Advanced Techniques and Tools for Secure Collaborative Modeling

Csaba Debreceni

Supervisor
Prof. Daniel Varro, D.Sc., Ph.D.

Co-Supervisors: Gabor Bergmann, Ph.D., Istvan Rath, Ph.D.

Hungarian Academy of Sciences

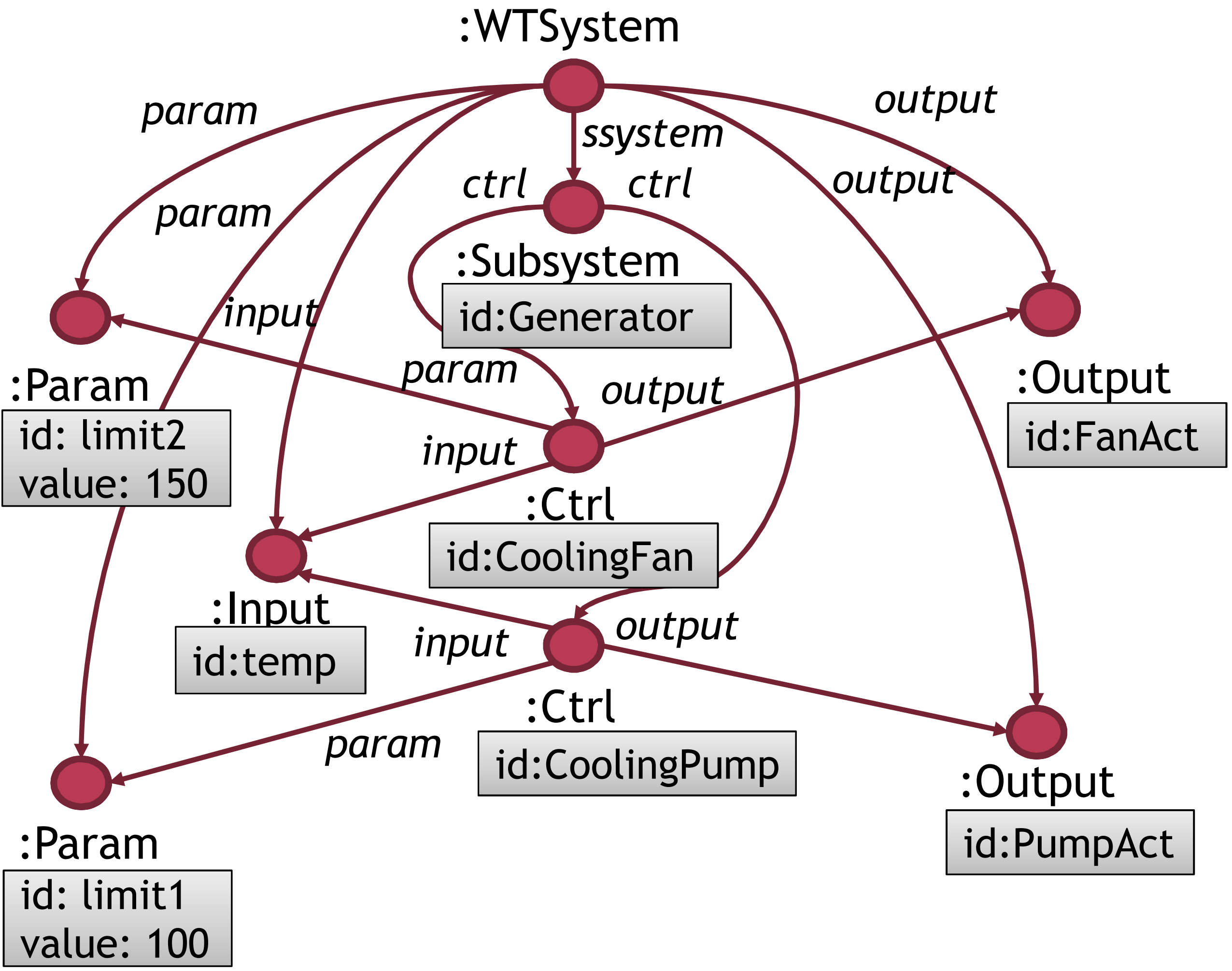
Collaborative Modeling

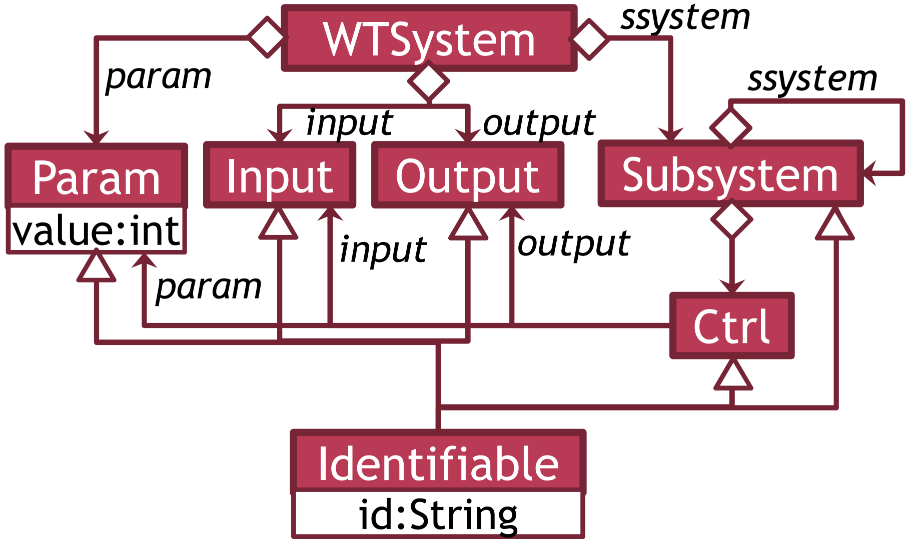
Stakeholders with different *responsibilities, clearances*

Concurrently working on the complex system model

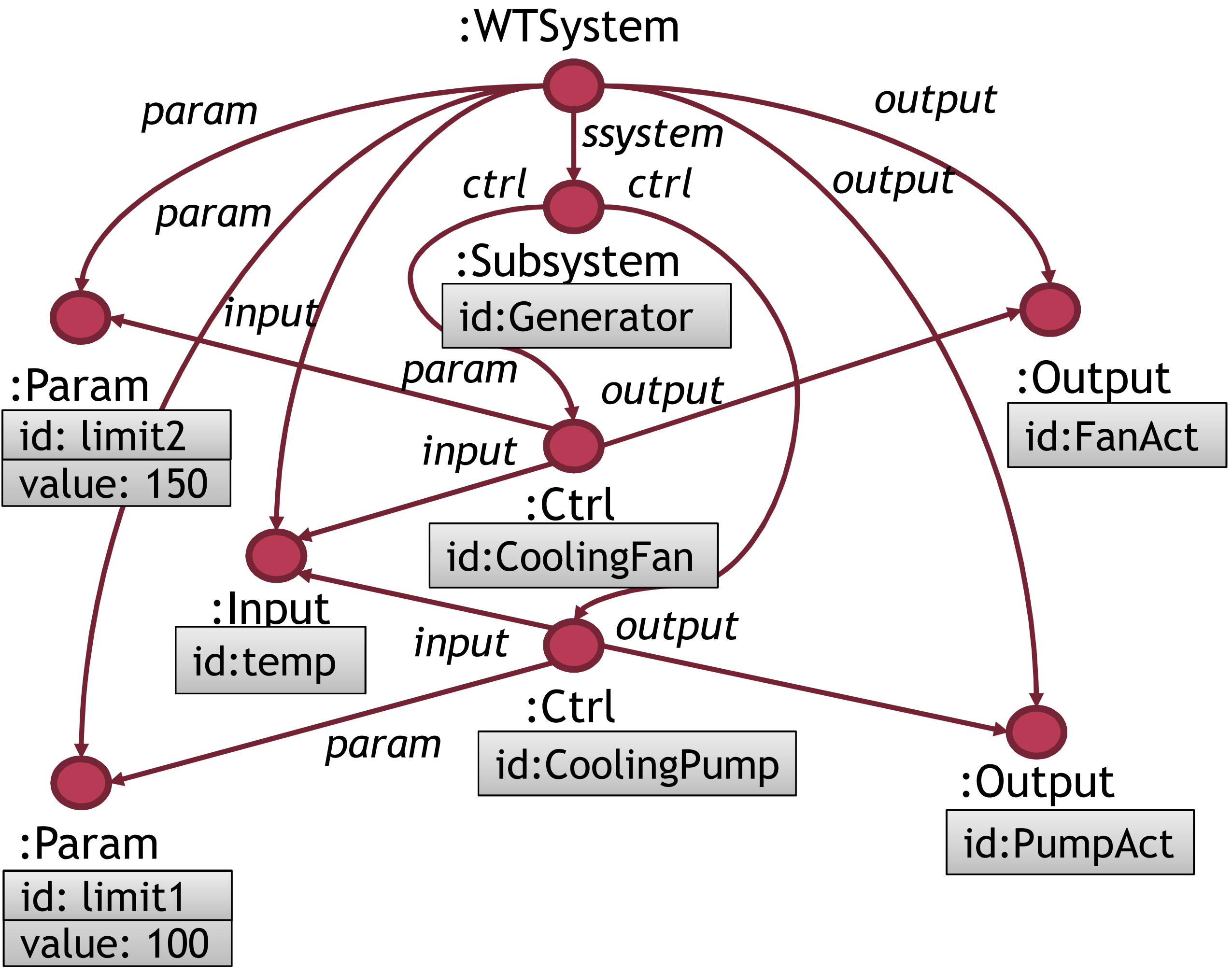
System Engineers with various *domain-specific knowledge*

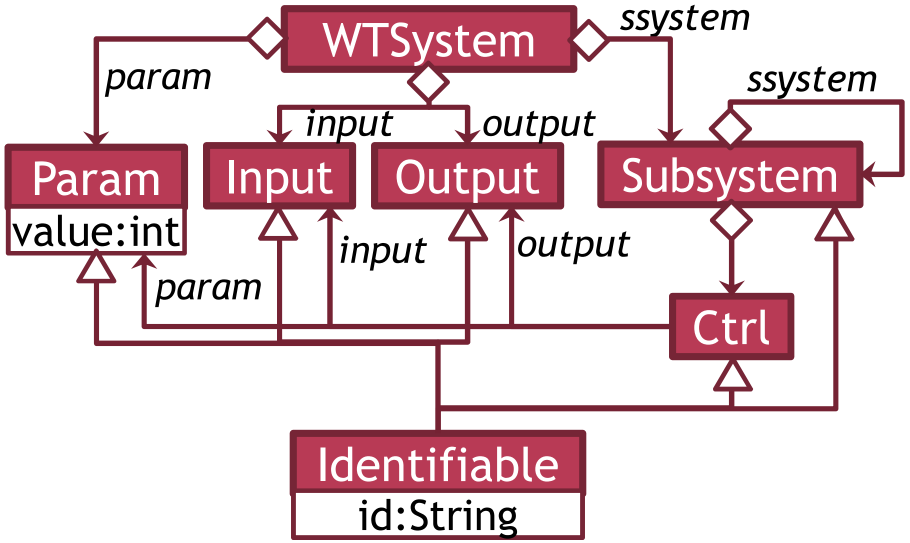
Preliminaries – Wind Turbine Case Study





Preliminaries – Wind Turbine Case Study

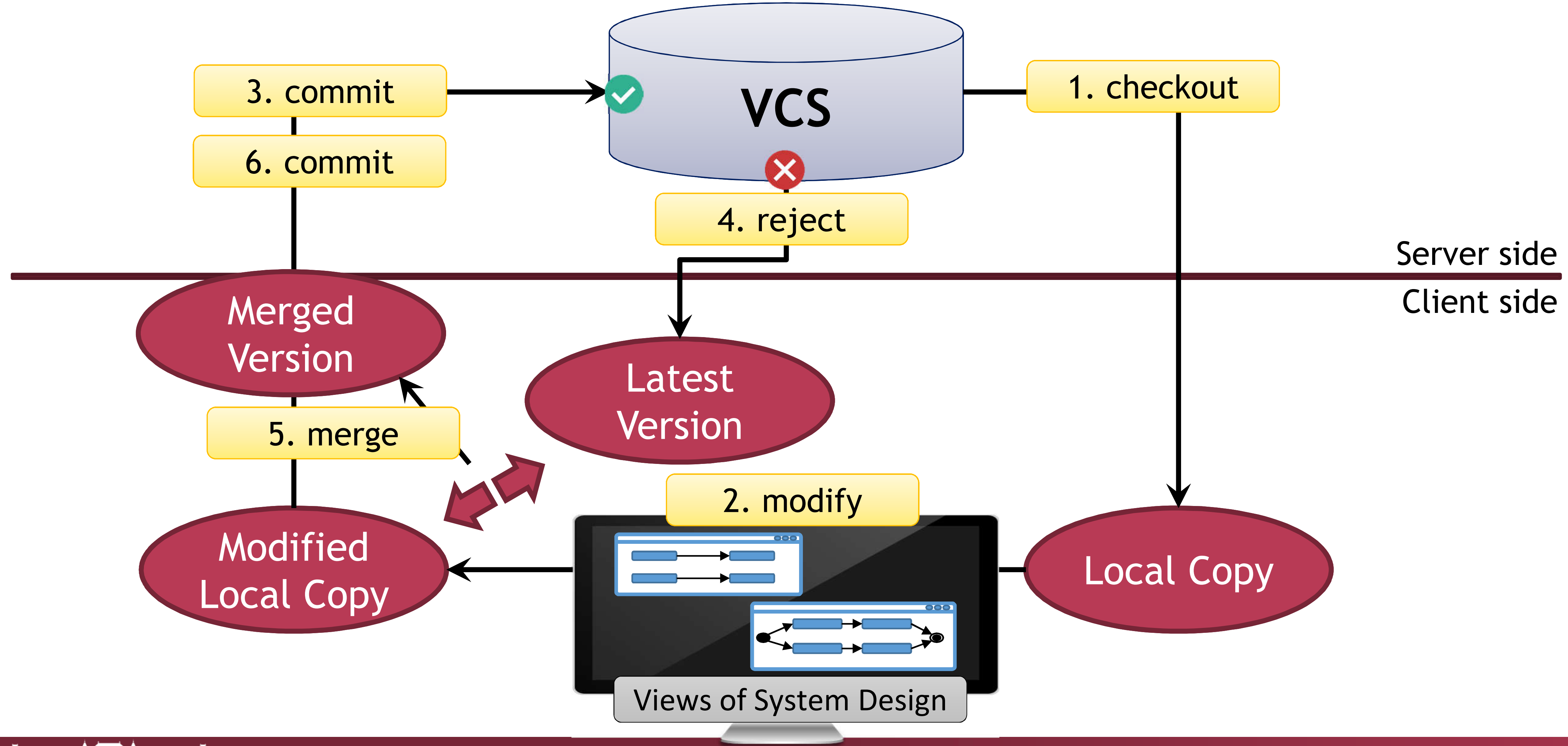




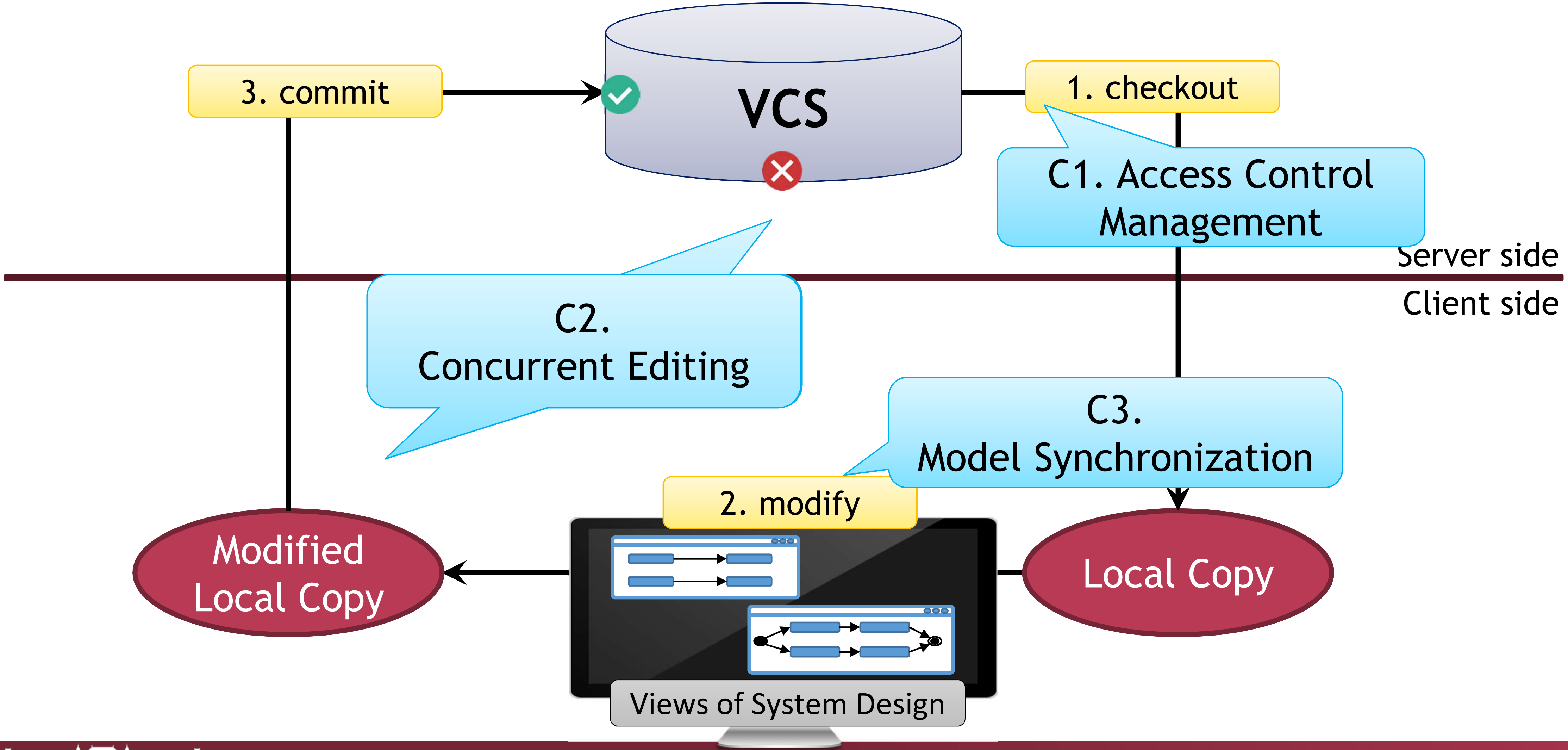
Graph Queries in VIATRA syntax

```
pattern ioObjects (io, id) {
  Param.id(io, id);
} or {
  Input.id(io, id);
} or {
  Output.id(io, id);
}
```

General Collaborative Scenario



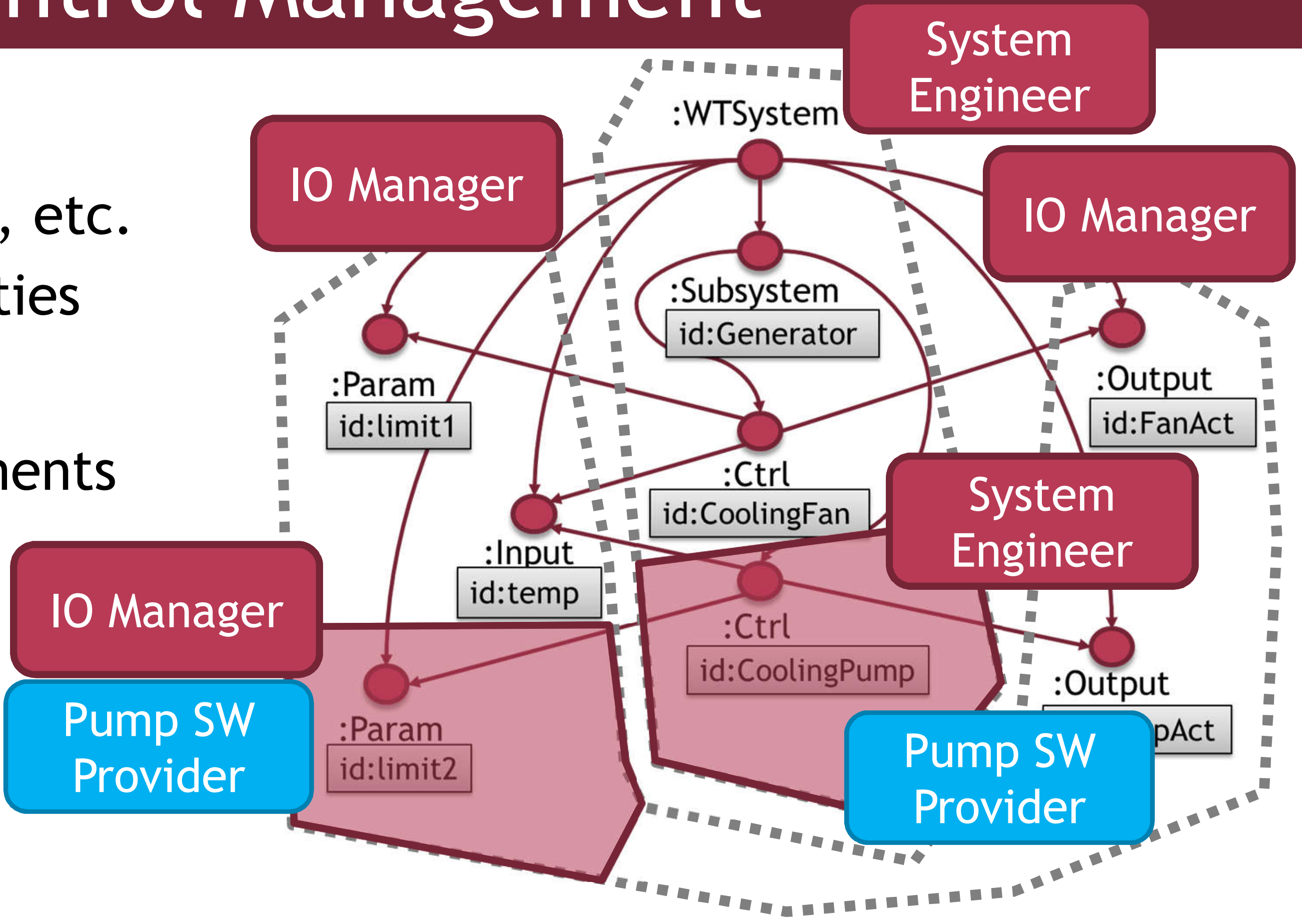
General Collaborative Scenario



ACCESS CONTROL MANAGEMENT

Access Control Management

- Motivation
  - Different stakeholders, teams, etc.
  - Protecting Intellectual Properties
- State of the art
  - Serialize the model into fragments
  - Assign R/W to fragments
- Problem
  - Inflexible fragmentation
  - 1000+ fragments



Challenge I - How to **specify** and **enforce** high-level **access control policies** during collaborative modeling in a scalable way?

Fine-grained Access Control Policies

- Graph Queries are used to select assets
  - Assets: object, reference, attribute
- Rules are applied on assets
  - Access: allow > obfuscate > deny
  - Operation: read (R) / write (W)
- Options:
  - Defaults, External / Internal priorities
- Effective Permissions
  - Resolve ***contradicting rules*** and respect ***internal consistencies***

```
policy Example deny RW by default {
    rule permitIO allow RW to IOMngr {
        from query ioObjects
        select obj(io)
    } on priority 1
} with restrictive resolution
```

```
pattern ioObjects (io)
{Param(io)} or {Input(io)}
or {Output(io)}
```

C. Debreceni, G. Bergmann, I. Ráth, and D. Varró, “Deriving Effective Permissions for Modeling Artifacts from Fine-grained Access Control Rules,” in COMMitMDE @MoDELS’16

Contribution 1

Fine-grained Access Control Policies

I proposed a domain-customizable modeling language to capture fine-grained access control policies and I realized a framework to efficiently evaluate the policies in online and offline scenarios.

1.1 Access Control Language. I proposed a rule-based access control language to describe high-level and fine-grained policies in both online and offline scenarios. Rules may allow, obfuscate or deny read and/or write permissions of model parts identified by graph patterns.

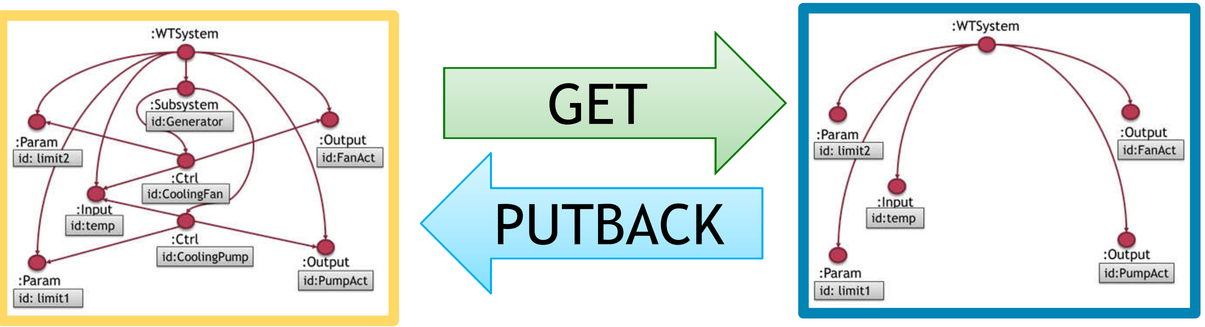
1.2 Read and Write Dependencies. I analyzed read and write dependencies implied by high-level access control policies as read and write permissions of a model part may depend on other model parts implied by internal consistency rules.

1.3 Deriving Effective Permissions. I implemented a prototype framework to derive a set of effective permissions from access control policies in the context of models providing batch and incremental evaluation to support offline and online collaboration, respectively.

1.4 Evaluation. I evaluated the scalability of the proposed prototype framework on a case study of offshore wind turbine controllers.

Provenly Secure Collaboration Scheme

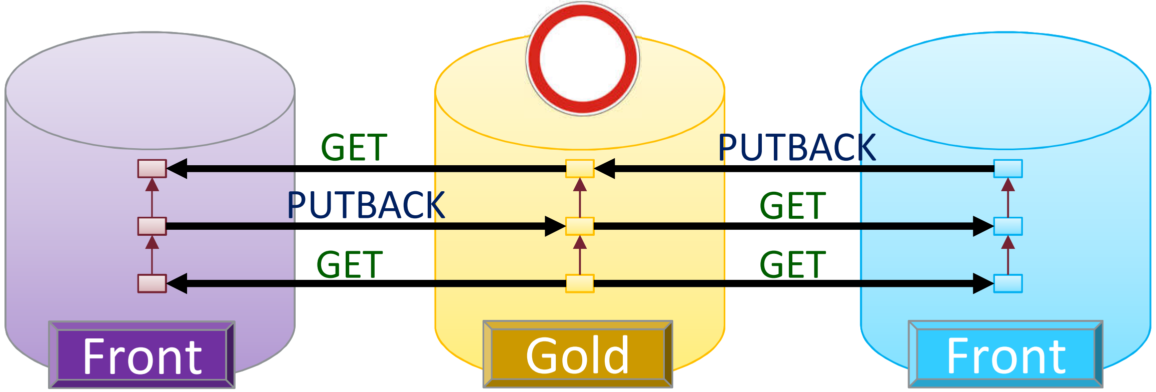
- Bidirectional Model Transformation – Lens
  - Gold Model – all information
  - Front Model – filtered model
  - GET / PUTBACK
    - Check Read permissions
    - Check Write permissions



„Gold” model

„Front” model for **IO Mngr**

- General Collaboration Scheme
  - Gold Repository – no access
  - Front Repository – front revisions



C. Debreceni, G. Bergmann, I. Ráth, and D. Varró, “Enforcing fine-grained access control for secure collaborative modelling using bidirectional transformations,” in SoSym’17

Contribution 2

Provenly Secure Collaboration Scheme

I formalized the enforcement of high-level fine-grained access control policies and realized provenly secure collaborative architecture that enforces such policies.

2.1 Formalization of Bidirectional Rules for Secure Views. I formalized transformation rules to derive secure front models with respect to the read and write permissions.

2.2 Secure Collaboration Scheme. I formalized a collaboration scheme as communicating sequential processes (CSP) to enforce high-level access control policies. I specified correctness criteria and proved the correctness of the scheme.

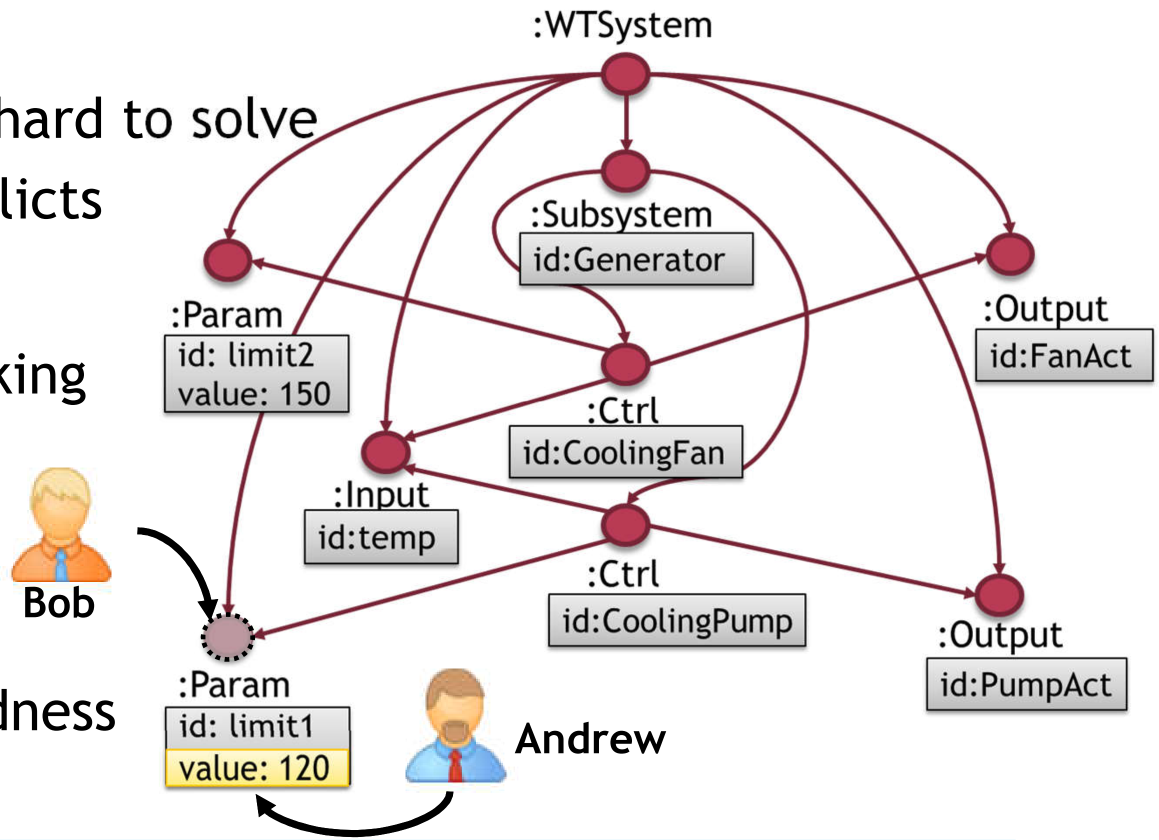
2.3 Realization of Secure Collaboration. I realized the collaboration scheme in case of offline scenarios by extending an existing version control system to enforce fine-grained access control while collaborators can use off-the-shelf tools.

2.4 Evaluation. I evaluated the scalability of the proposed architecture on a case study of offshore wind turbine controllers.

CONCURRENT EDITING

Concurrent Editing

- Motivation
  - Conflicts: Easy to introduce, hard to solve
  - Prevent and / or resolve conflicts
- State of the art
  - Object / Fragment based locking
  - Semi-automated resolutions
- Problem
  - Overlocking / Underlocking
  - Transformation / Well-formedness

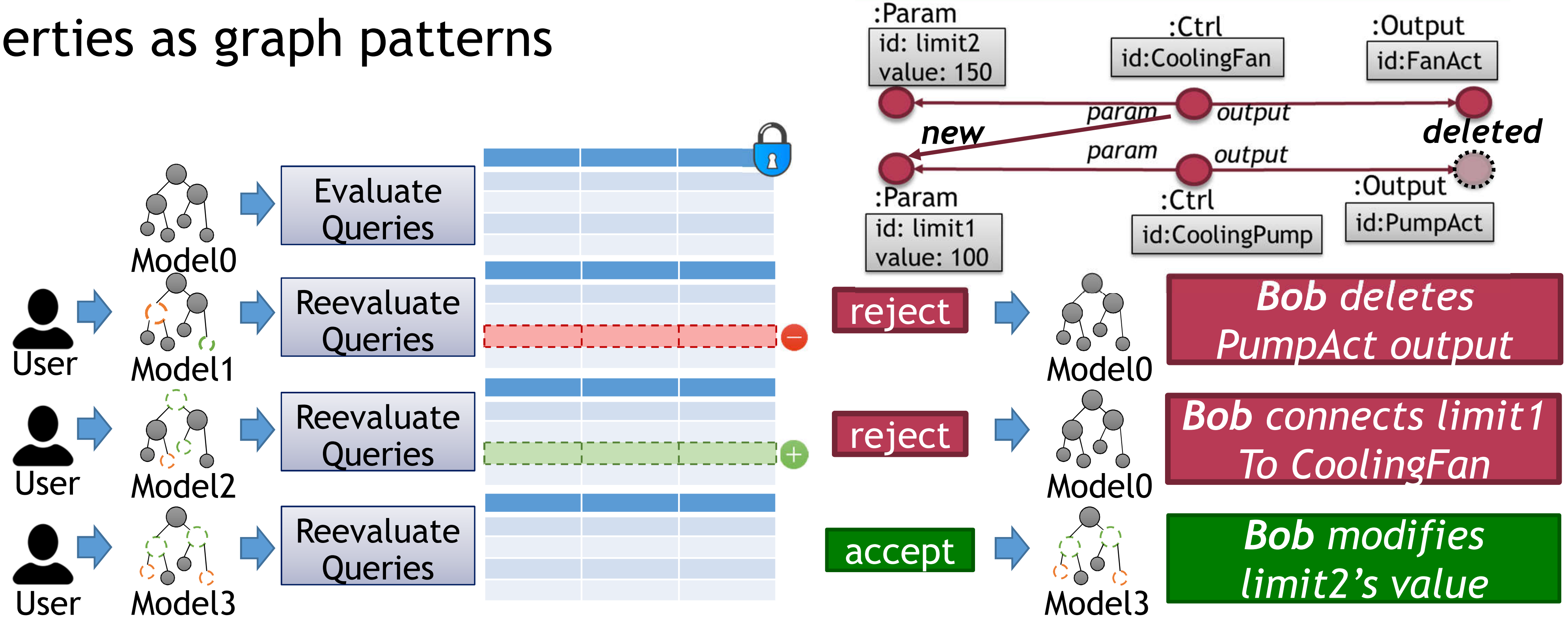


Challenge II - How to provide **fine-grained prevention** and **automated resolution** strategies of conflicts?

Property-based Locking

- Lock: Lock owner + Property
- Properties as graph patterns

Andrew locks the property: „A param is used for producing certain output."



Bob deletes PumpAct output

Bob connects limit1 To CoolingFan

Bob modifies limit2's value

C. Debreceni, G. Bergmann, I. Ráth, and D. Varró, "Property-based locking in collaborative modeling," in MoDELS'17

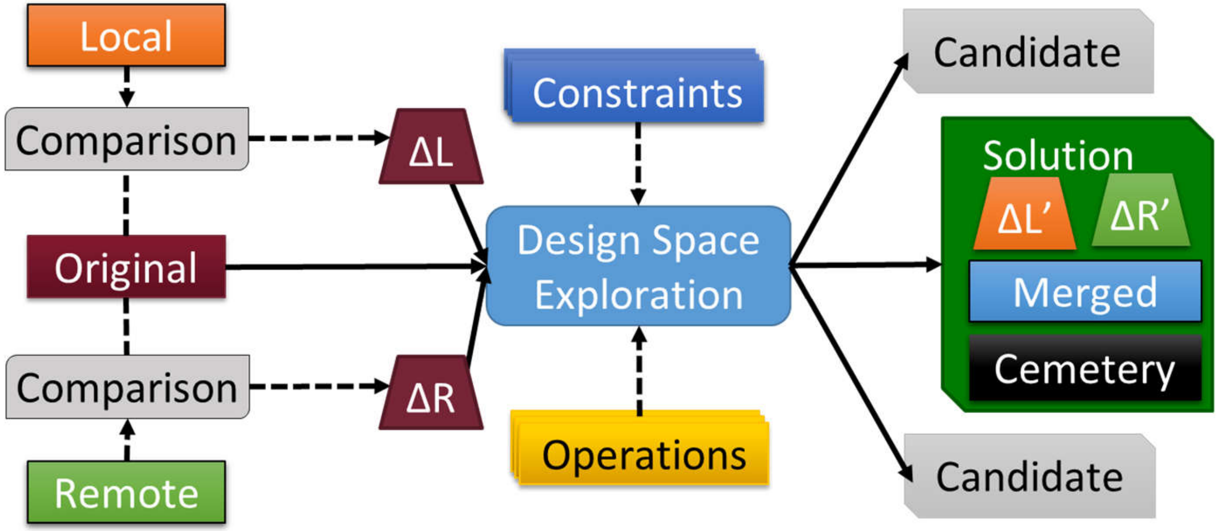
Automated Model Merge using DSE

- Automated Model Merge
  - ○ Design Space Exploration

    1) Initial Model, 2) Goals,

    3) Exploration Rules, 4) Constraints

    5) Design Candidates, 6) Solution
  - ○ Model Merge as DSE



Local

Comparison

ΔL

Original

Comparison

ΔR

Remote

Constraints

Design Space Exploration

Operations

Candidate

Solution

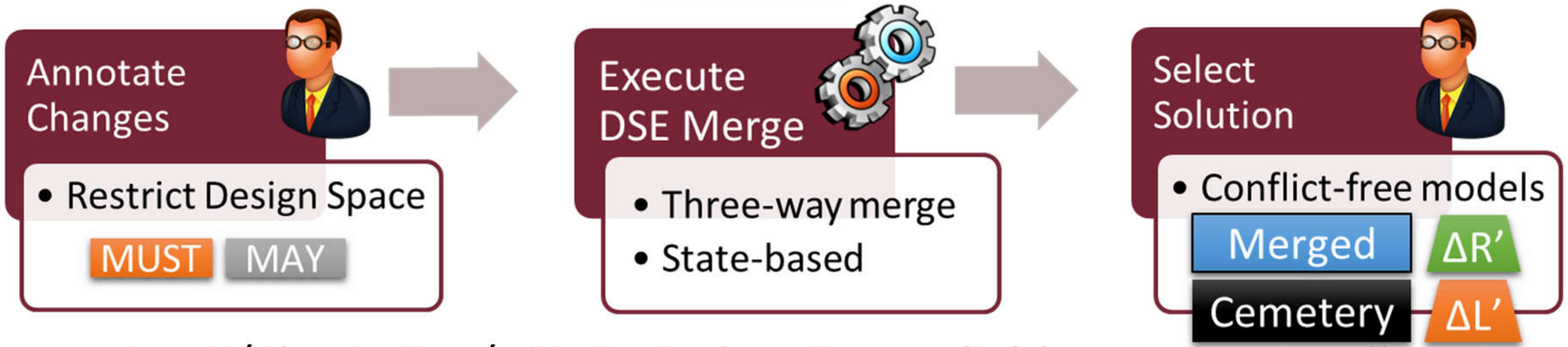
ΔL’

ΔR’

Merged

Cemetery

Candidate



Annotate Changes

- Restrict Design Space

MUST MAY

Execute DSE Merge

- Three-way merge
- State-based

Select Solution

- Conflict-free models

Merged ΔR’

Cemetery ΔL’

C. Debreceni, I. Ráth, D. Varró, X. D. Carlos, X. Mendialdua, and S. Trujillo, “Automated model merge by design space exploration,” in FASE’16

Contribution 3

Property-based Locking and Automated Model Merge using DSE

I proposed a fine-grained property-based locking technique to avoid conflicts and an automated three-way model merge technique to resolve conflicts.

3.1 Fine-grained Property-based Locking. I proposed a property-based locking technique as generalization of traditional fragment-based and object-based locking techniques which captures fine-grained locks as graph patterns and exploits incremental query engines to maintain and evaluate locks.

3.2 Automated Model Merge using DSE. I formalized an automated three way model merge technique by adapting rule-based design space exploration to derive consistent and semantically correct merged models.

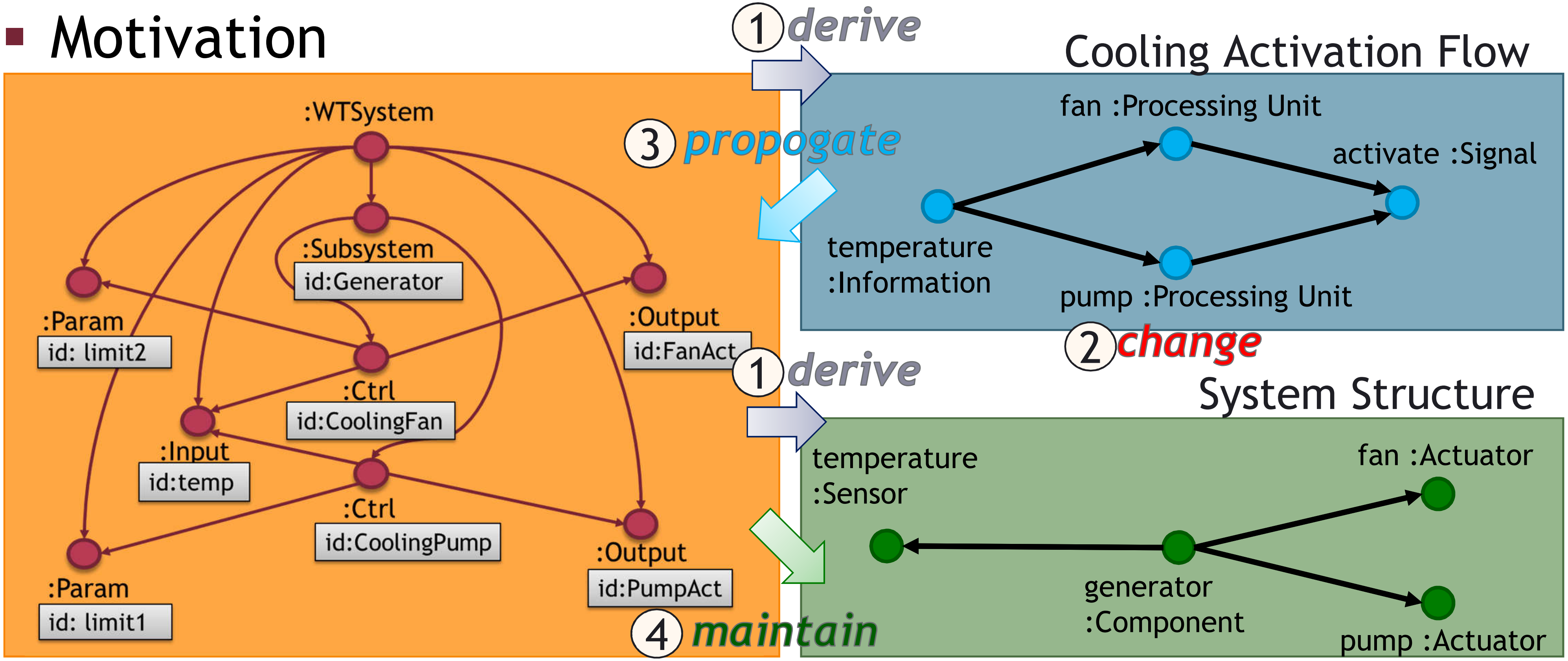
3.3 Generic Scalability Benchmark. I proposed a scalability benchmark for model merge by adapting an existing performance benchmark for model queries.

3.4 Evaluation. I evaluated the scalability of the automated model merge and I compared the effectiveness of fine-grained property-based locking and traditional locking strategies for conflict prevention on a case study of offshore wind turbine controllers.

MODEL SYNCHRONIZATION

Model Synchronization

- Motivation



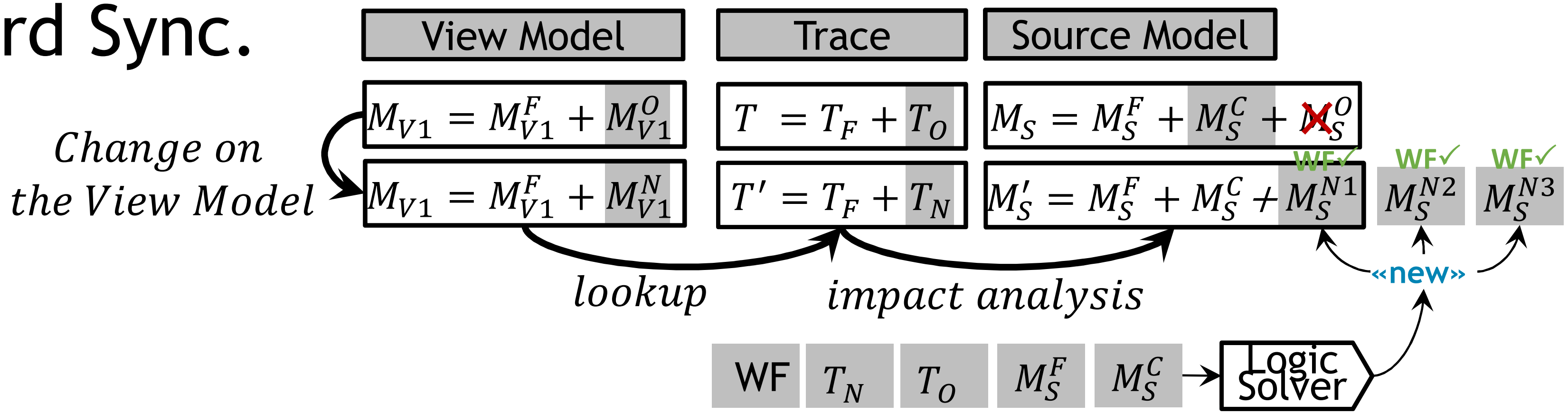
Challenge III - How to **derive** and **incrementally maintain** view models and **trace back** changes to source models?

Bidirectional Synchronization of View Models

- Derivation Rules (Query-based Object/Feature)
  - Graph Patterns with annotations
    - Precondition: a pattern match
    - Execution rule: defined in annotations

```
@QBO(ctrl, „Processing Unit”)
pattern ctrlUnits (ctrl)
{ Ctrl(ctrl); }
```

- Backward Sync.



O. Semeráth, C. Debreceni, A. Horváth, and D. Varró, “Incremental backward change propagation of view models by logic solvers,” in MoDELS’16

Contribution 4

A novel technique of bidirectional synchronization of view models

I proposed a novel technique of bidirectional synchronization of view models where the forward incremental synchronization is achieved by unidirectional derivation rules while the backward propagation of changes is generated using logic solvers.

4.1 Incremental Forward Synchronization. I formalized a fully forward incremental, unidirectional synchronization technique of view models allowing chaining of views where the object of view model depend on the match set of the precondition of derivation rules.

4.2 Change Impact Analysis. I analyzed the impact of changes in underlying source models in case of backward propagation. The impacted part is added to the logic solver as additional constraints to calculate minimally modified source model candidates.

4.3 Realization of Forward Synchronization. I realized the incremental and forward view synchronization technique where elementary derivation rules are captured by graph patterns and the reactive synchronization process uses the Viatra EVM.

4.4 Evaluation. I evaluated the scalability of the proposed approaches on case studies from the avionics and the health-care domain.

Conclusion

- Acknowledgement
  - I would like to thank my advisor, Daniel Varro for his guidance during my research. I would also like to express my gratitude to István Ráth, Gábor Bergmann, Oszkár Semeráth and Ákos Horvath along with numerous colleagues and co-authors for sharing their ideas.
  - I would like to express my gratitude for the support of the MTA-BME Lendület Cyber-Physical Systems Research Group project. This research was partially supported by the EU projects MONDO and CONCERTO, the Hungarian CERTIMOT project and a collaborative project with Embraer called TRANS-IMA.

MONDO
SCALABLE MODELLING AND MODEL MANAGEMENT ON THE CLOUD

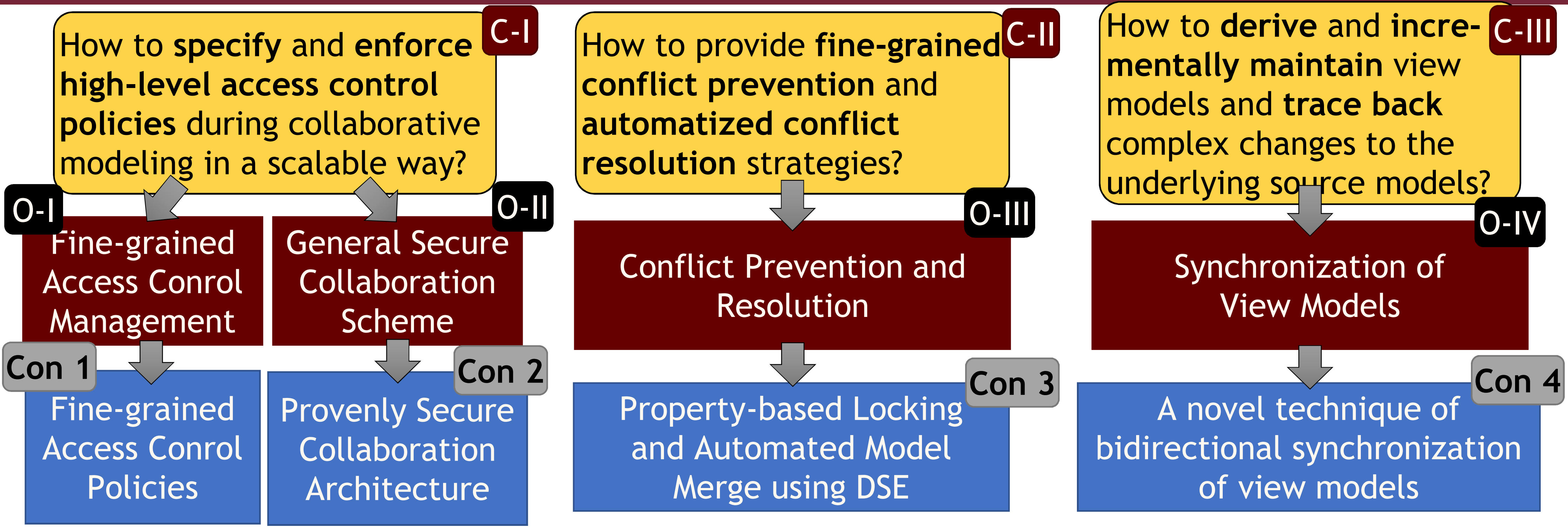
CONCERTO
Guaranteed Component Assembly with Round Trip Analysis for Energy Efficient High-integrity Multi-core Systems

EMBRAER

CertiMoT

- Tooling
  - MONDO Collaboration Framework: https://github.com/FTSRG/mondo-collab-framework
  - VIATRA Viewers Addon: https://www.eclipse.org/viatra/

Summary



C-I

How to **specify** and **enforce** **high-level access control policies** during collaborative modeling in a scalable way?

O-I

Fine-grained Access Conrol Management

Con 1

Fine-grained Access Conrol Policies

O-II

General Secure Collaboration Scheme

Con 2

Provenly Secure Collaboration Architecture

C-II

How to provide **fine-grained conflict prevention** and **automatized conflict resolution** strategies?

O-III

Conflict Prevention and Resolution

Con 3

Property-based Locking and Automated Model Merge using DSE

C-III

How to **derive** and **incre-mentally maintain** view models and **trace back** complex changes to the underlying source models?

O-IV

Synchronization of View Models

Con 4

A novel technique of bidirectional synchronization of view models

- Publication List
  - No. peer-reviewed publications: ***21***
  - No. independent citations: ***68***
  - No. journals indexed by WoS/Scopus: ***3***
  - No. int. conference papers: ***16***
  - No. Additional papers: ***2***