling

- Built on the *Formalism Transformation Graph + Process Model* (FTG+PM) framework
- Modified to be able to capture relevant system information for inconsistency management purposes
  - Still conform to the original FTG+PM specification
- Constructed an advanced modeling tool
  - Eclipse-based
  - Various modeling interfaces (graphical and table-based)

Lúcio, Levi, Sadaf Mustafiz, Joachim Denil, Hans Vangheluwe, and Maris Jukss: „FTG+ PM: an integrated framework for investigating model transformation chains”. In *International SDL Forum*, pp. 182-202. Springer, Berlin, Heidelberg, 2013.

Process modeling

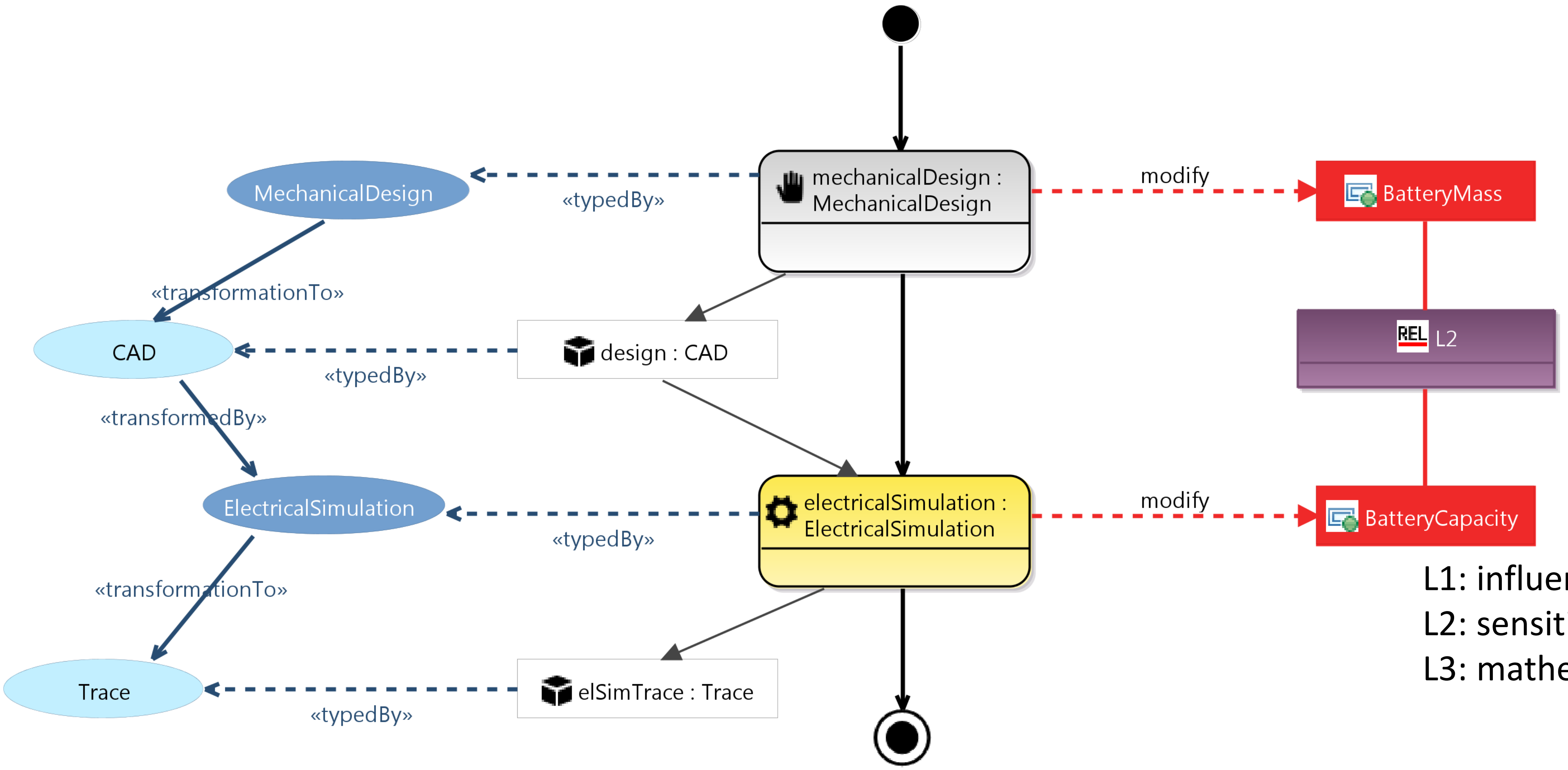


Process modeling

FTG

PM

System characteristics



L1: influence graph
L2: sensitivity
L3: mathematical relationship

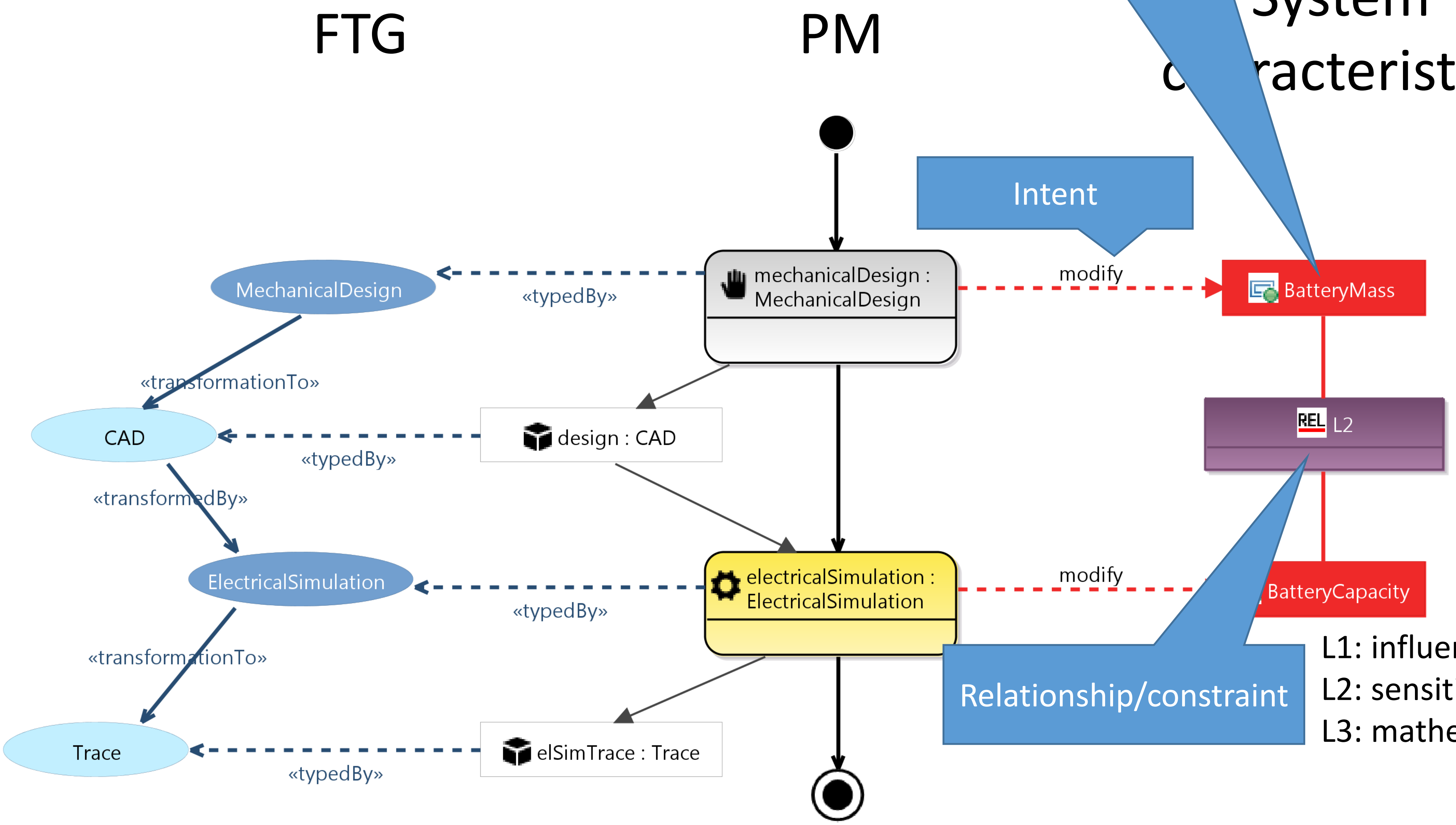
Process modeling

Process optimization to achieve **zero** inconsistencies (Off-line management)

Process model enactment

Process monitoring for inconsistencies + fix (On-line management)

Process modeling



System attribute (conceptually: an extra artifact in the FTG+PM)

Intent

Relationship/constraint

L1: influence graph
L2: sensitivity
L3: mathematical relationship

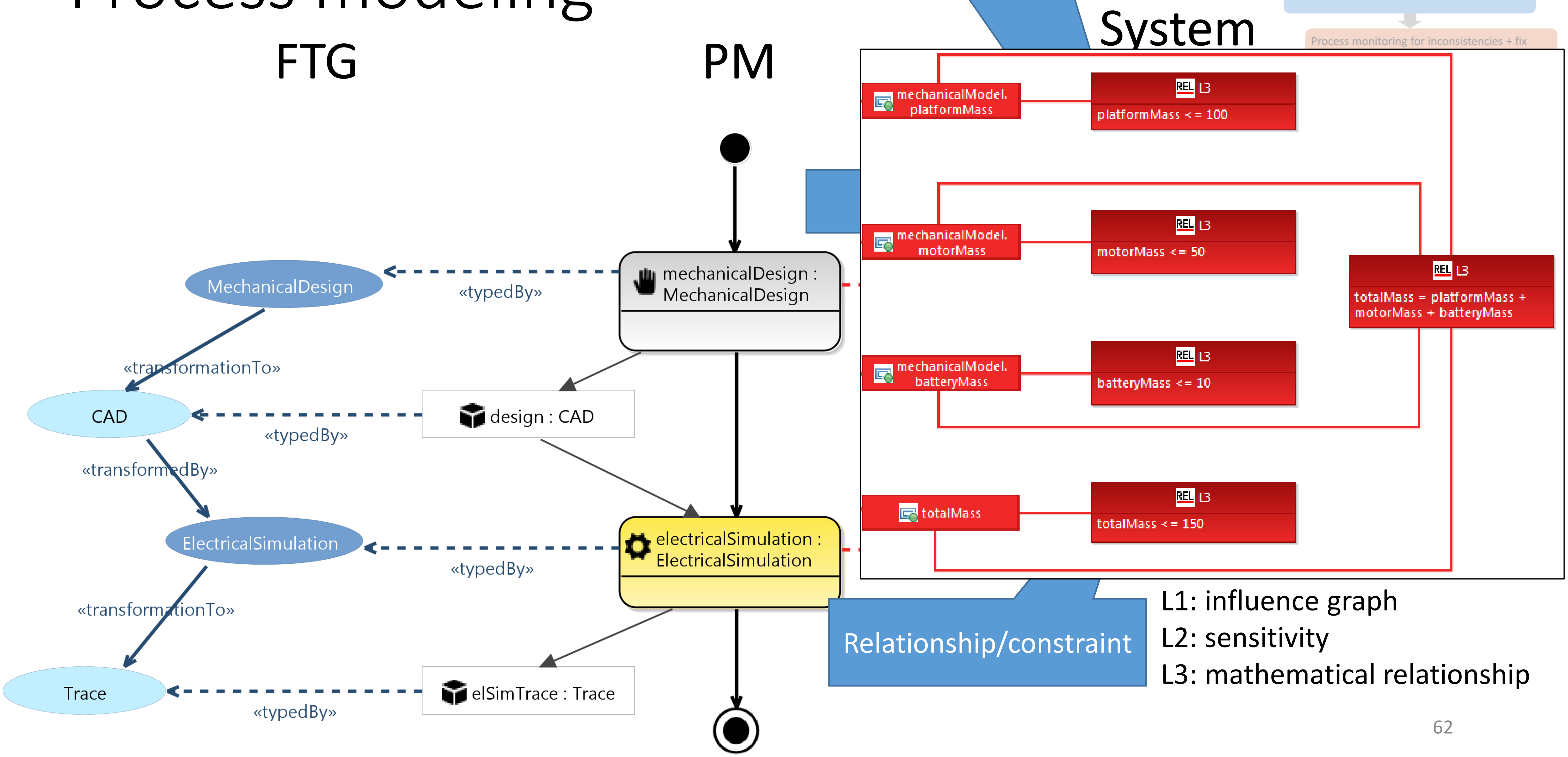
Process modeling

Process optimization to achieve **zero** inconsistencies (Off-line management)

Process model enactment

Process monitoring for inconsistencies + fix

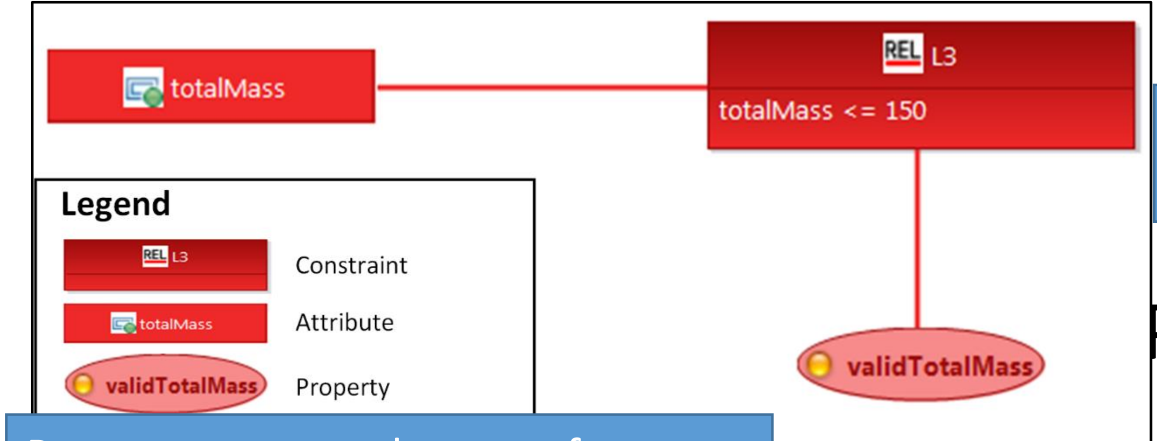
Process modeling



System attribute (conceptually: an extra artifact in the FTG+PM)

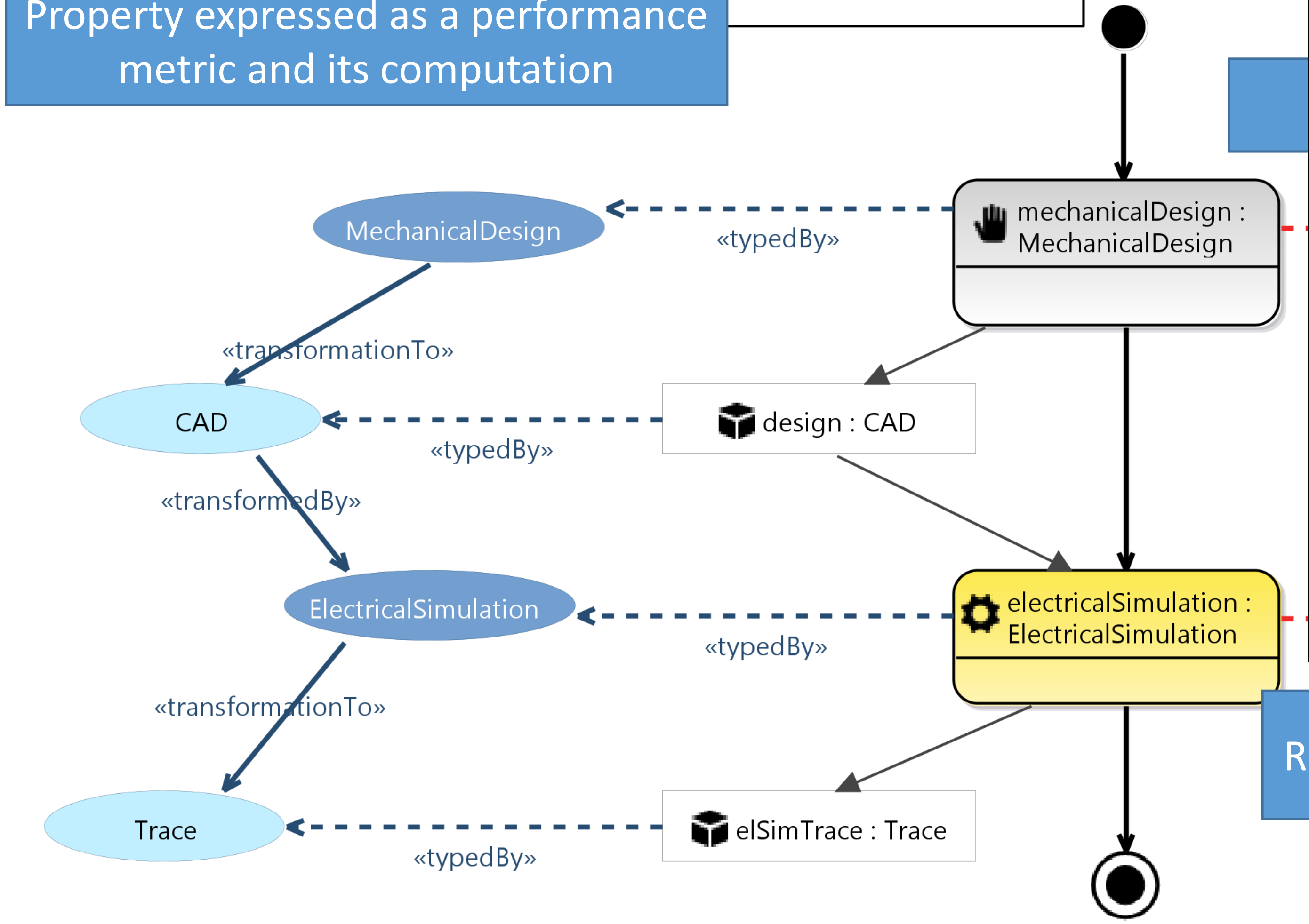
Relationship/constraint

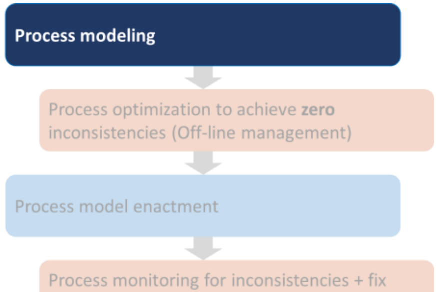
L1: influence graph
L2: sensitivity
L3: mathematical relationship



Property expressed as a performance metric and its computation

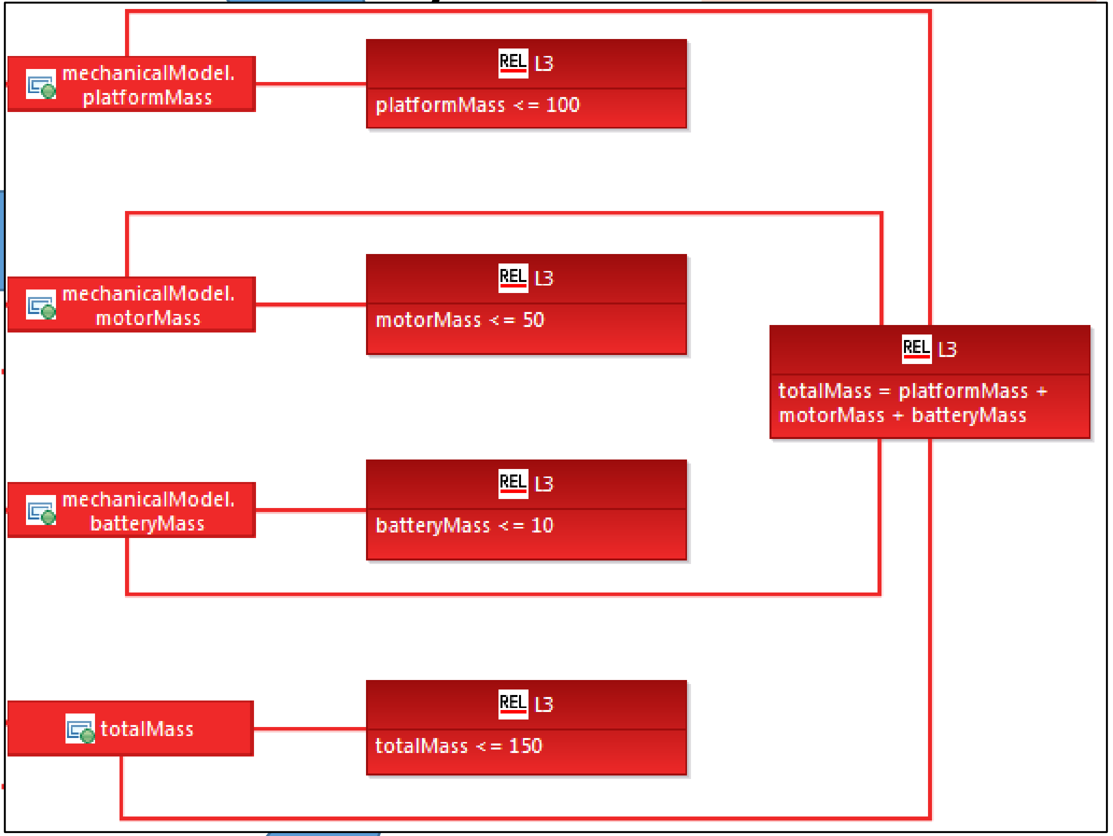
PM




Process modeling
Process optimization to achieve zero inconsistencies (Off-line management)
Process model enactment
Process monitoring for inconsistencies + fix

System attribute (conceptually: an extra artifact in the FTG+PM)

System


mechanicalModel.platformMass
REL L3
platformMass <= 100
mechanicalModel.motorMass
REL L3
motorMass <= 50
REL L3
totalMass = platformMass + motorMass + batteryMass
mechanicalModel.batteryMass
REL L3
batteryMass <= 10
totalMass
REL L3
totalMass <= 150

Relationship/constraint

L1: influence graph
L2: sensitivity
L3: mathematical relationship

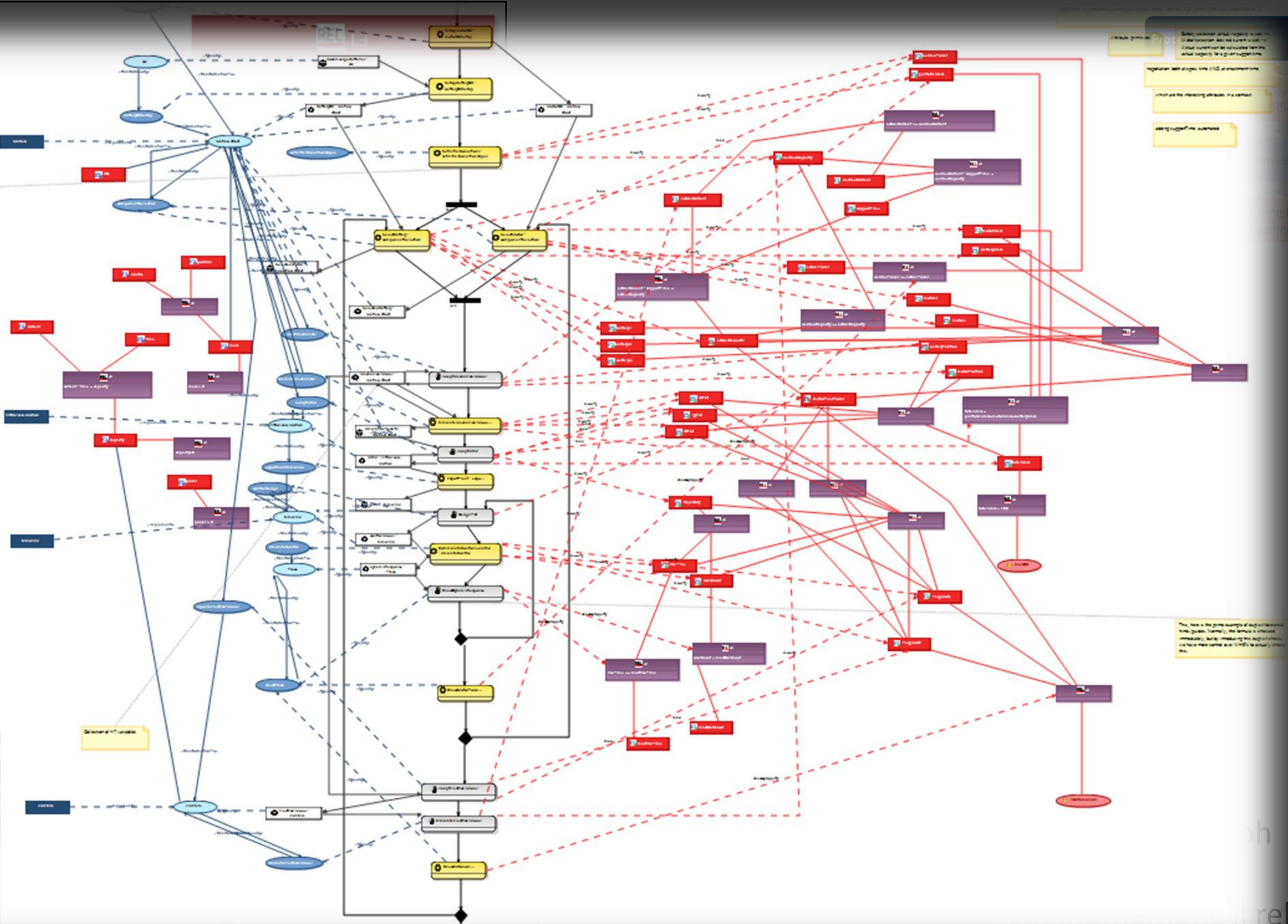
Legend

totalM

validTotal

Property

met



…on to achieve **zero** …ff-line management)

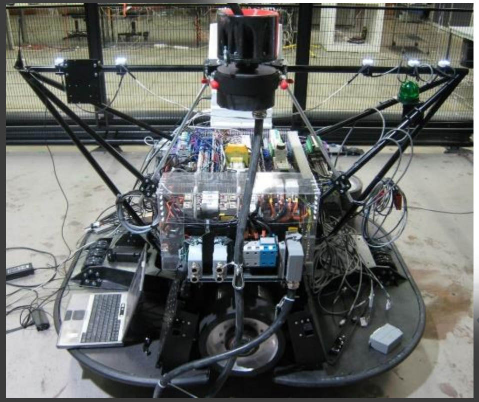
…ment

…ring for inconsistencies + fix …ement)

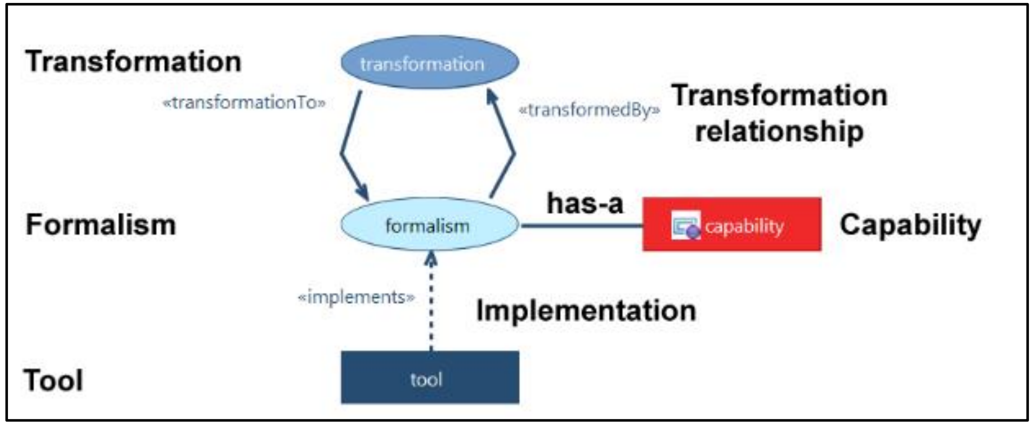
«typedBy»

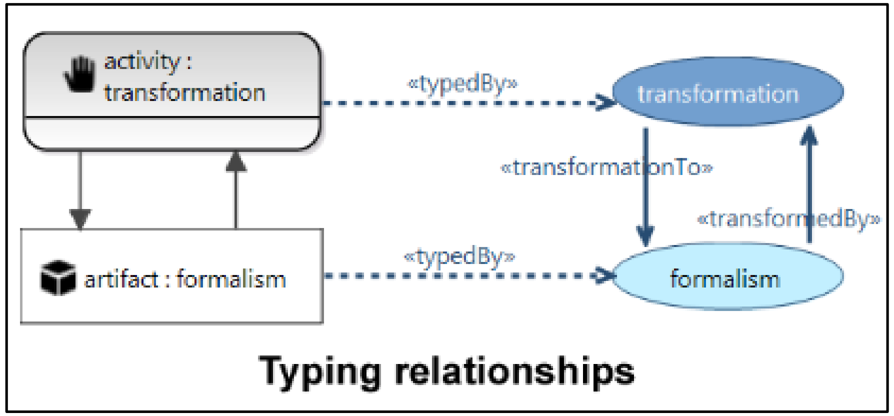
elSimTrace : Trace

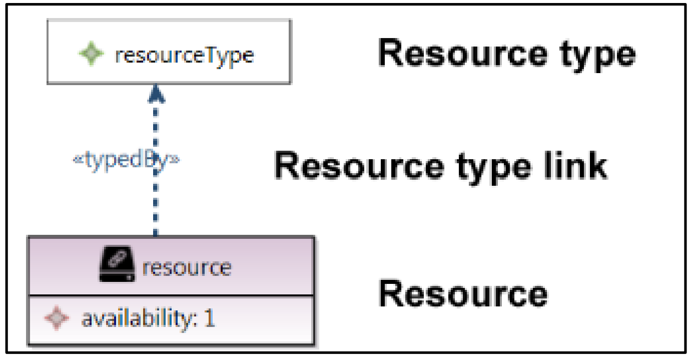
relationship

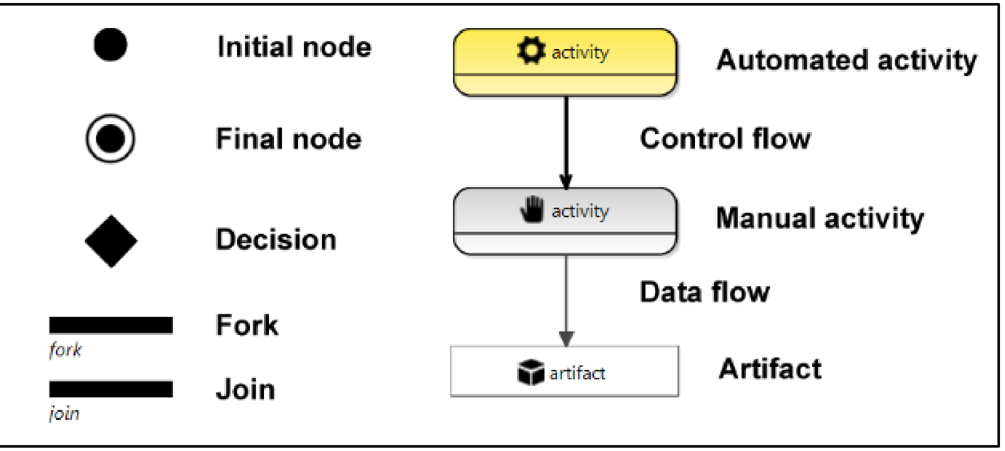


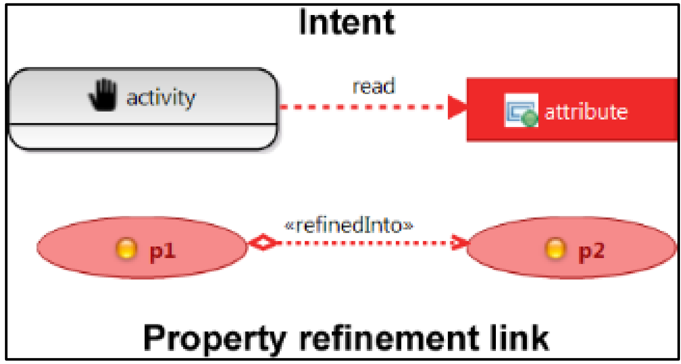
Elements of the formalism

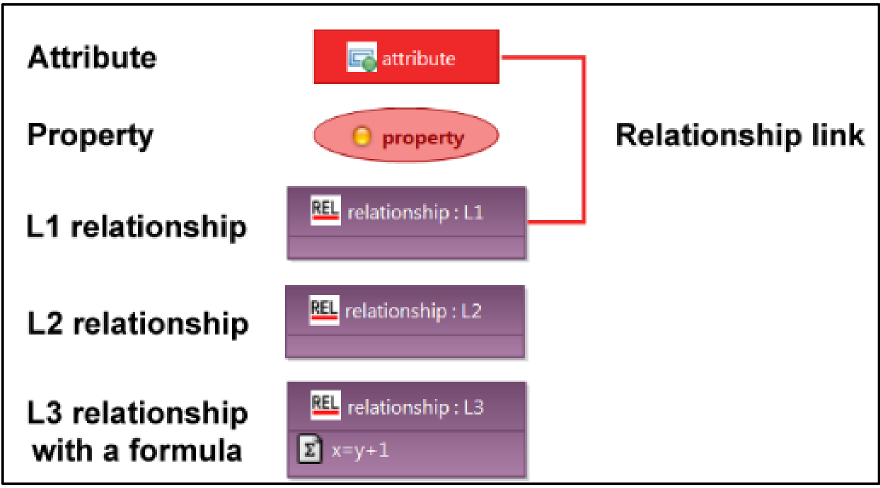




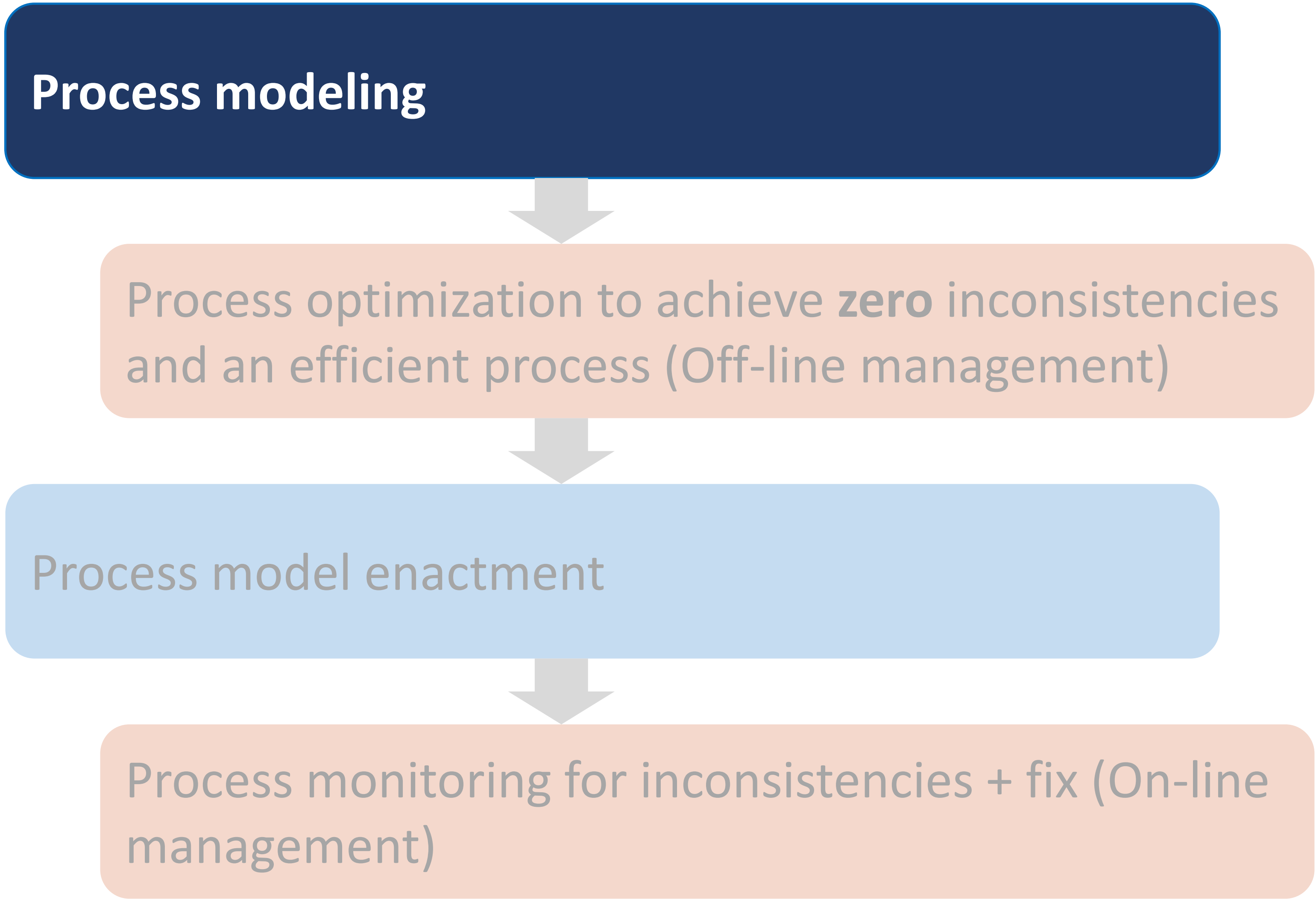








Approach



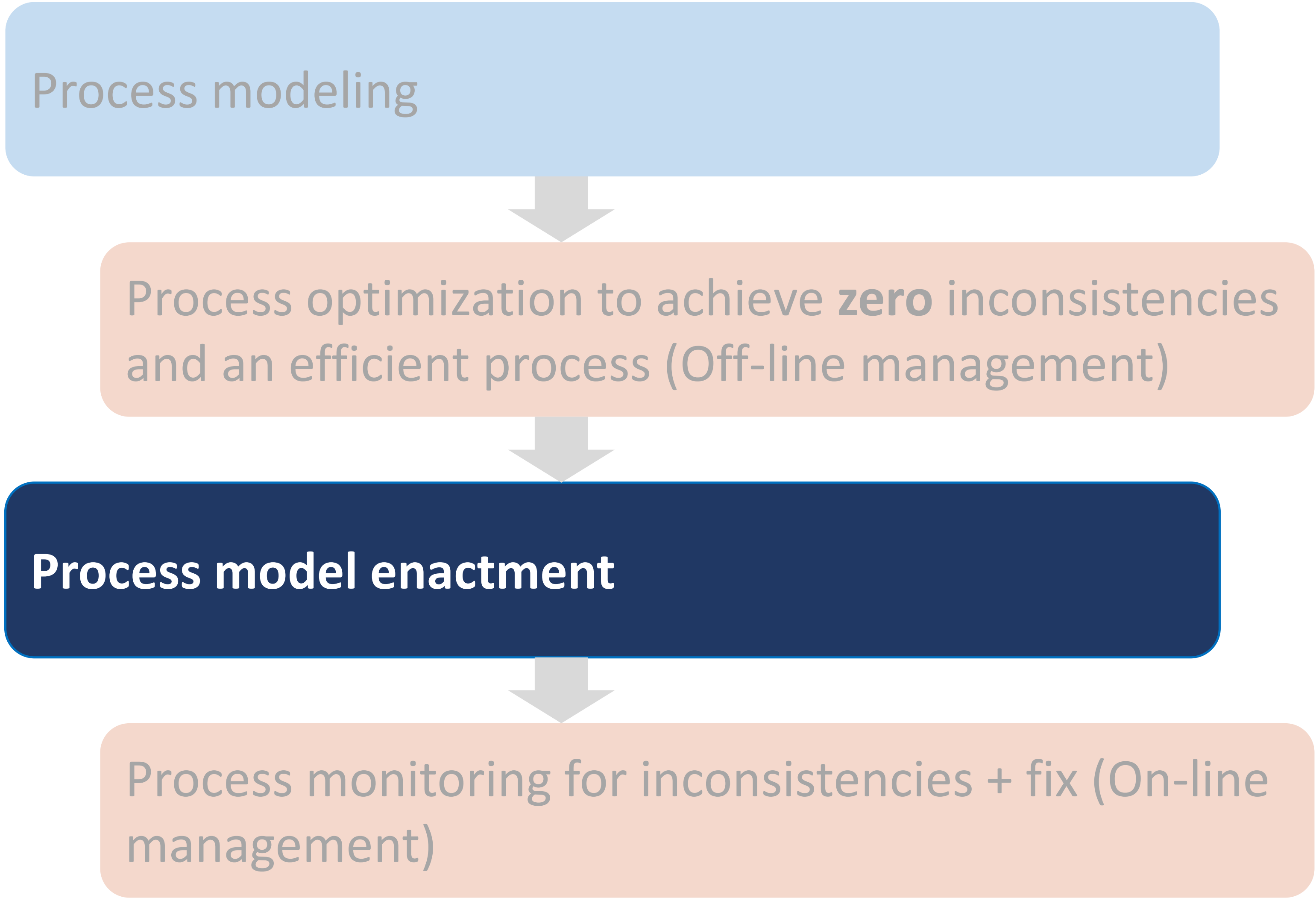
Process modeling

Process optimization to achieve **zero** inconsistencies and an efficient process (Off-line management)

Process model enactment

Process monitoring for inconsistencies + fix (On-line management)

Approach



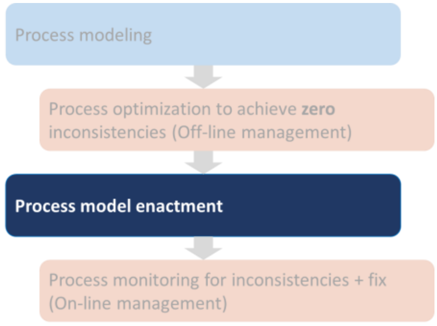
Process modeling

Process optimization to achieve **zero** inconsistencies and an efficient process (Off-line management)

Process model enactment

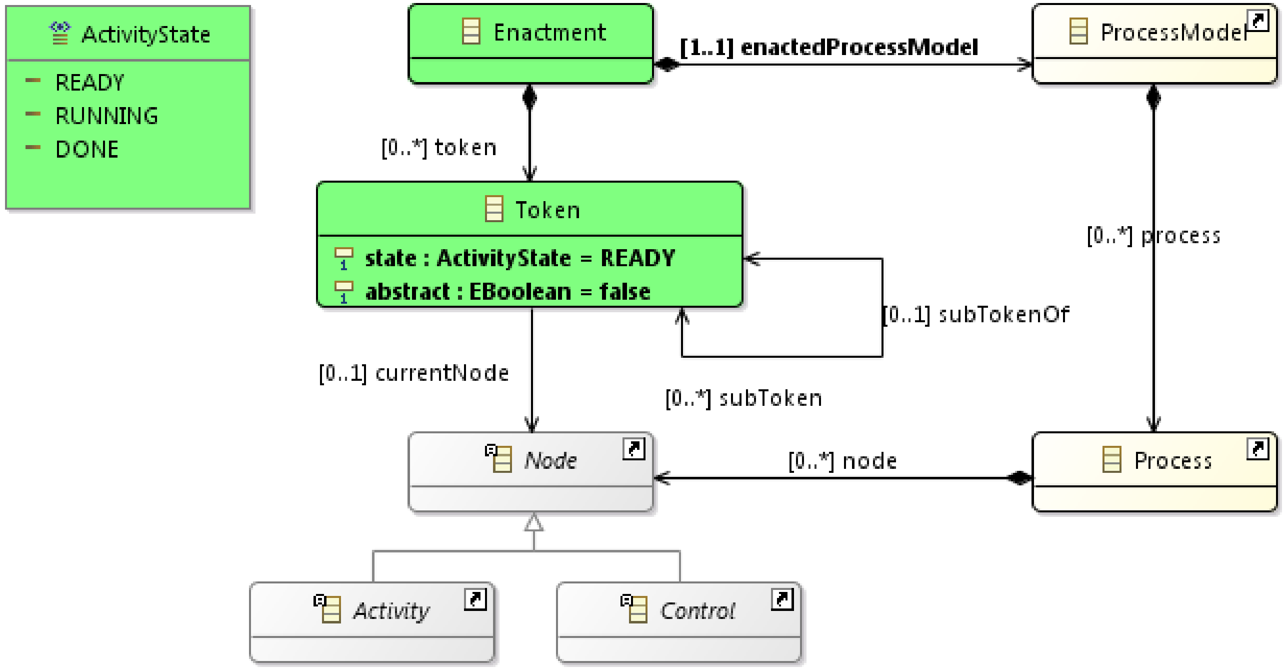
Process monitoring for inconsistencies + fix (On-line management)

Enactment of the FTG+PM



Process modeling
Process optimization to achieve **zero** inconsistencies (Off-line management)
Process model enactment
Process monitoring for inconsistencies + fix (On-line management)

- Execution semantics (based on enactment of model transformations) published by J. Denil
- Eclipse-based engine
  - Explicitly modeled internals
  - External service interoperability



Lúcio, Levi, Sadaf Mustafiz, Joachim Denil, Hans Vangheluwe, and Maris Jukss: „FTG+ PM: an integrated framework for investigating model transformation chains”. In *International SDL Forum*, pp. 182-202. Springer, Berlin, Heidelberg, 2013.

J. Denil: „*Design, Verification and Deployment of Software Intensive Systems – A Multi-Paradigm Modelling Approach*”. PhD Thesis, 2013.

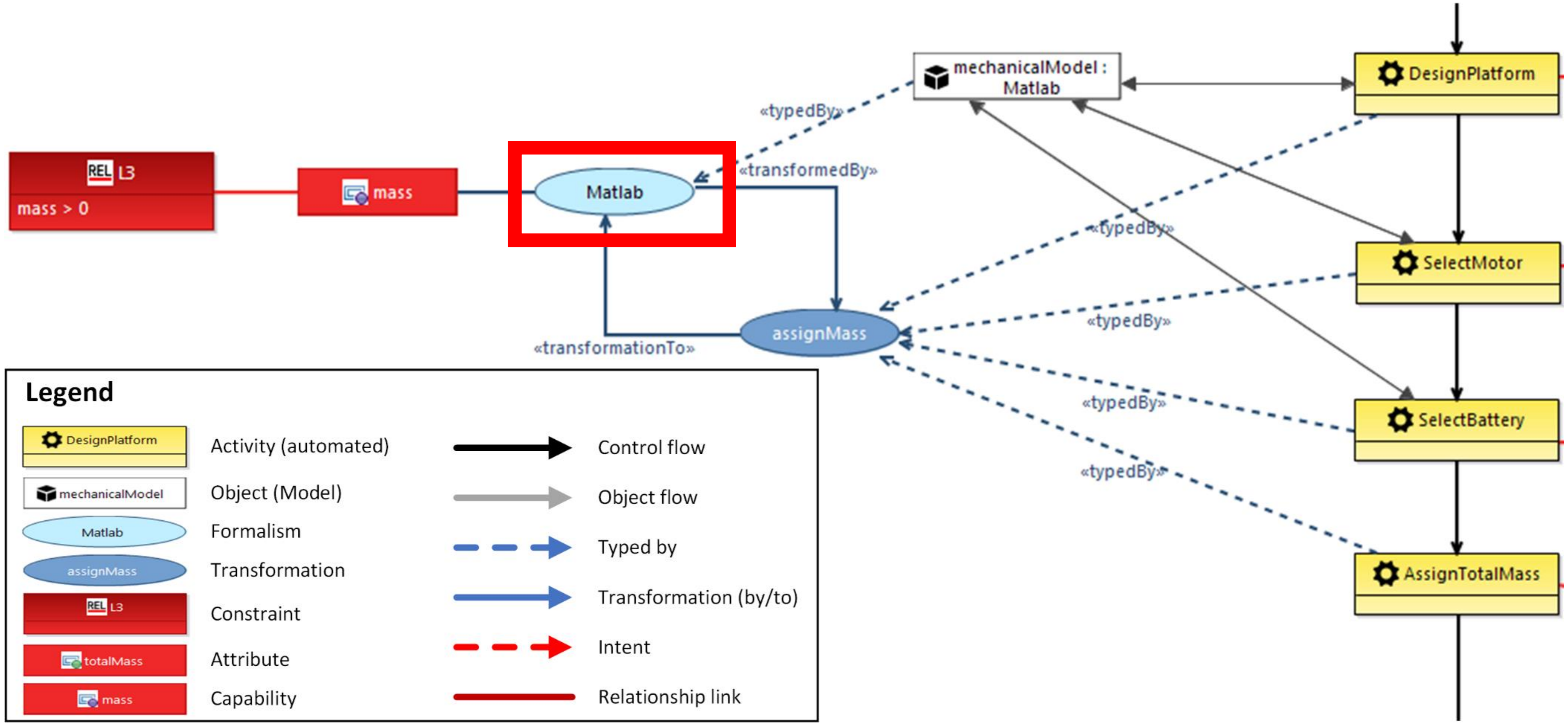
Process modeling

Process optimization to achieve **zero** inconsistencies (Off-line management)

Process model enactment

Process monitoring for inconsistencies + fix (On-line management)

Enactment of the FTG+PM



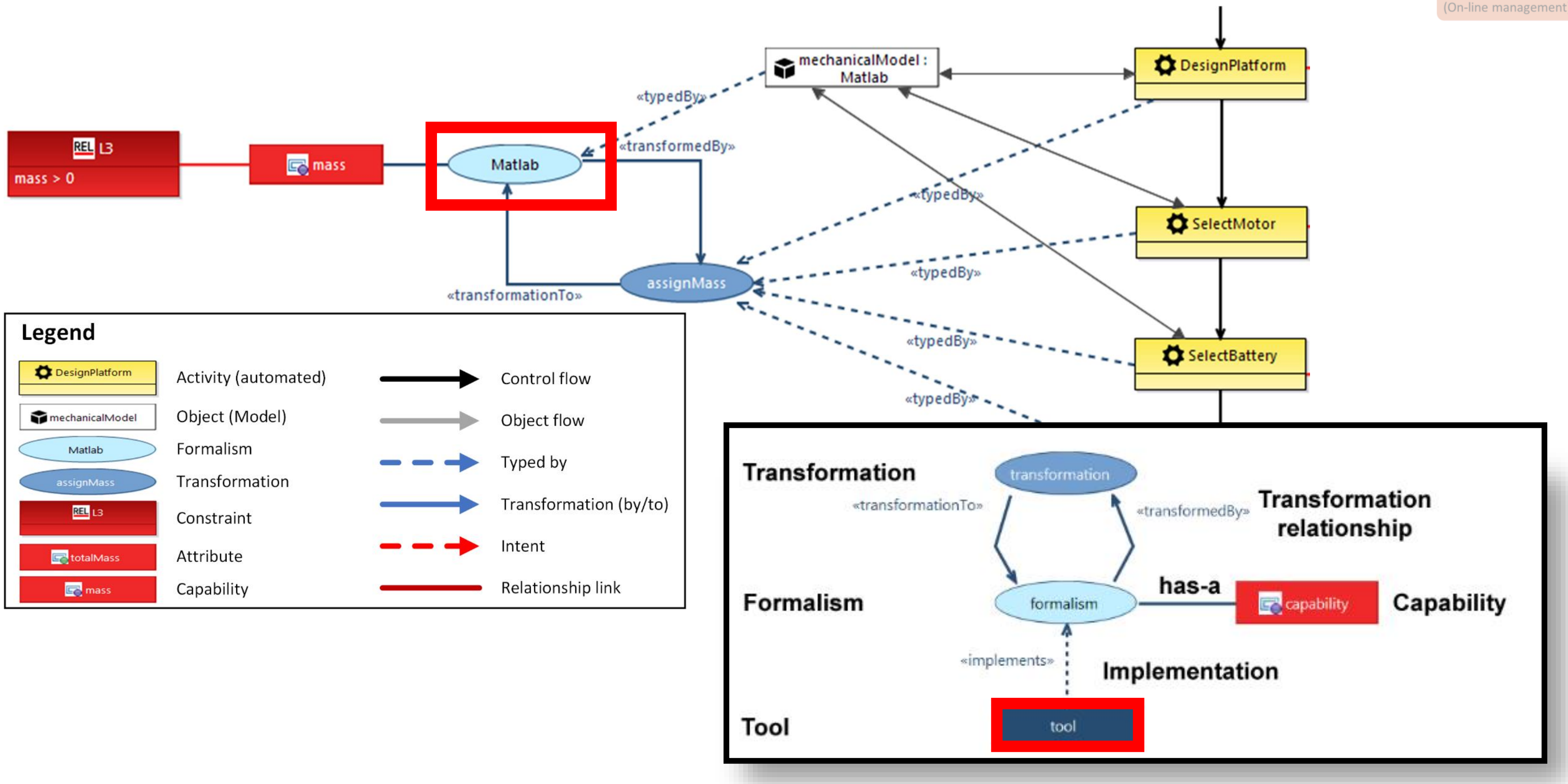
Process modeling

Process optimization to achieve **zero** inconsistencies (Off-line management)

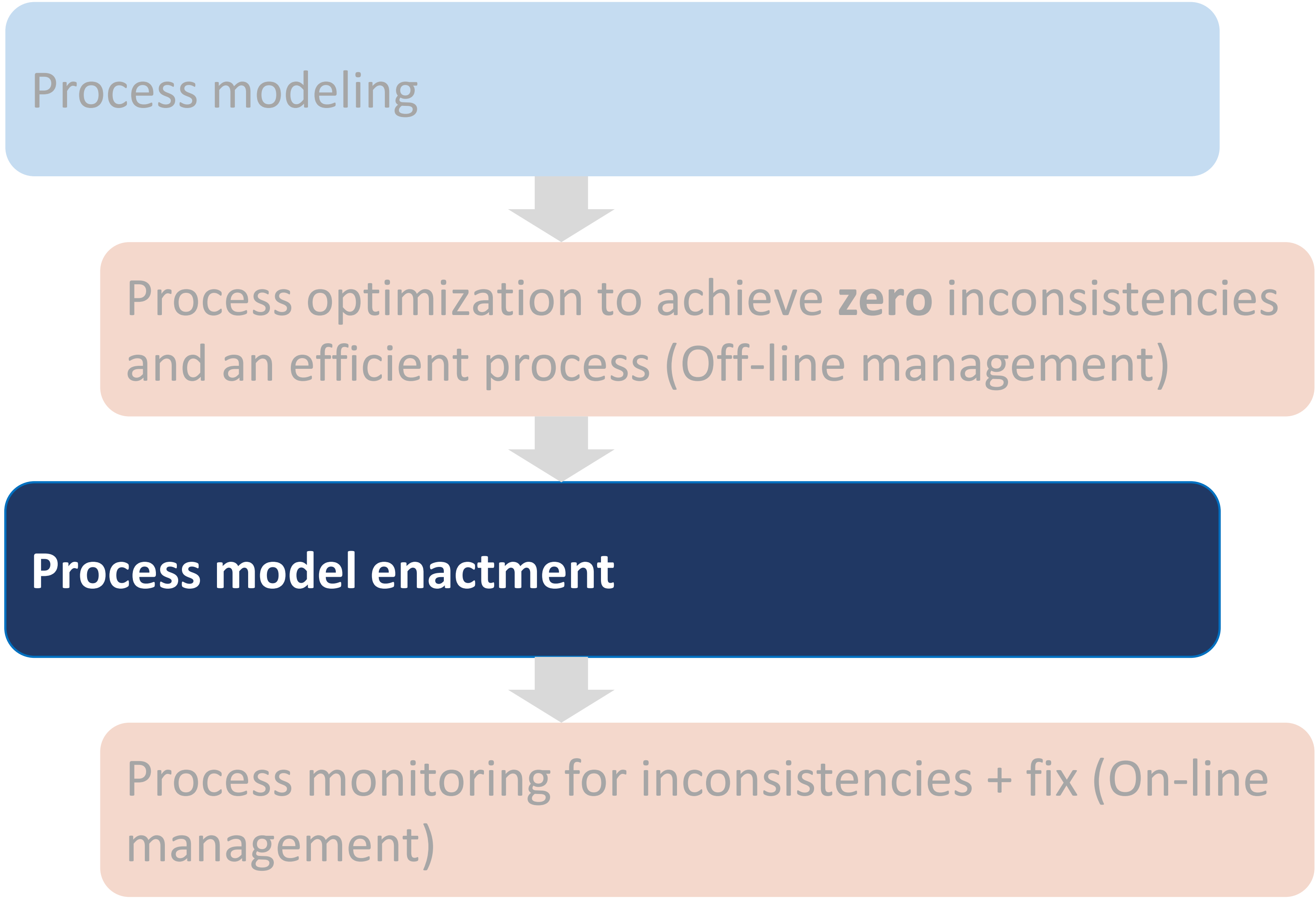
Process model enactment

Process monitoring for inconsistencies + fix (On-line management)

Enactment of the FTG+PM



Approach



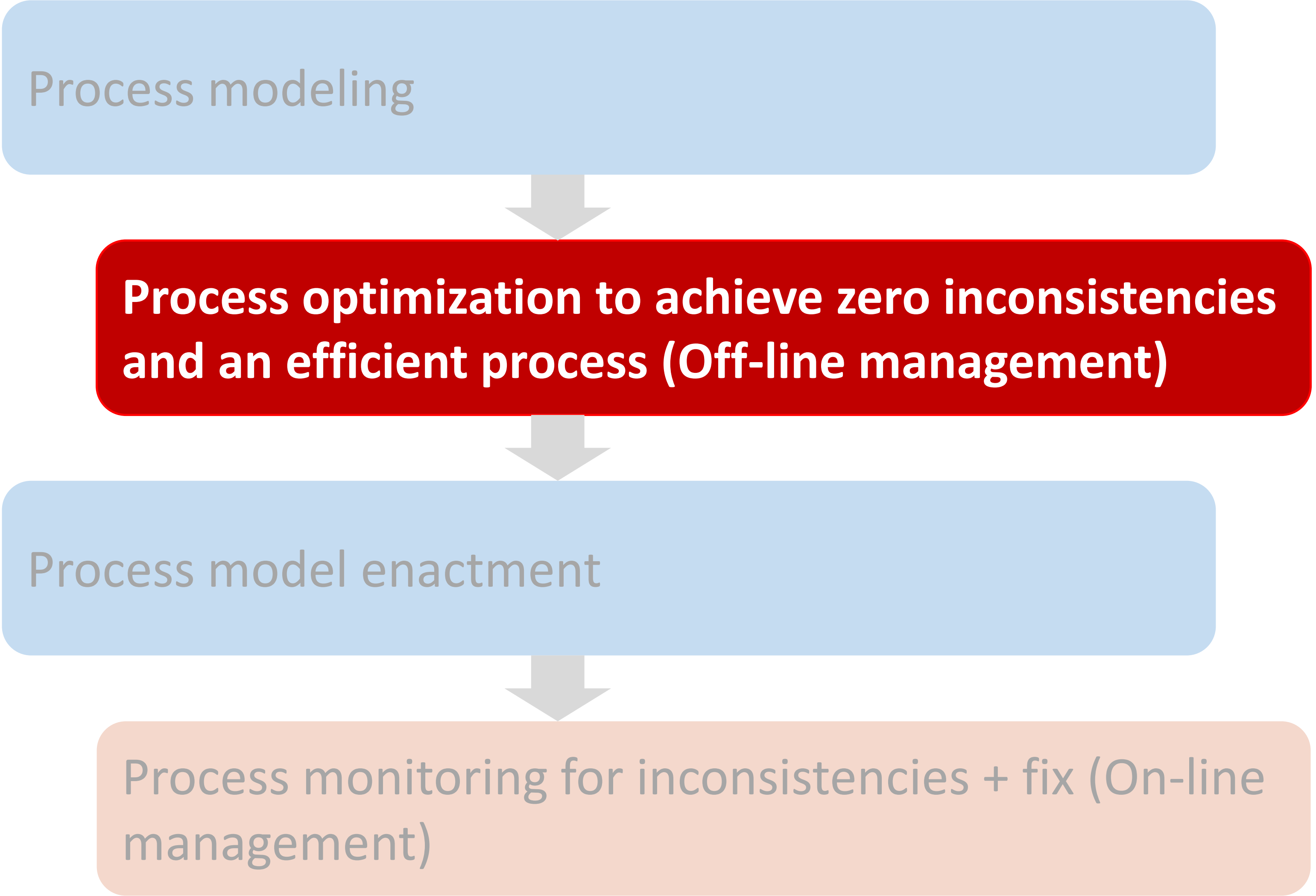
Process modeling

Process optimization to achieve **zero** inconsistencies and an efficient process (Off-line management)

Process model enactment

Process monitoring for inconsistencies + fix (On-line management)

Approach



Process modeling

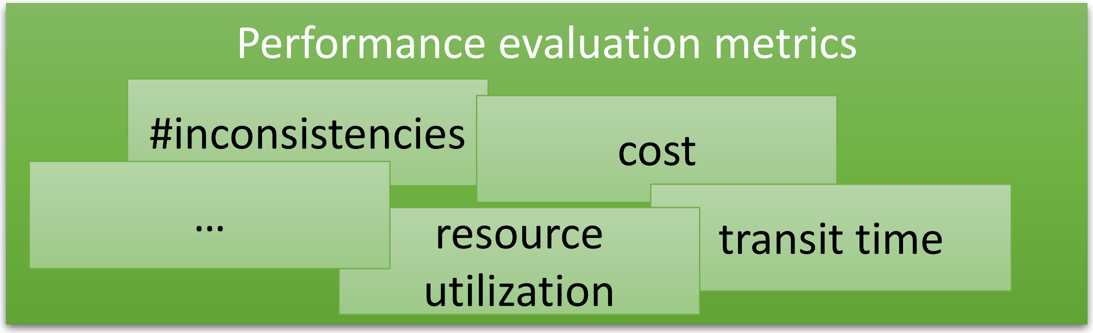
Process optimization to achieve zero inconsistencies and an efficient process (Off-line management)

Process model enactment

Process monitoring for inconsistencies + fix (On-line management)

Evaluating processes

- Quantitative (performance) metrics are required



Performance evaluation metrics

#inconsistencies

cost

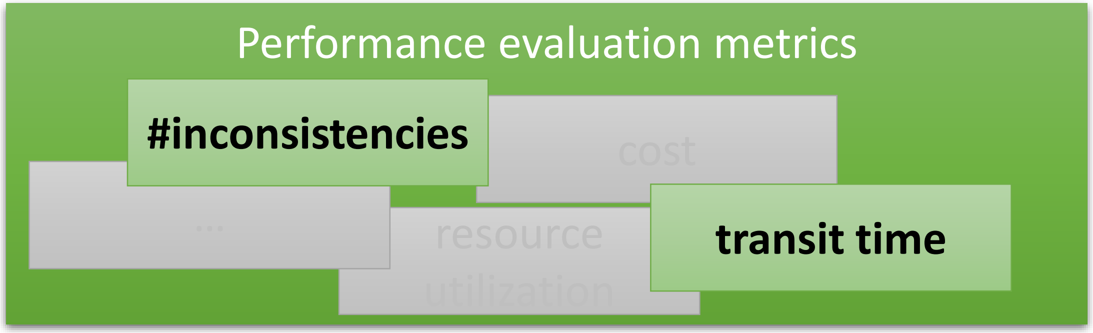
…

resource utilization

transit time

Evaluating processes

- Quantitative (performance) metrics are required



Performance evaluation metrics

#inconsistencies

cost

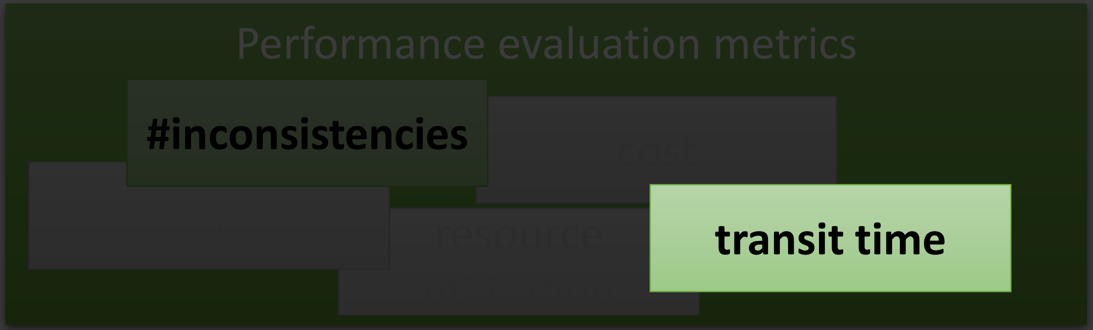
…

resource utilization

transit time

Evaluating processes

- Quantitative (performance) metrics are required



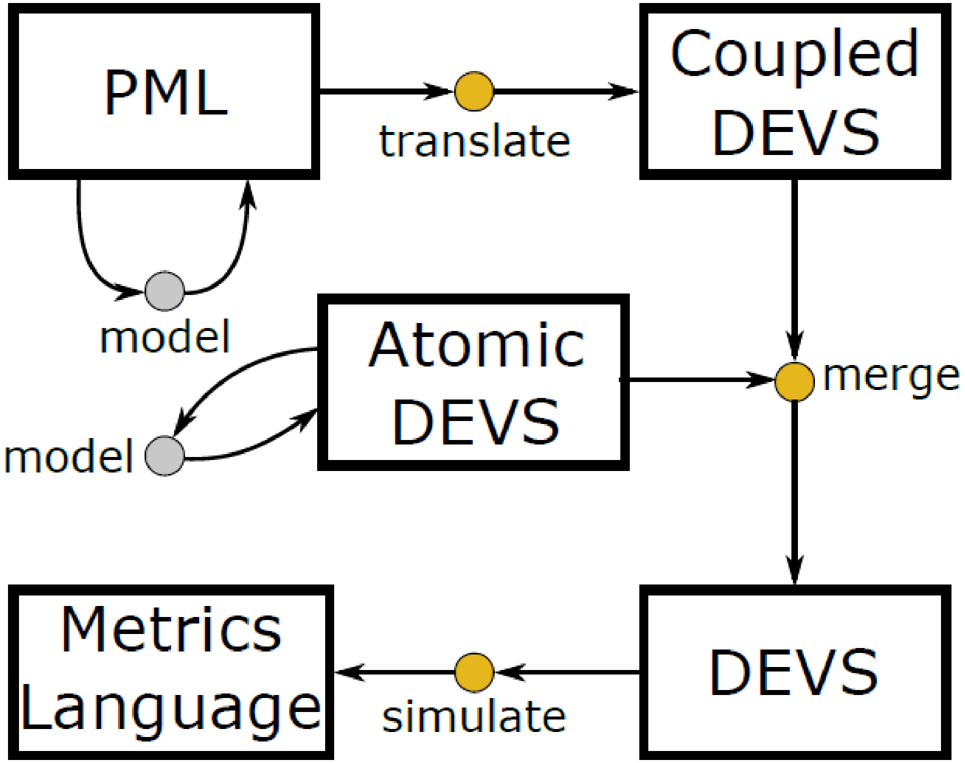
Process modeling

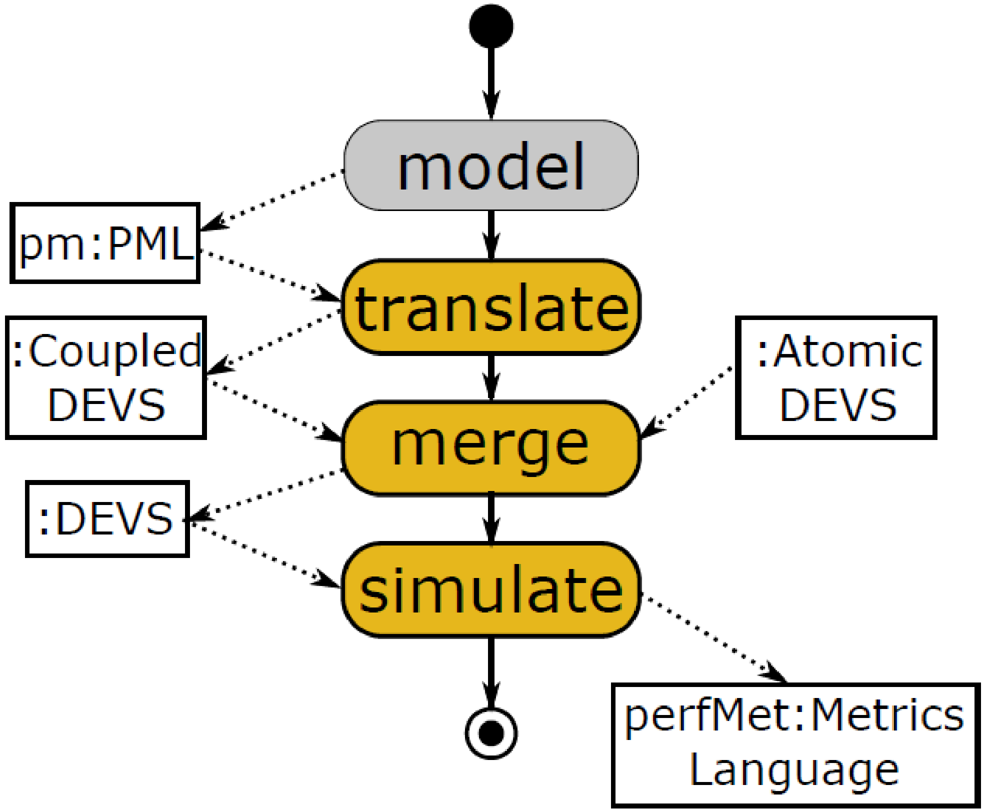
Process optimization to achieve zero inconsistencies (Off-line management)

Process model enactment

Process monitoring for inconsistencies + fix (On-line management)

Stochastic simulations of transit time

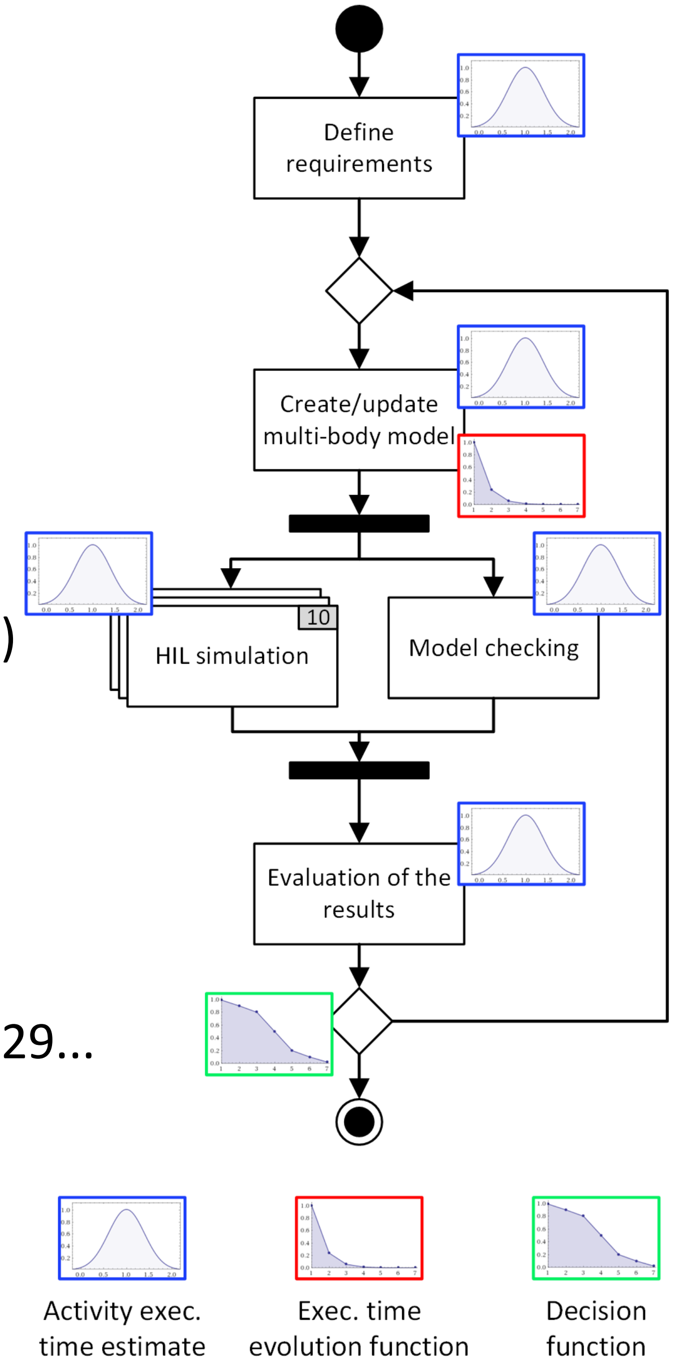


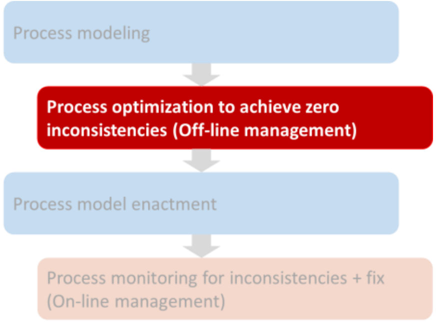


I. Dávid, Y. Van Tendeloo, H. Vangheluwe: „*Translating Engineering Workflow Models to DEVS for Performance Evaluation*”. In Proceedings of the 2018 Winter Simulation Conference, pp. 616-627. IEEE Press, 2018.

Calibration

- Activity execution time
  - Rule
    - Gaussian distribution
    - 80% of the estimations within the 20% error range: t(a)=N(μ, 0.15625μ)
- Execution time evolution
  - Rule
    - $e^{-1/0.7i}$ (i: iteration)
  - Resulting values
    - 1.0, 0.2397, 0.05743, 0.01376, 0.00329…
- Decision function
  - Manually set
  - 0.99, 0.9, 0.8, 0.5, 0.2, 0.1


Define requirements
Create/update multi-body model
HIL simulation
10
Model checking
Evaluation of the results
Activity exec. time estimate
Exec. time evolution function
Decision function


Process modeling

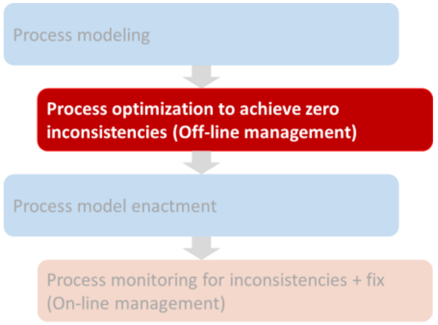
Process optimization to achieve zero inconsistencies (Off-line management)

Process model enactment

Process monitoring for inconsistencies + fix (On-line management)

A catalogue of workflow patterns

- Wil van der Aalst (RWTH Aachen University)
- http://www.workflowpatterns.com/
- Patterns
  - Control (2003, rev. 2006)
  - Resource (2004)
  - Data (2004)
  - Exception handling (2006)
  - Presentation (2011)
  - Log imperfection (added in 2017)


Process modeling

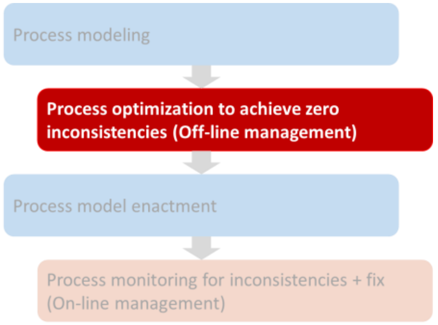
Process optimization to achieve zero inconsistencies (Off-line management)

Process model enactment

Process monitoring for inconsistencies + fix (On-line management)

A catalogue of workflow patterns

- Wil van der Aalst (RWTH Aachen University)
- http://www.workflowpatterns.com/
- Patterns
  - **Control (2003, rev. 2006)** ⬅ Our scope
  - Resource (2004)
  - Data (2004)
  - Exception handling (2006)
  - Presentation (2011)
  - Log imperfection (added in 2017)



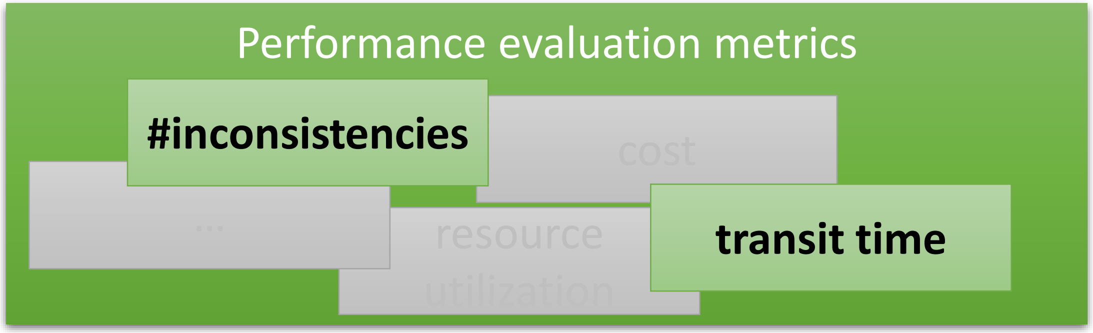
Process modeling

Process optimization to achieve zero inconsistencies (Off-line management)

Process model enactment

Process monitoring for inconsistencies + fix (On-line management)

Process space exploration



Performance evaluation metrics

#inconsistencies

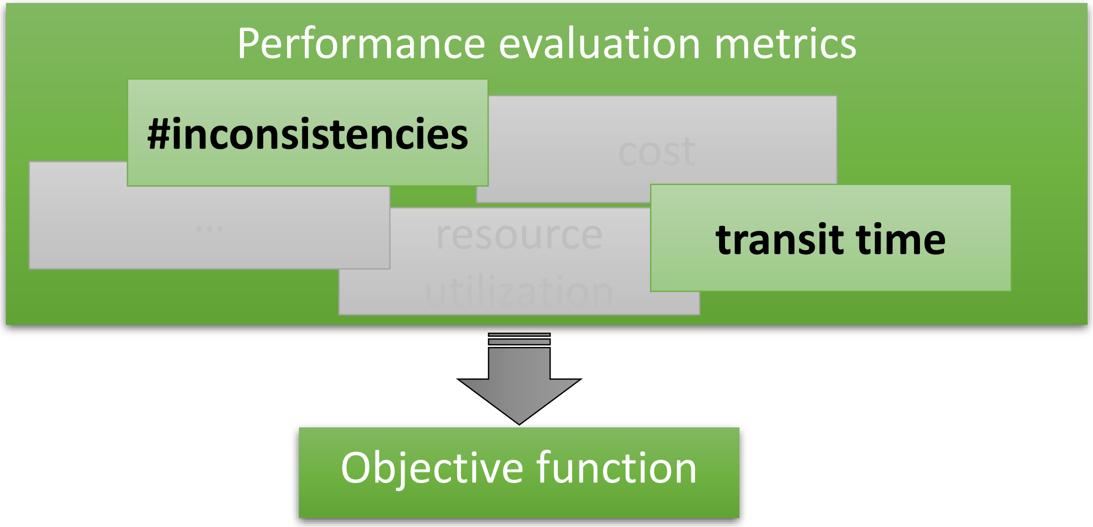
cost

...

resource utilization

transit time

Process space exploration



Performance evaluation metrics

#inconsistencies

cost

…

resource utilization

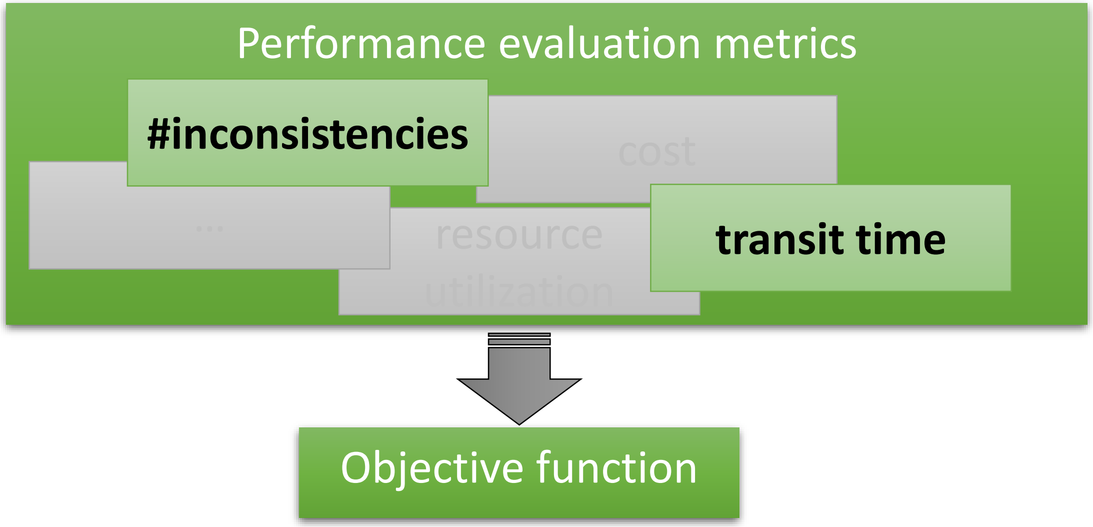
transit time

Objective function

Process space exploration

R_p Process rewrite rules

- Inconsistency management rules
- Performance improvement rules



Performance evaluation metrics

#inconsistencies

cost

…

resource utilization

transit time

Objective function

Process space exploration

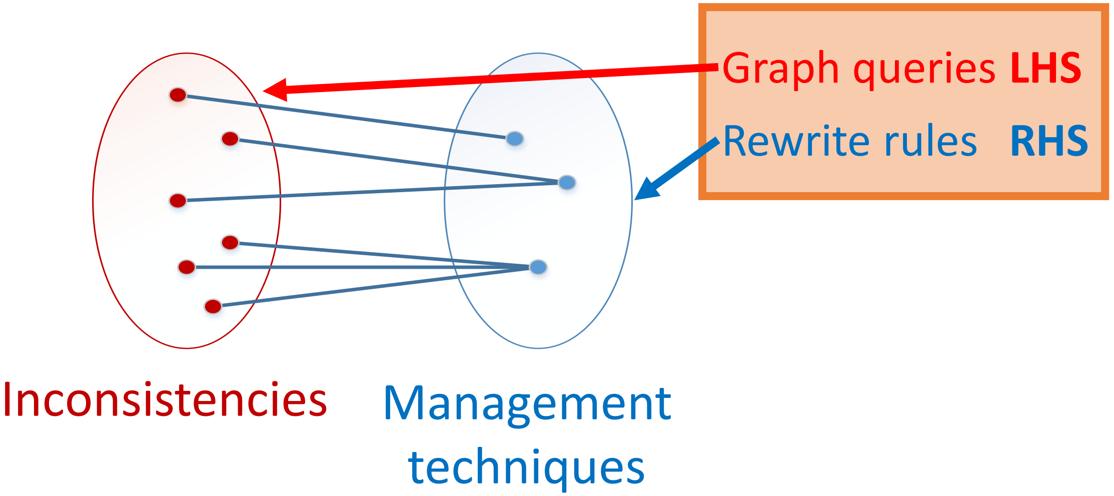
R_p Process rewrite rules

- Inconsistency management rules
- Performance improvement rules

Performance evaluation metrics

- **#inconsistencies**
- cost
- …
- resource utilization
- **transit time**

Objective function

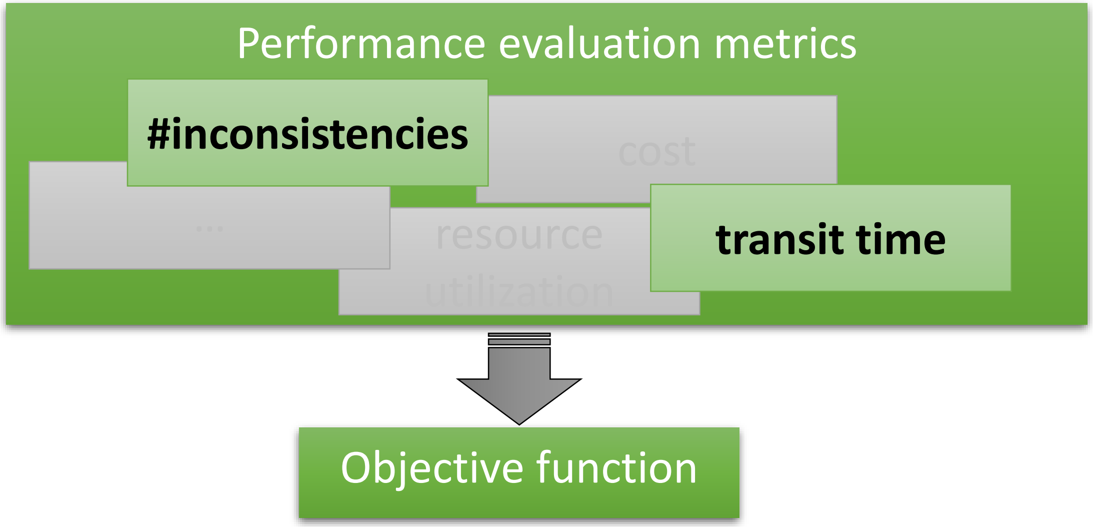


Applying an inconsistency management rule
==
executing a model transformation

Process space exploration

R_p Process rewrite rules

- Inconsistency management rules
- Performance improvement rules



Performance evaluation metrics

#inconsistencies

cost

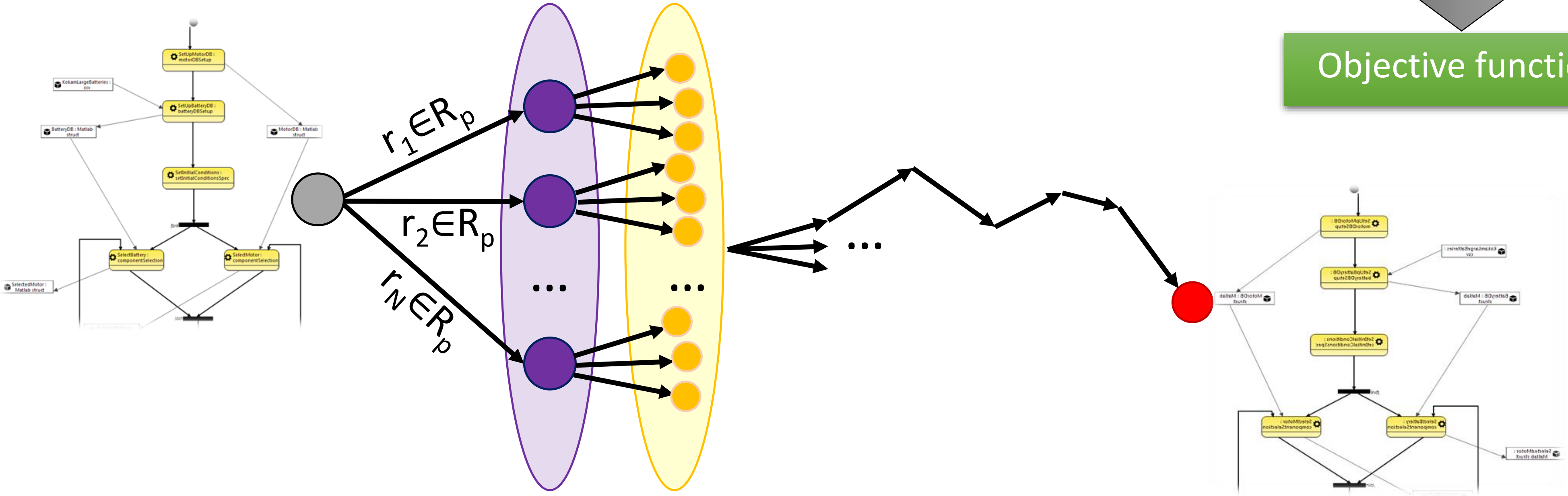
...

resource utilization

transit time

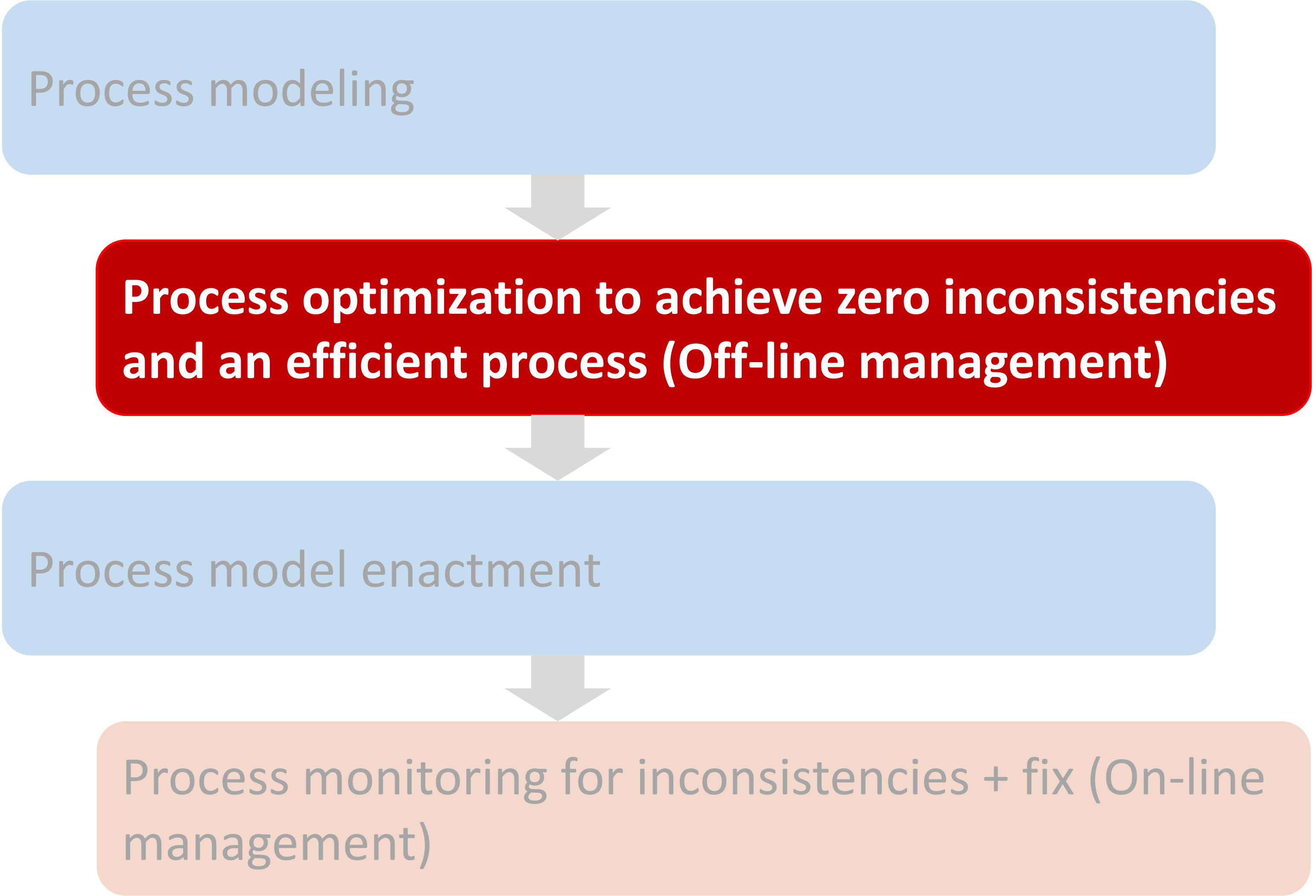
Objective function

$$P(1) = \cup_{\forall r \in R_p} r(P_0)$$



$$P(2) = \cup_{\forall r \in R_p,\, p \in P(1)} r(p)$$

Approach



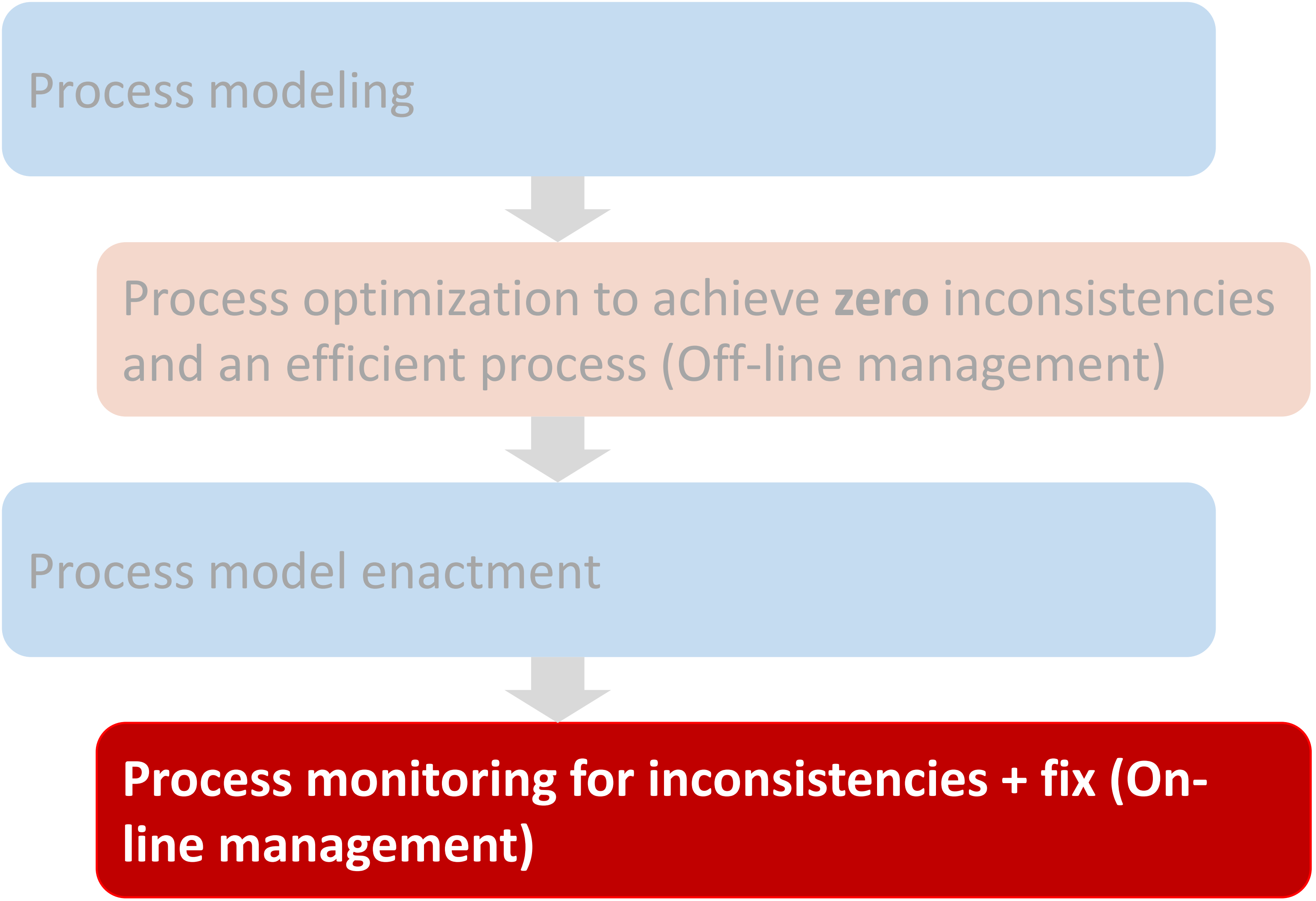
Process modeling

Process optimization to achieve zero inconsistencies and an efficient process (Off-line management)

Process model enactment

Process monitoring for inconsistencies + fix (On-line management)

Approach



Process modeling

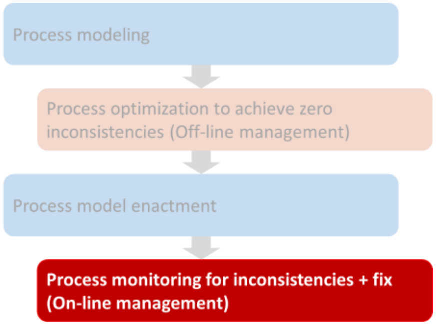
Process optimization to achieve **zero** inconsistencies and an efficient process (Off-line management)

Process model enactment

Process monitoring for inconsistencies + fix (On-line management)

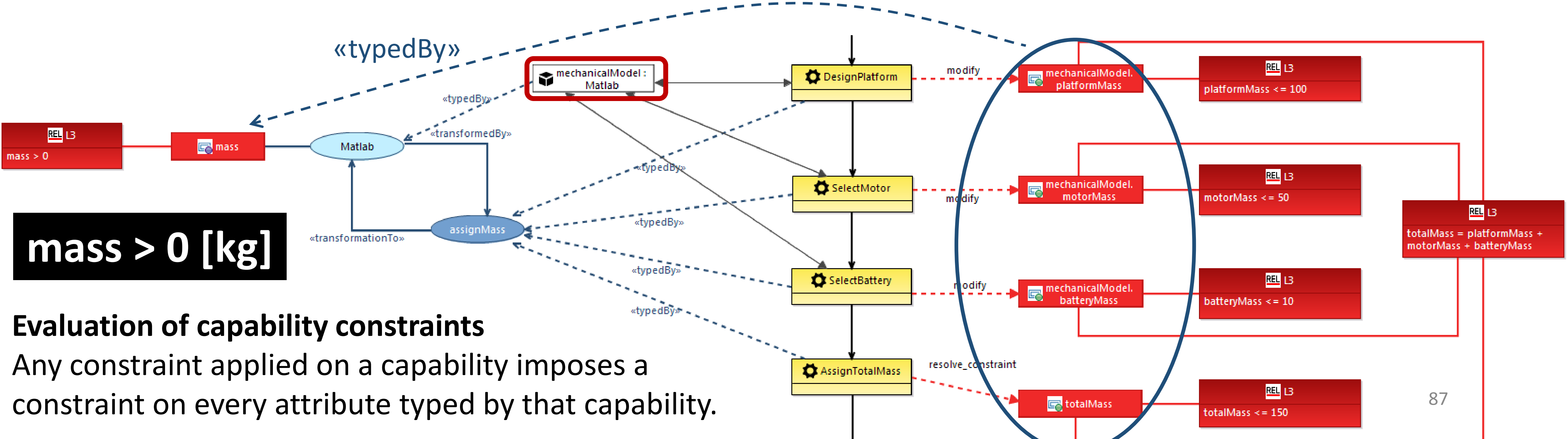
On-line management

SymPy



Process modeling
Process optimization to achieve zero inconsistencies (Off-line management)
Process model enactment
Process monitoring for inconsistencies + fix (On-line management)

- Continuous maintenance of constraints
  - Symbolic mathematical computation
- Forward/backward propagation of solution sets

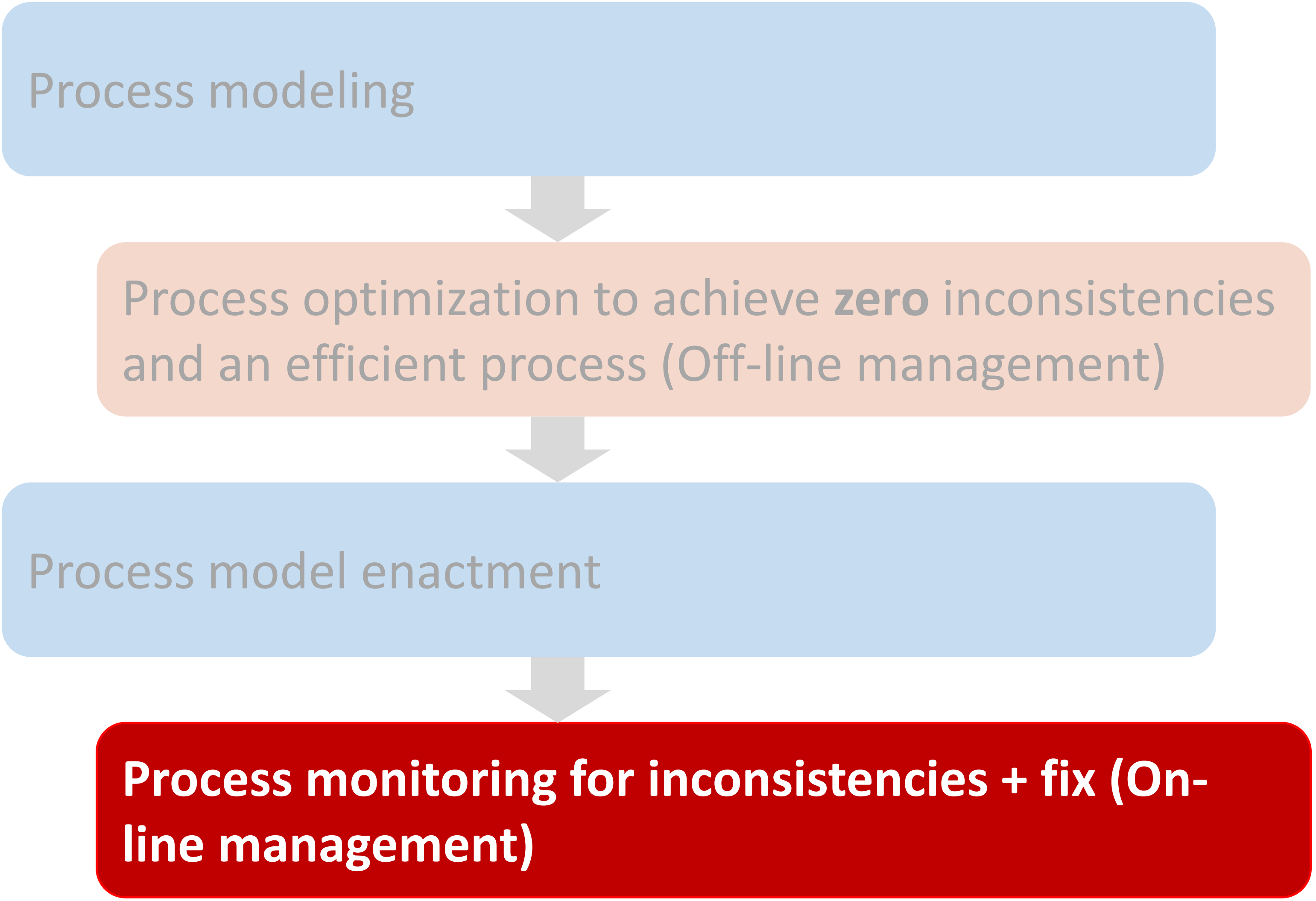


mass > 0 [kg]

Evaluation of capability constraints

Any constraint applied on a capability imposes a constraint on every attribute typed by that capability.

Approach



Process modeling

Process optimization to achieve **zero** inconsistencies and an efficient process (Off-line management)

Process model enactment

Process monitoring for inconsistencies + fix (On-line management)

Inconsistency Management

Theory/Foundations | Formalism | Additional enablers

Contributions

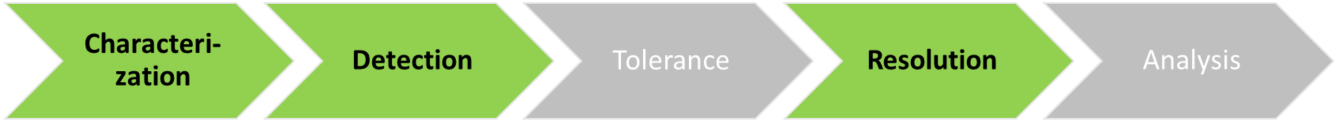
- Contribution 1: A **mapping study of the state of the art**, in order to identify the shortcomings of the currently available inconsistency management techniques.
- Contribution 2: A revised **definition of model (in)consistency**, in order to support the management of semantic inconsistencies.
- Contribution 3: A **process modeling formalism**, serving as the foundation for the rest of the contributions.
- Contribution 4: **Off-line management of inconsistencies** by the means of rule-based multi-objective process-space exploration.
- Contribution 5: **Enactment of the optimized process** by the means of explicit model transformations.
- Contribution 6: **On-line management of inconsistencies** by monitoring the process under enactment.
- Contribution 7: **DEVS-based simulation of process models** for evaluating the performance of the process candidates in the off-line management phase.
- Contribution 8: **External service integration** in processes under enactment, by explicit transformations over SCCD models.

Publications

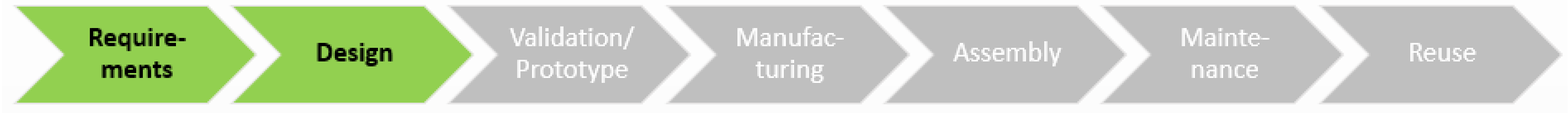
- **Translating Engineering Workflow Models to DEVS for Performance Evaluation**, I. Dávid, Y. Van Tendeloo, H. Vangheluwe. In Proceedings of the 2018 Winter Simulation Conference, pp. 616-627. IEEE Press, 2018.
- **Process-oriented Inconsistency Management in Collaborative Systems Modeling**, I. Dávid, J. Denil, H. Vangheluwe, 16th annual Industrial Simulation Conference, 2018.
- **A Multi-Paradigm Approach for Modelling Service Interactions in Model-Driven Engineering Processes**, S. Van Mierlo, Y. Van Tendeloo, I. Dávid, B. Meyers, A. Gebremichael, H. Vangheluwe, Model-driven Approaches for Simulation Engineering Symposium (MOD4SIM) -- Spring Simulation Multi-Conference, 2018.
- **Modeling and Enactment Support for Early Detection of Inconsistencies in Engineering Processes**, I. Dávid, B. Meyers, K. Vanherpen, Y. Van Tendeloo, K. Berx, H. Vangheluwe, 2nd International Workshop on Collaborative Modelling in MDE, 2017.
- **Towards Inconsistency Tolerance by Quantification of Semantic Inconsistencies**, I. Dávid, E. Syriani, C. Verbrugge, D. Buchs, D. Blouin, A. Cicchetti, K. Vanherpen, 1st International Workshop on Collaborative Modelling in MDE, 2016.
- **Engineering Process Transformation to Manage (In)consistency**, I. Dávid, J. Denil, K. Gadeyne, H. Vangheluwe, 1st International Workshop on Collaborative Modelling in MDE, 2016.
- **A Multi-Paradigm Modeling Foundation for Collaborative Multi-view Model/System Development**, I. Dávid, ACM Student Research Competition (SRC) MoDELS 2016.
- **Ontological Reasoning for Consistency in the Design of Cyber-Physical Systems**, K. Vanherpen, J. Denil, I. Dávid, P. De Meulenaere, P. J. Mosterman, M. Törngren, A. Qamar, H. Vangheluwe. CPPS 2016 - 1st International Workshop on Cyber-Physical Production Systems, 2016.
- **Patterns of inconsistency management in mechatronics - A survey**, I. Dávid, J. Denil, H. Vangheluwe. Technical report, 2015.
- **Towards Inconsistency Management by Process-Oriented Dependency Modeling**, I. Dávid, J. Denil, H. Vangheluwe. 9th International Workshop on Multi-Paradigm Modeling, 2015.

Potential future directions

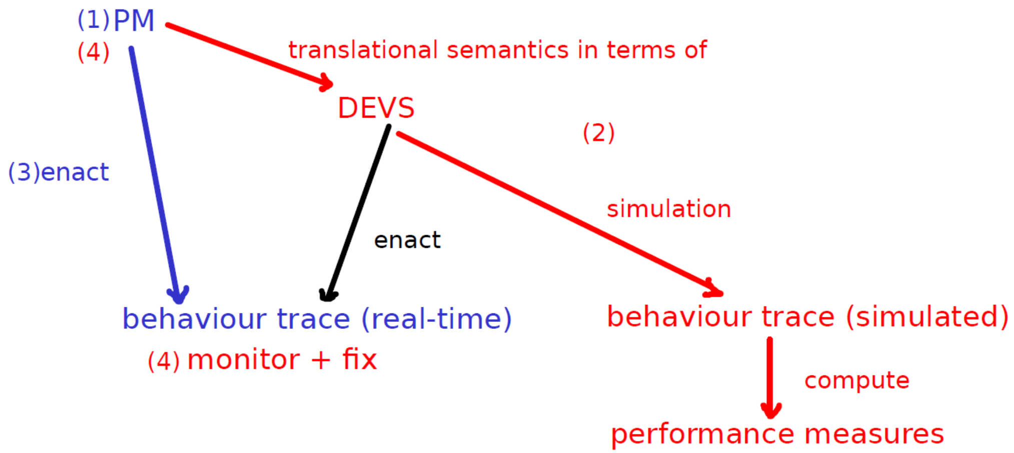
- Extending the scope to Tolerance and Analysis


Characteri-zation
Detection
Tolerance
Resolution
Analysis

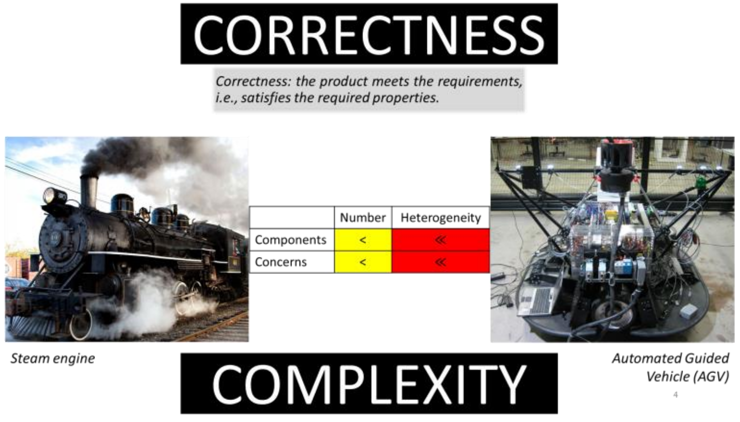
- Explicit reasoning about the trade-off between Prevention and Detect-and-Resolve
- Extending the scope of the technique over the entire engineering process


Require-ments
Design
Validation/ Prototype
Manufac-turing
Assembly
Mainte-nance
Reuse

- Enactment by DEVS
- Partial process inference


(1)PM
(4)
translational semantics in terms of
DEVS
(2)
(3)enact
simulation
enact
behaviour trace (real-time)
(4) monitor + fix
behaviour trace (simulated)
compute
performance measures

Conclusions



CORRECTNESS

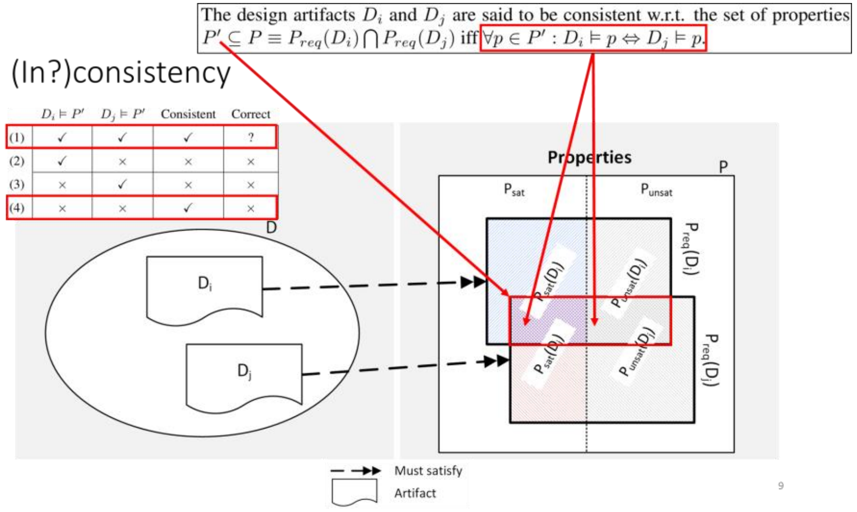
Correctness: the product meets the requirements, i.e., satisfies the required properties.

	Number	Heterogeneity
Components	<	≪
Concerns	<	≪

Steam engine

Automated Guided Vehicle (AGV)

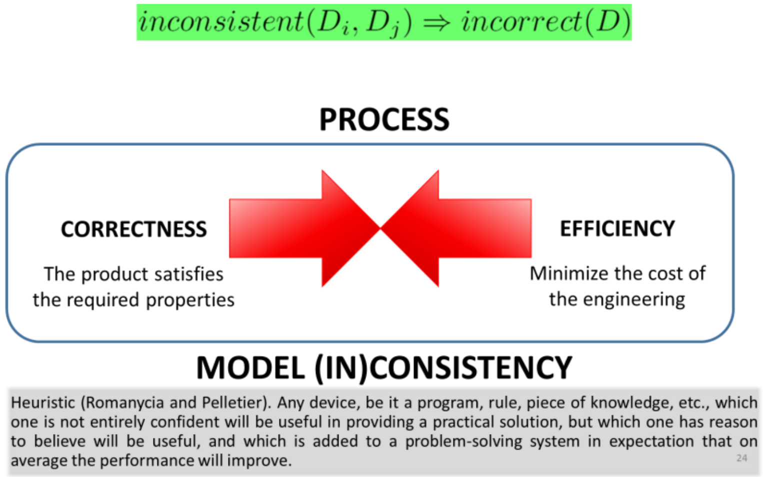
COMPLEXITY



The design artifacts D_i and D_j are said to be consistent w.r.t. the set of properties $P' \subseteq P \equiv P_{req}(D_i) \bigcap P_{req}(D_j)$ iff $\forall p \in P' : D_i \models p \Leftrightarrow D_j \models p$.

(In?)consistency

	$D_i \models P'$	$D_j \models P'$	Consistent	Correct
(1)	✓	✓	✓	?
(2)	✓	×	×	×
(3)	×	✓	×	×
(4)	×	×	✓	×



$inconsistent(D_i, D_j) \Rightarrow incorrect(D)$

PROCESS

CORRECTNESS

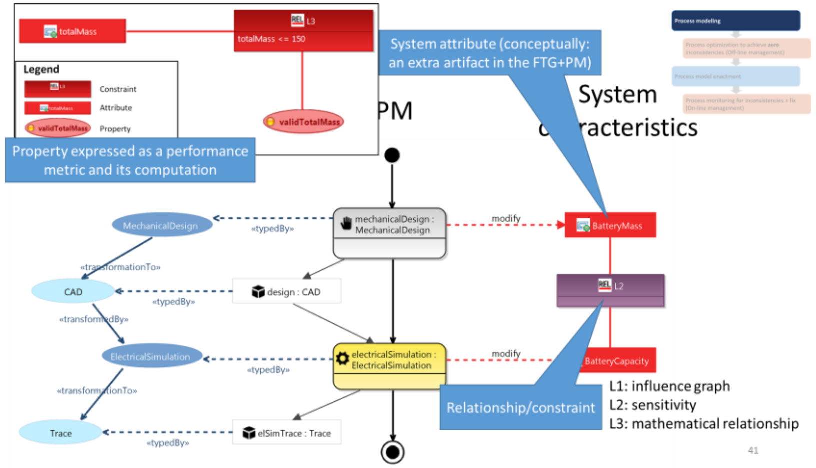
The product satisfies the required properties

EFFICIENCY

Minimize the cost of the engineering

MODEL (IN)CONSISTENCY

Heuristic (Romanycia and Pelletier). Any device, be it a program, rule, piece of knowledge, etc., which one is not entirely confident will be useful in providing a practical solution, but which one has reason to believe will be useful, and which is added to a problem-solving system in expectation that on average the performance will improve.

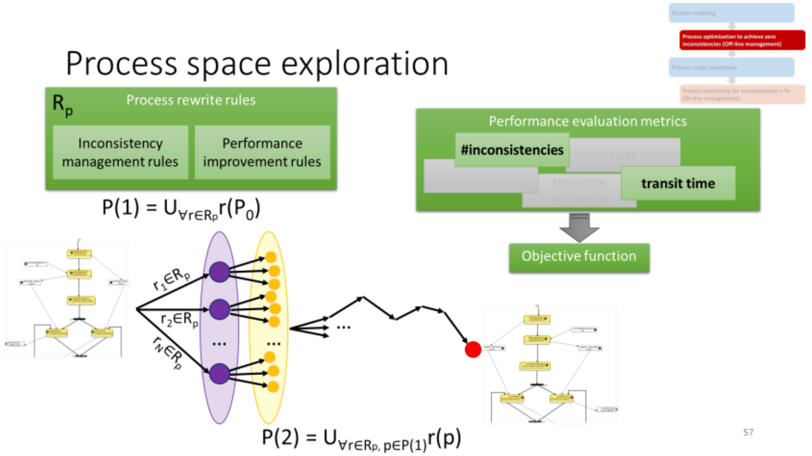


System attribute (conceptually: an extra artifact in the FTG+PM)

Property expressed as a performance metric and its computation

Relationship/constraint

L1: influence graph
L2: sensitivity
L3: mathematical relationship

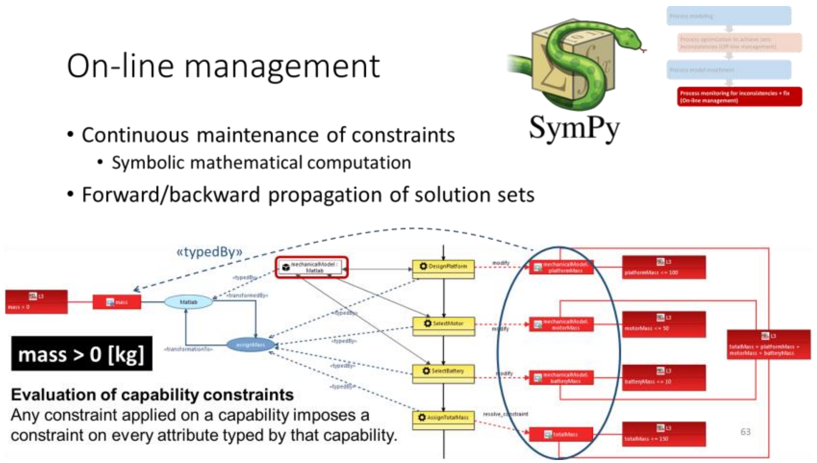


Process space exploration

R_p
Process rewrite rules
Inconsistency management rules
Performance improvement rules
Performance evaluation metrics
#inconsistencies
cost
resource
transit time
Objective function
$r_1 \in R_p$
$r_2 \in R_p$
$r_N \in R_p$
...
Process modeling
Process optimization to achieve zero inconsistencies (Off-line management)
Process model enactment
Process monitoring for inconsistencies + fix (On-line management)
57

$P(1) = \mathrm{U}_{\forall r \in R_p} r(P_0)$

$P(2) = \mathrm{U}_{\forall r \in R_p,\, p \in P(1)} r(p)$



On-line management

- Continuous maintenance of constraints
  - Symbolic mathematical computation
- Forward/backward propagation of solution sets

SymPy

mass > 0 [kg]

Evaluation of capability constraints
Any constraint applied on a capability imposes a constraint on every attribute typed by that capability.

Thank you!