On Interoperability Challenges and Scenarios

Morad Behandish[1]; Vadim Shapiro[1,2]

[1]International Computer Science Institute (ICSI), [2]University of Wisconsin-Madison

Abstract

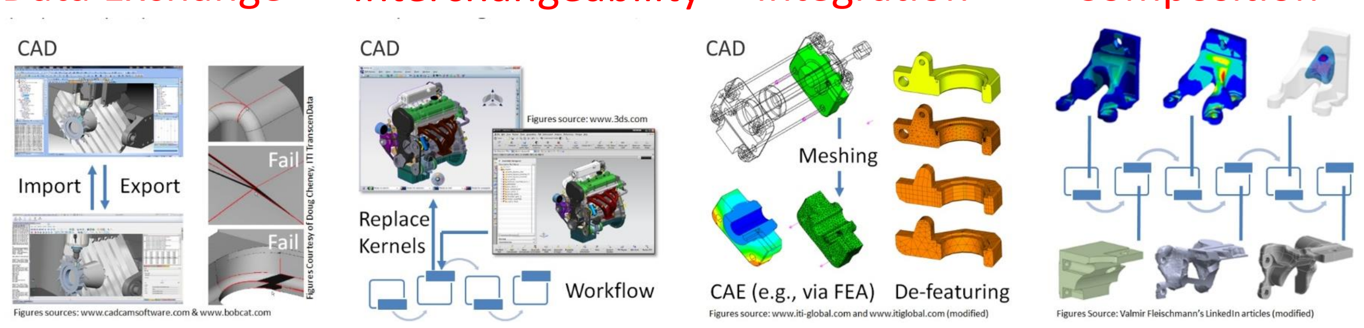
A common informal definition of **interoperability** refers to the ability of a system to communicate and work with other products or systems, present or future, without any restricted access or implementation [1]. It subsumes the problems of data sharing, exchange, and translation, as well as systems integration.

Limited or lacking interoperability has emerged as a central unsolved technical problem in research, development, maintenance, scalability, and security of engineering systems, with crippling effects on further advances in conceptual design, simulation, synthesis, optimization, manufacturing planning, and productivity gains. In addition, it has become a major economic problem within the past two decades, costing the US manufacturing industry billions of dollars every year [2]. This research aims to provide a generic framework to enable semantic interoperability of modern information-intensive design engineering systems that are heterogeneous (in scope, domain and functionality), distributed (in space-time), multi-scale/multi-physical, and interactive.

Scope and Impact

Interoperability subsumes important unsolved problems in CAx:

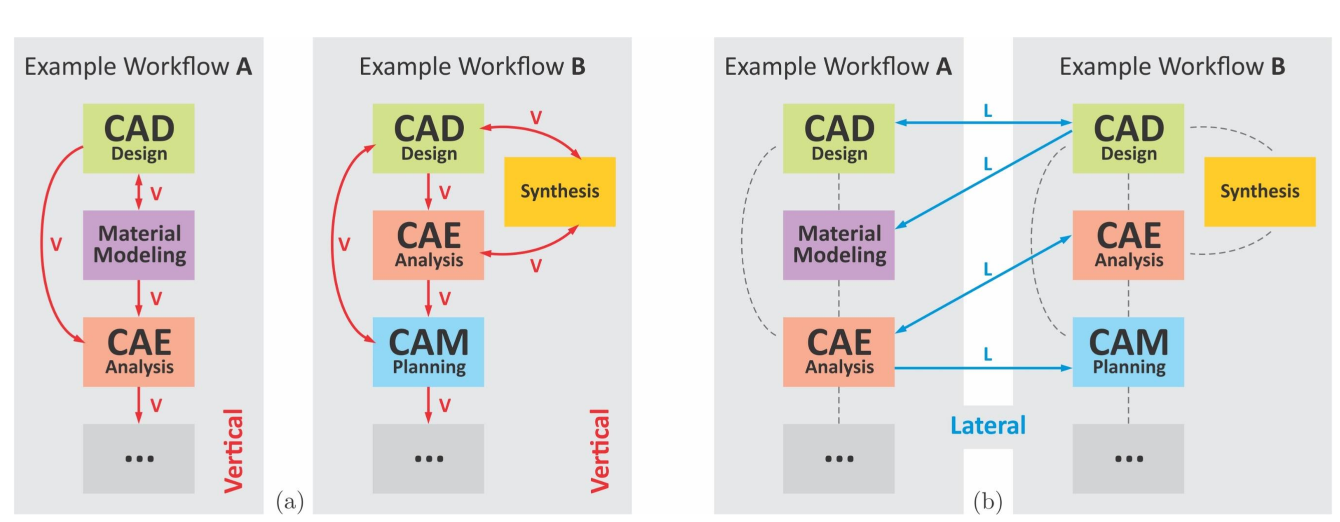
Data Exchange | Interchangeability | Integration | Composition



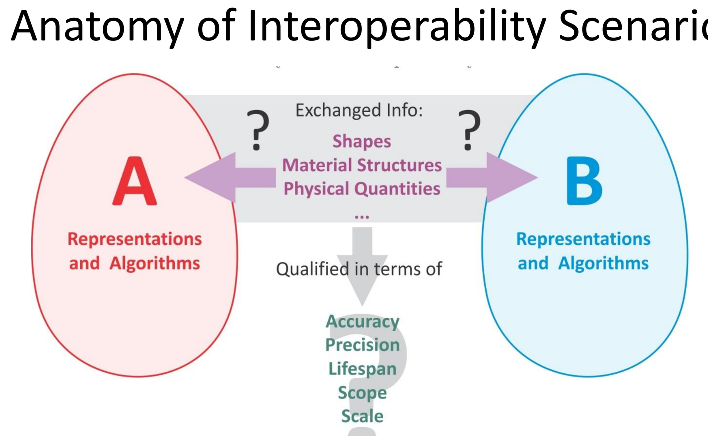
- Data-centric Approaches: fail due to lack of *common semantics.*
  - System-to-Neutral conversion (e.g., STEP [3], IGES [4], etc.);
  - System-to-System conversion.
- Query-based Approaches: are less studied/validated.
  - API-based interaction – extended in [5] and applied to [6, 7].

Interoperability can be classified in a number of different ways:

- Syntactic versus Semantic
- Automatic versus Manual
- Pipelined versus Interactive
- Vertical versus Lateral



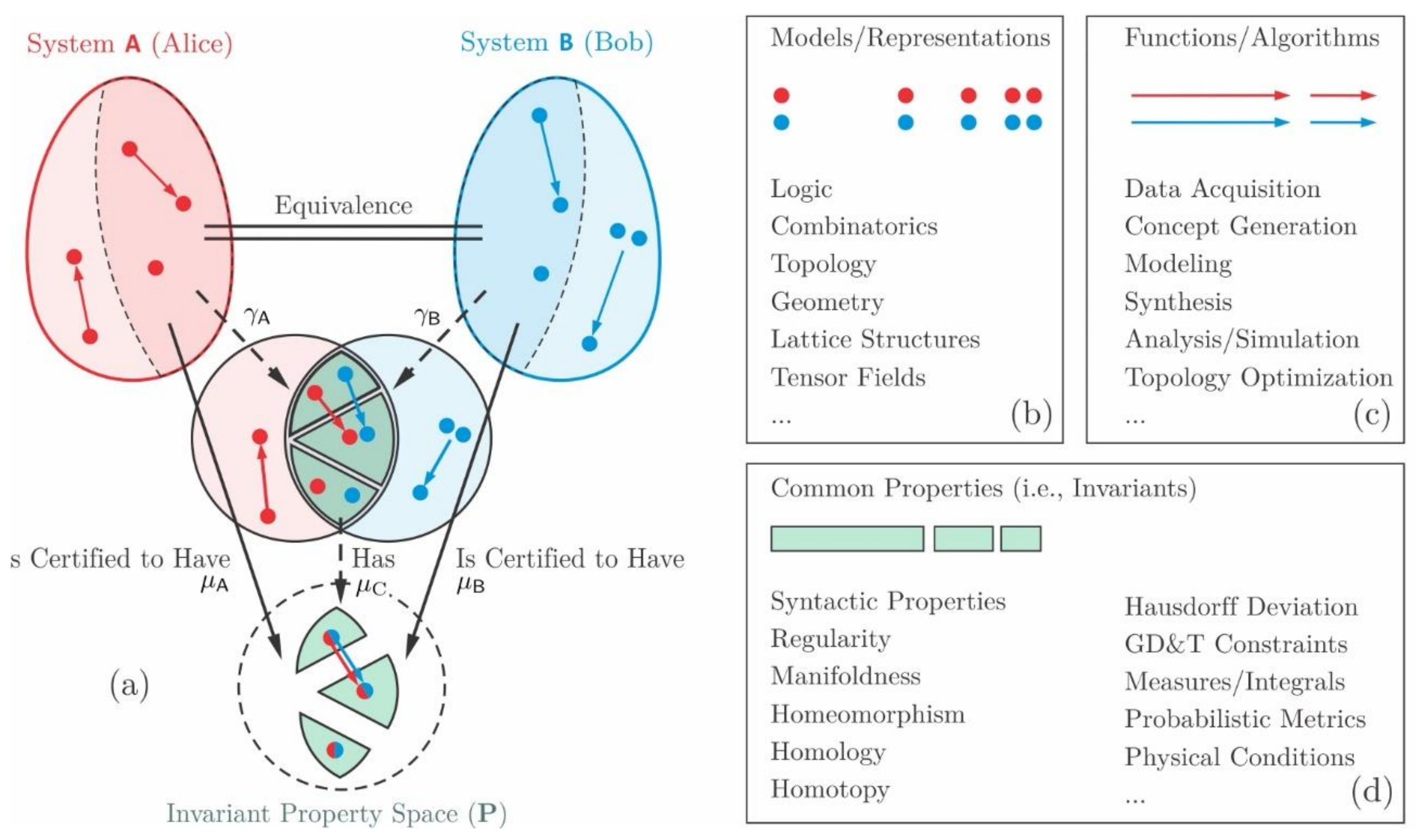
Anatomy of Interoperability Scenario



Currently: "patch-work"

How to Plug & Play?

Foundations



Interchangeability:

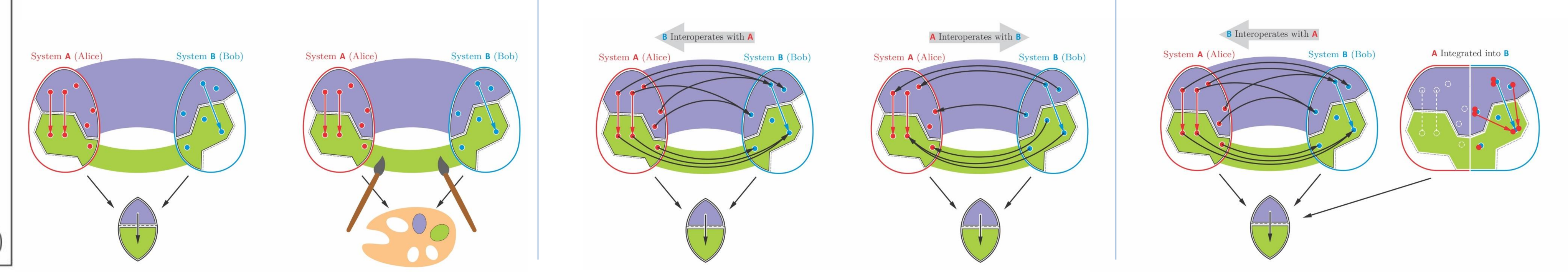
- is an equivalence relation;
- w.r.t. external reference:
  - is ~~modeling semantics;~~
  - is invariant properties;
- is never *total* or *exact*;
- must be decidable.

Interoperability:

- requires interchangeability;
- specific "interoperability map:"
  - must be computable;
  - can be explicit
  - can be implicit
- must preserve properties.

Integration:

- requires interoperability;
- upon pairing map i/o;
- same external reference;
- integrated system
  - interoperates with A;
  - interoperates with B;



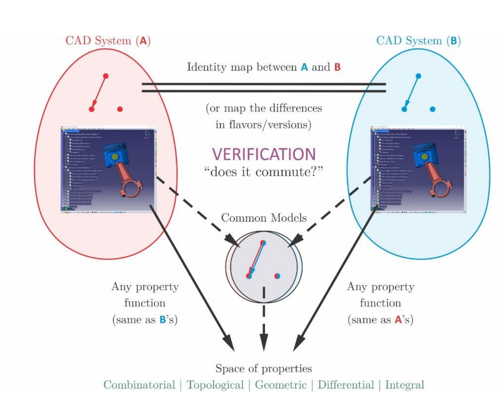
Use-Case Scenarios (CAD-CAD)

Use-Case #1: Use same reps and algs

Preserved properties:
- Everything modeled (assuming consistent *usage* in both systems)

Interoperability map:
- Identity function

Verification:
- A priori verified



Advantages:
- Strong interoperability
- Convenient (e.g., for lateral interoperability)

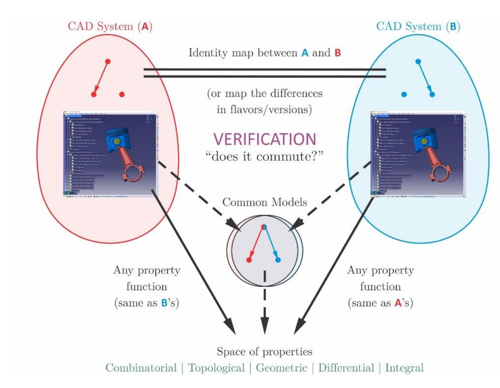
Drawbacks:
- Limited innovation
- No "hooks" for interoperating with others

Use-Case #2: Use same reps (not algs)

Preserved properties:
- Everything dependent on the identical reps + using only interchangeable algs

Interoperability map:
- Identity function

Verification:
- May require a posteriori verification



Advantages:
- Similar to Use-Case #1
- Allows *some* innovation
- Useful for version control

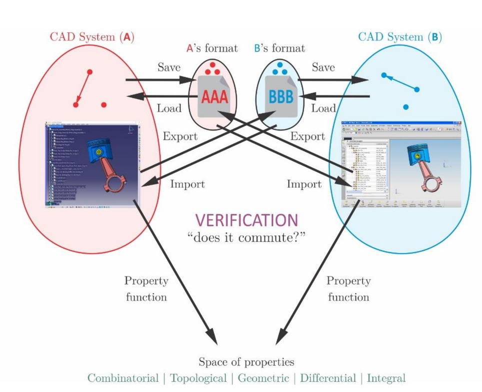
Drawbacks:
- Still limited interoperability
- No straightforward path to algorithm interoperability

Use-Case #3: Exchange formats (sys2sys)

Preserved properties
- Restricted by systems' expressive capacity

Interoperability map:
- Format translation process (export + import)

Verification:
- Requires a posteriori verification



Advantages:
- Most effective to-date
- Can optimize for a pair of systems by collaboration

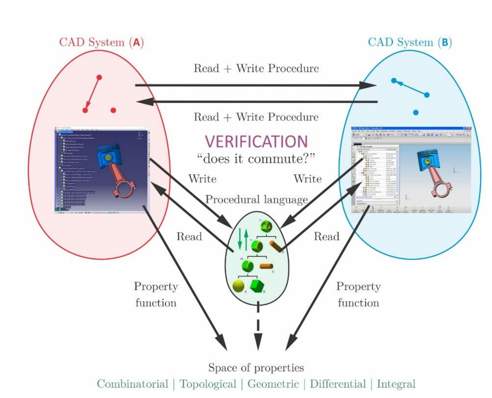
Drawbacks:
- Error accumulation
- N^2 problem: redundancy
- Hard to identify invariants

Use-Case #5: Exchange procedural recipes

Preserved properties
- Restricted to the procedural language's expressive capacity

Interoperability map:
- Procedure re-evaluation

Verification:
- Requires a posteriori verification



Advantages:
- Mostly symbolic exchange, less numerics = less error
- Enables standardization

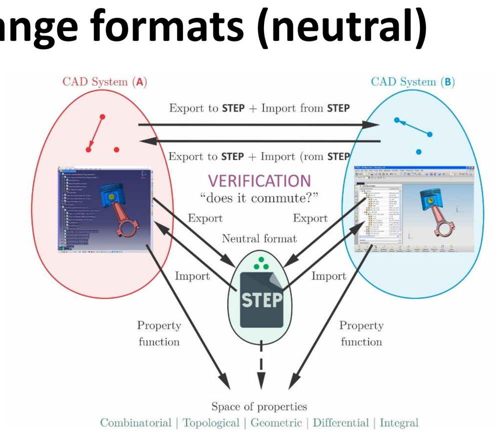
Drawbacks:
- Semantic inconsistencies
- Hard to maintain persistent references

Use-Case #4: Exchange formats (neutral)

Preserved properties
- Restricted to the neutral file's expressive capacity

Interoperability map:
- Format translation process (export + import)

Verification:
- Requires a posteriori verification



Advantages:
- N^2 problem reduces to N
- Enables standardization
- Complete task decoupling

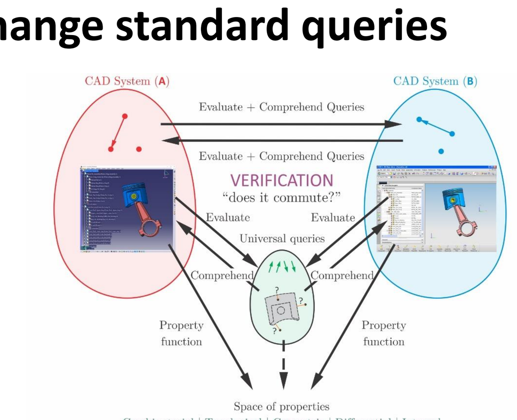
Drawbacks:
- Error accumulation
- Limited to expressiveness of neutral file format

Use-Case #6: Exchange standard queries

Preserved properties
- Restricted to inferable from queries

Interoperability map:
- Query evaluation and comprehension

Verification:
- May require a posterior verification



Advantages:
- Adaptable, extendable, resilient, self-correcting
- Proprietary protection

Drawbacks:
- Less understood
- Requires infrastructural changes (nontrivial)

Use-Case Scenarios (CAD-CAx)

CAD-CAx Interoperability:

- Geometry as a *surrogate*
- Interchangeability of specifications vs solutions

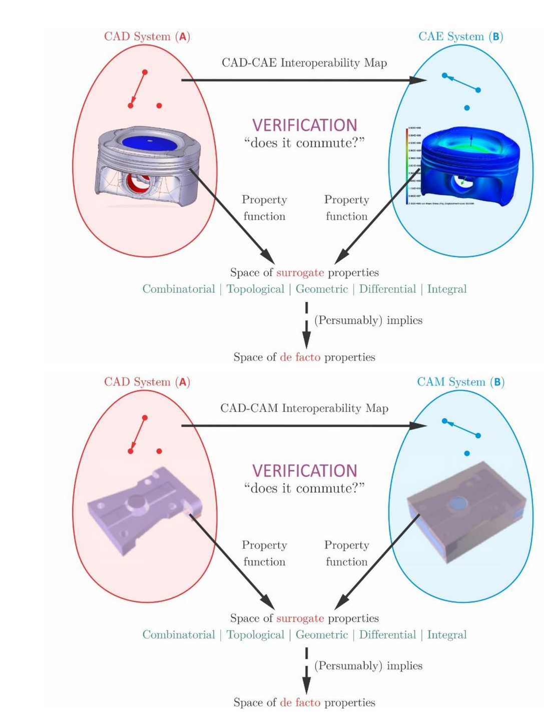
e.g., CAD-CAE Integration:

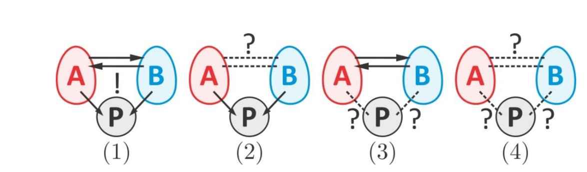
- Boundary conditions
- Integral computations
- …

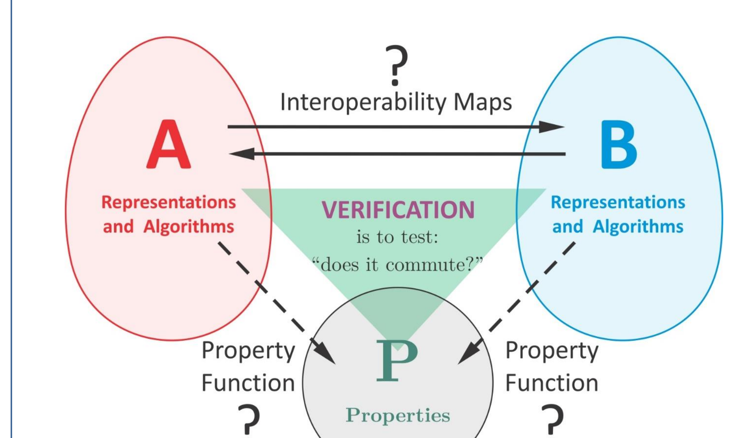
e.g., CAD-CAM Integration:

- Accessibility analysis
- Motion planning
- …

e.g., Multi-scale Material Modeling …







Types of Problems:

Depending on missing info:

1) VERIFICATION
2) CORRESPONDENCE
3) CHARACTERIZATION
4) INNOVATION

Conclusions

- External reference for common semantics is indisputable.
- Established anatomy of a generic interoperability scenario:
  - invariant properties (specified explicitly);
  - preserved via interoperability map (explicit or implicit).
- Outcomes: **formal abstractions** for semantic interoperability for
  - seamless integration of CAD, CAE, CAM, and system engr.;
  - reconfigurable and autonomous computational systems;
  - supporting multi-scale and multi-physical design with new processes (e.g., AM) & materials (e.g., knitted composites).

Acknowledgements

1. DARPA Seedling: "Interoperability Challenges and Scenarios in Computational Design and Manufacturing (CDM)." PI: Vadim Shapiro
2. NSF Award CMMI-1547189: "EAGER/Cybermanufacturing: Abstractions and Architectures for Open Composable Services." PI: Vadim Shapiro
3. NSF Award CMMI-1361862: "Developing the Foundations and Systems for Facilitating Geometric Interoperability." PI: Vadim Shapiro

[1] IEEE 610-1990. IEEE standard computer dictionary. A compilation of IEEE standard computer glossaries, Institute of Electrical and Electronics Engineers (IEEE), 1990.
[2] S. B. Brunnermeier and S. A. Martin. Interoperability costs in the US automotive supply chain. Supply Chain Management: An International Journal, 7(2):71–82, 2002.
[3] S. J. Kemmerer. STEP: The Grand Experience. US Department of Commerce, Technology Administration, National Institute of Standards and Technology (NIST), 1999.
[4] B. Smith, K. Brauner, P. Kennicott, M. Liewald, and J. Wellington. Initial graphics exchange specification (IGES) version 2.0. Technical report, NTIS, Springfield, VA, USA, 1983.
[5] C. M. Hoffmann, V. Shapiro, and V. Srinivasan. Geometric interoperability via queries. Computer-Aided Design, 46:148–159, 2014.
[6] M. Freytag, M. Shapiro, and I. Tsukanov. Finite element analysis in situ. Finite Elements in Analysis and Design, 47(9):957–972, 2011.
[7] S. Nelaturi and V. Shapiro. Representation and analysis of additively manufactured parts. Computer-Aided Design, 67-68:13–23, 2015.
[8] X. Liu and V. Shapiro. Homogenization of material properties in additively manufactured structures. Computer-Aided Design, 78:71–82, 2016.