The complexity of current engineered systems has increased drastically over the last decades. To tackle to this complexity, these systems are typically developed in a collaboration involving stakeholders from different domains. Such collaborative endeavors are severely hindered by inconsistencies that arise due to semantic overlap between different models.

To cope with this problem, we propose an inconsistency management approach for better understanding how inconsistencies arise, evolve and how they should be managed, including inconsistency tolerance. The core of our approach is a rich process modeling formalism that allows modeling multiple aspects of the development workflow. The efficient and inconsistency-free cross-domain collaboration is achieved by optimizing the original process for various optimality criteria, such as consistency and development costs.

We support our approach with an open-source prototype tool for modeling engineering processes, specifying inconsistency patterns and their respective management alternatives; and for enacting the optimized process for orchestration and tool interoperability.

**Process optimization**

- Formalism for modeling processes
- Multi-objective Design Space Exploration
- Example:
  1. During the mechanical design, the battery mass property is used
  2. The simulation of the electrical model modifies the battery capacity
  3. The battery capacity influences the battery mass
  4. The information used during the mechanical design may be incorrect

**Prototype tool**

- Specification and analysis
  - Ontology-aided approach
  - Reuse of domain-knowledge
  - Open-source (EPL)
  - https://github.com/david-istvan/icm

- Runtime architecture
  - Tool interoperability
  - Process orchestration
  - Explicitly modeled execution environment
  - Models@run.time principles
  - Protocol automata

**Inconsistency tolerance**

- Temporal tolerance
  - Inconsistencies are stateful entities that might occur, evolve and later potentially disappear as the natural consequence of the design workflow.

- Parameter tolerance
  - Relaxed parameter constraints
  - Composition of tolerance rules

- Spatial tolerance
  - Smart scoping
  - Relaxed impact propagation

- Composition of tolerance rules

**Consistency management: correctness vs efficiency**

- Correctness
  - Ensure that the system satisfies the required properties
  - Relationships in the semantic domain

- Efficiency
  - Maximize the appropriate performance measures
  - Typically multi-objective problems

**Prototype tool**

- Specification and analysis
  - Ontology-aided approach
  - Reuse of domain-knowledge
  - Open-source (EPL)
  - https://github.com/david-istvan/icm

- Runtime architecture
  - Tool interoperability
  - Process orchestration
  - Explicitly modeled execution environment
  - Models@run.time principles
  - Protocol automata

**Prototype tool**

- Specification and analysis
  - Ontology-aided approach
  - Reuse of domain-knowledge
  - Open-source (EPL)
  - https://github.com/david-istvan/icm

- Runtime architecture
  - Tool interoperability
  - Process orchestration
  - Explicitly modeled execution environment
  - Models@run.time principles
  - Protocol automata