Consistency Maintenance for Collaborative Real-Time Feature Modeling

Elias Kuiter

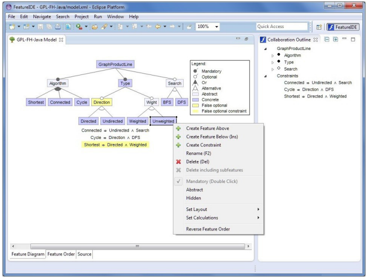
April 23, 2019

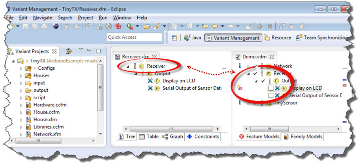
Bachelor Thesis

Advisors: Gunter Saake, Sebastian Krieter, Jacob Krüger

Motivation

- State of the Art: Single-User Feature Modeling
- No dedicated support for collaboration
- *Asynchronous Collaboration* with VCS
  But: not real-time; divergence





Why Real-Time?

- Engineers can discuss the feature model with domain experts
  $\Rightarrow$ Real-time feedback

Motivation

Why Real-Time?

- Engineers can discuss the feature model with domain experts
  $\Rightarrow$ Real-time feedback
- Domain knowledge is spread across different collaborators
  $\Rightarrow$ Leverage group synergies for problem solving

Motivation

Why Real-Time?

- Engineers can discuss the feature model with domain experts
  $\Rightarrow$ Real-time feedback
- Domain knowledge is spread across different collaborators
  $\Rightarrow$ Leverage group synergies for problem solving
- VCS does not allow tight collaboration (e.g., pair programming)
  $\Rightarrow$ Complements a VCS for short-term evolution

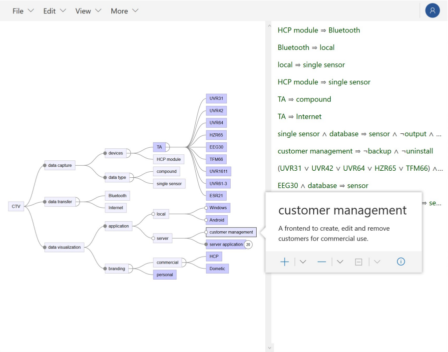
Motivation

Our Contribution:

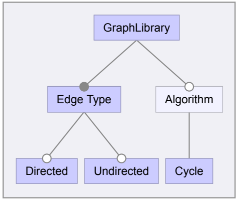
- The Foundations of Collaborative Real-Time Feature Modeling, focusing on
  - Consistency Maintenance
  - Conflict Detection & Resolution

Our Contribution:

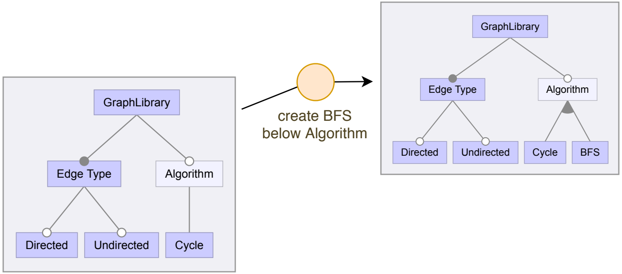
- The Foundations of Collaborative Real-Time Feature Modeling, focusing on
    - Consistency Maintenance
    - Conflict Detection & Resolution
- Open-Source Research Prototype



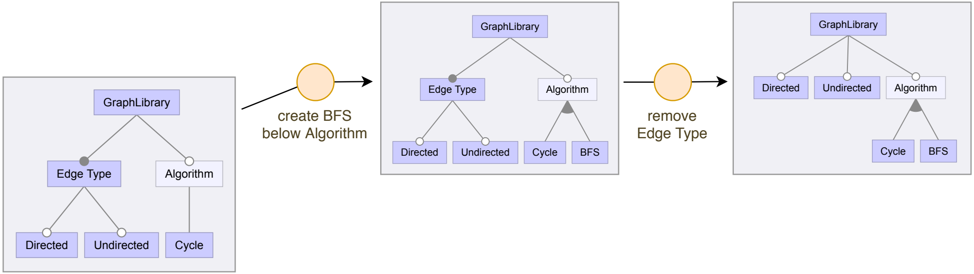
Running Example



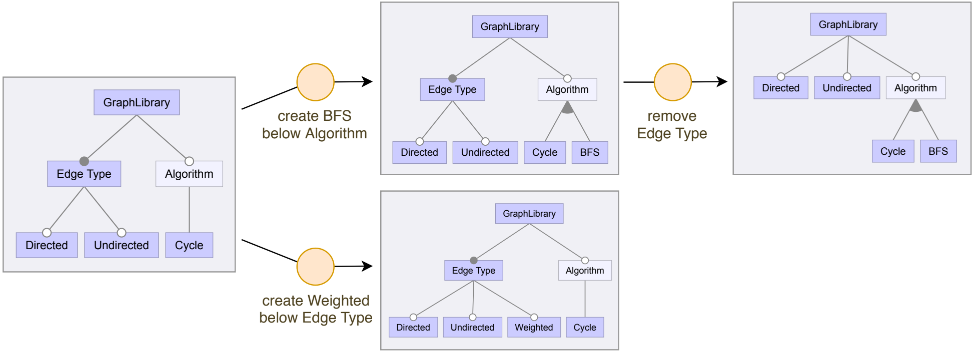
Running Example



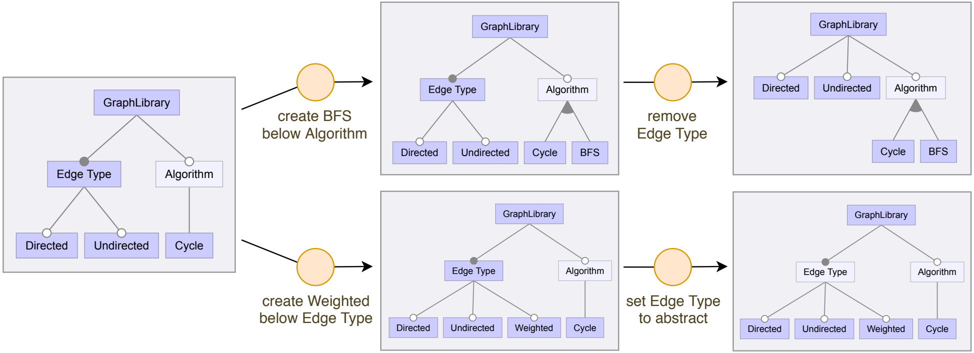
Running Example



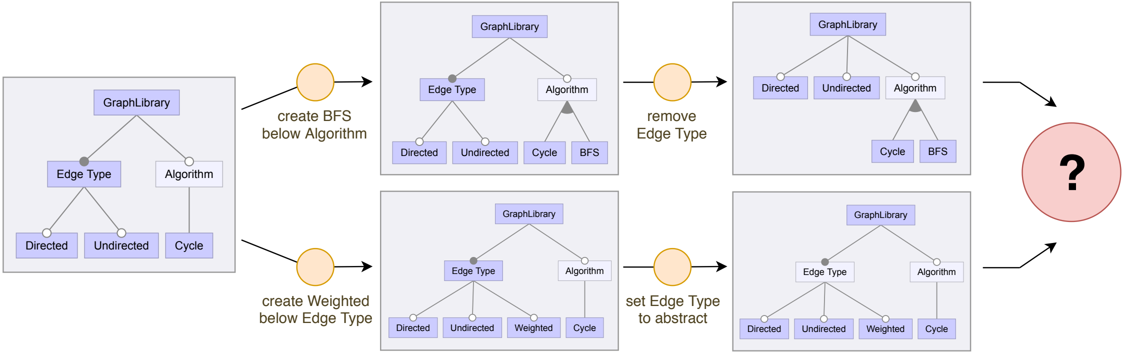
Running Example



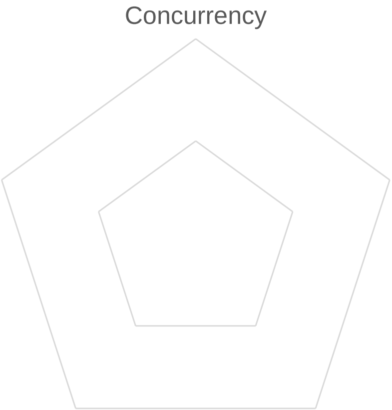
Running Example



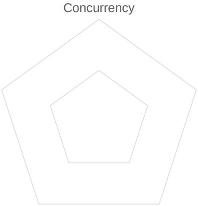
Running Example

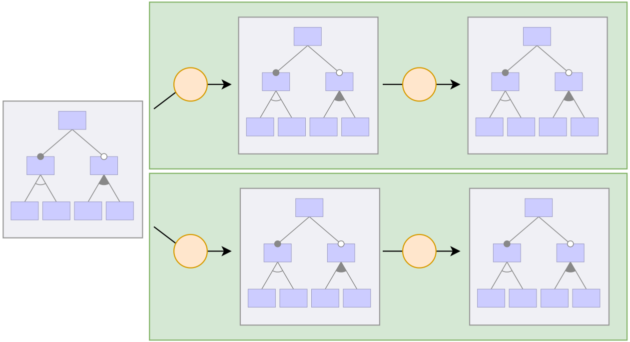


Requirements



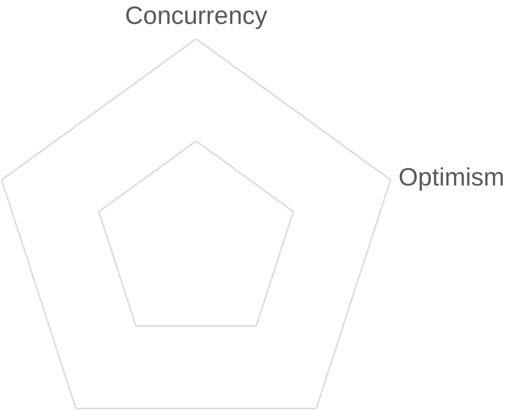
Requirements



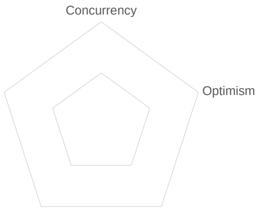


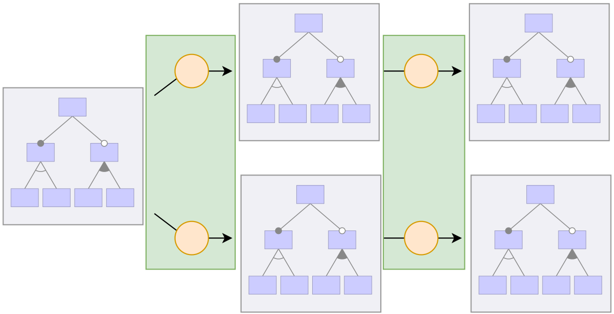
Concurrent Operation Chains

Requirements



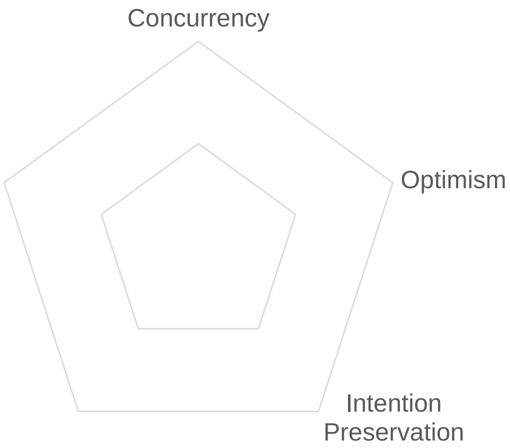
Requirements



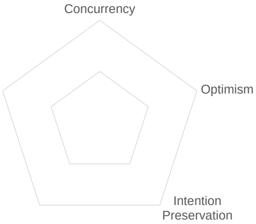


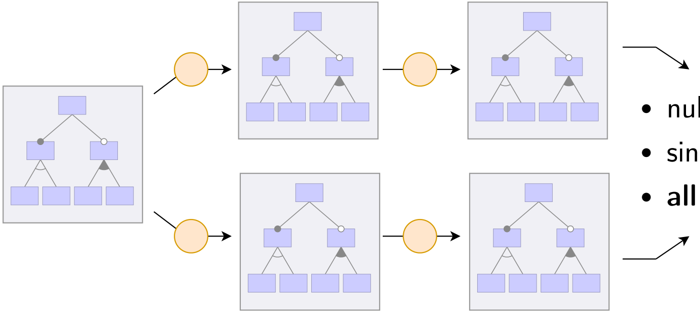
Immediate Operation Execution

Requirements



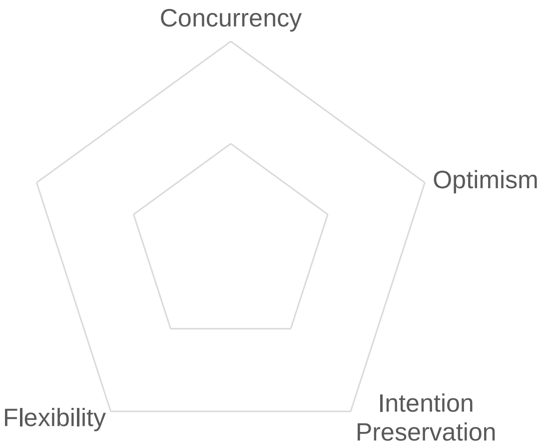
Requirements



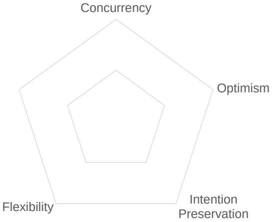


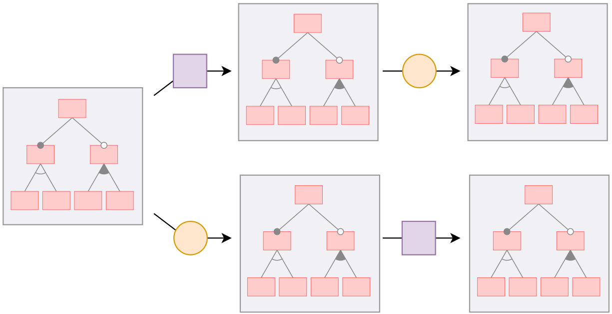
- null
- single op
- all ops

Requirements



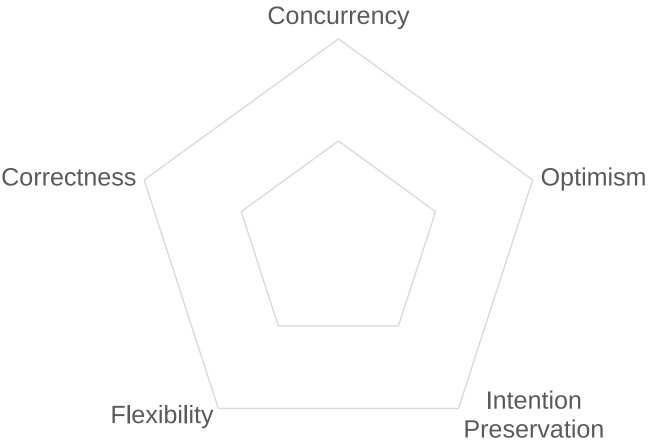
Requirements



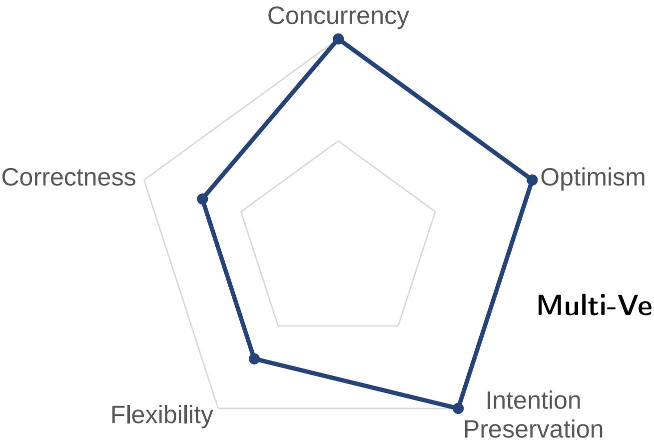


Additional Model Representations & Operations

Requirements

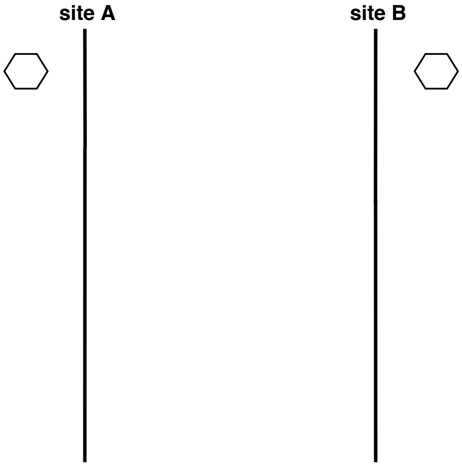


Requirements

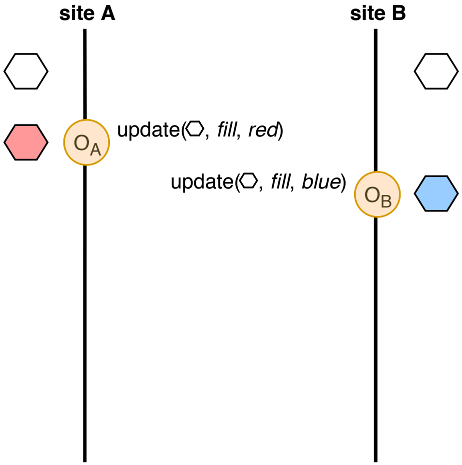


Multi-Version Multi-Display

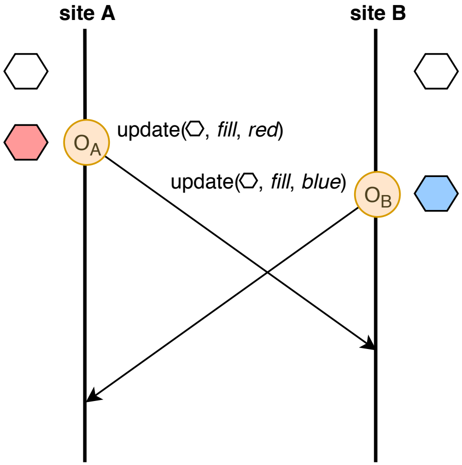
Multi-Version Multi-Display



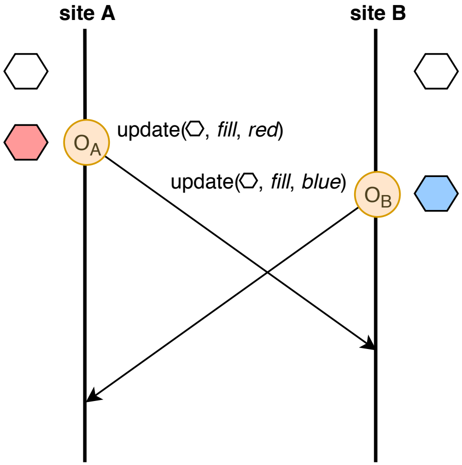
Multi-Version Multi-Display



Multi-Version Multi-Display

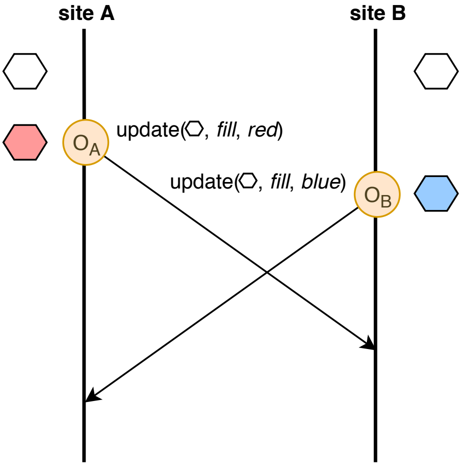


Multi-Version Multi-Display



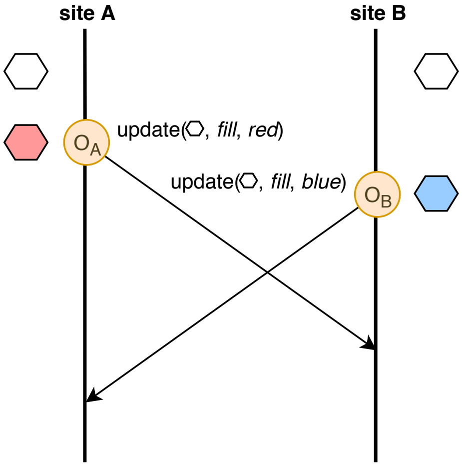
$O_A \otimes O_B$?

Multi-Version Multi-Display



$O_A \otimes O_B$? **Yes.**

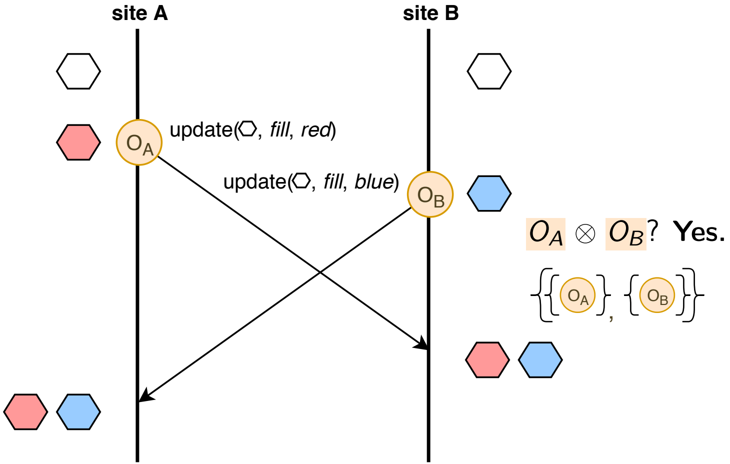
Multi-Version Multi-Display



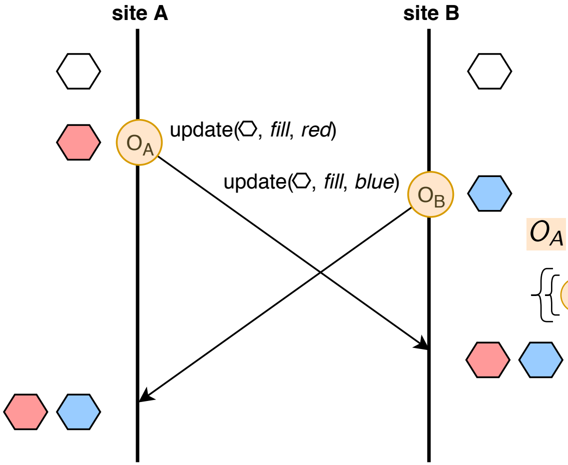
$O_A \otimes O_B$? **Yes.**

$\{\{O_A\}, \{O_B\}\}$

Multi-Version Multi-Display



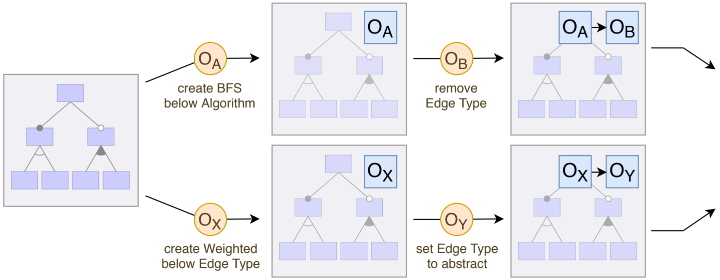
Multi-Version Multi-Display



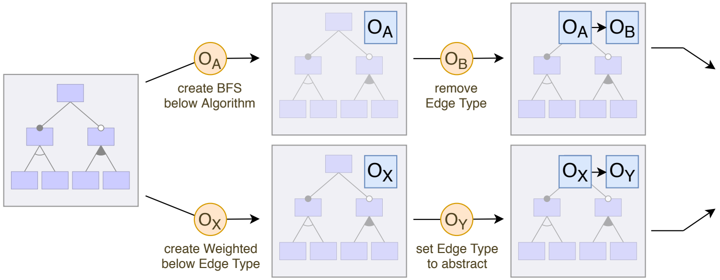
$O_A \otimes O_B$? **Yes.**

$\{\{O_A\}, \{O_B\}\}$ ⇐ guarantees min #versions

Feature Modeling Conflicts

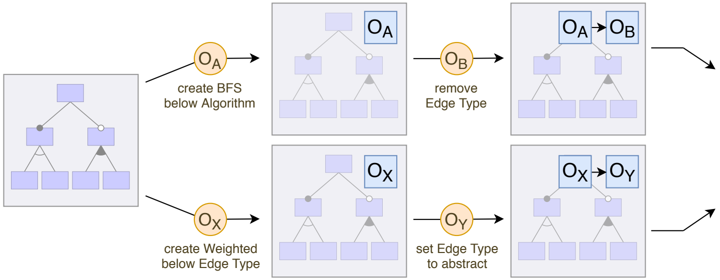


Feature Modeling Conflicts



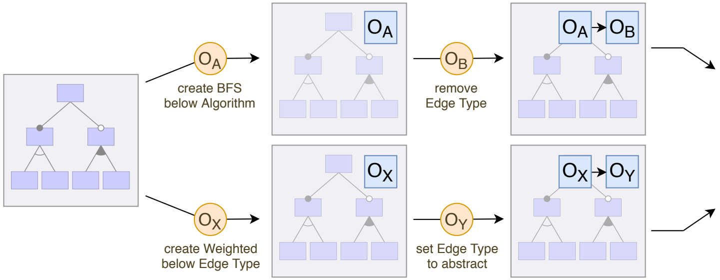
$O_A \otimes O_X$?

Feature Modeling Conflicts



$O_A \otimes O_X$? No.

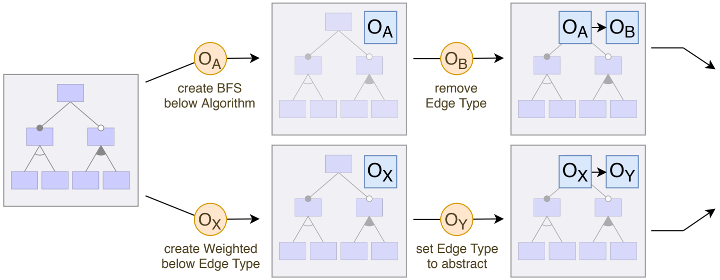
Feature Modeling Conflicts



$O_A \otimes O_X$? **No.**

$O_A \otimes O_Y$, based on O_X?

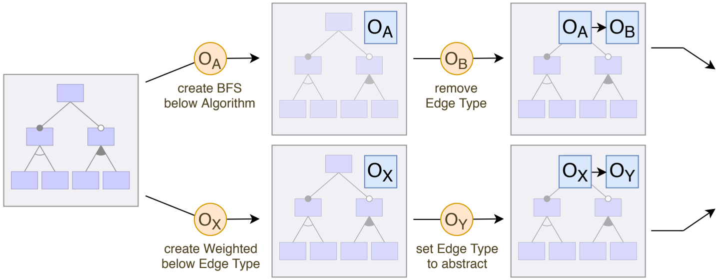
Feature Modeling Conflicts



$O_A \otimes O_X$? **No.**

$O_A \otimes O_Y$, based on O_X? **No.**

Feature Modeling Conflicts

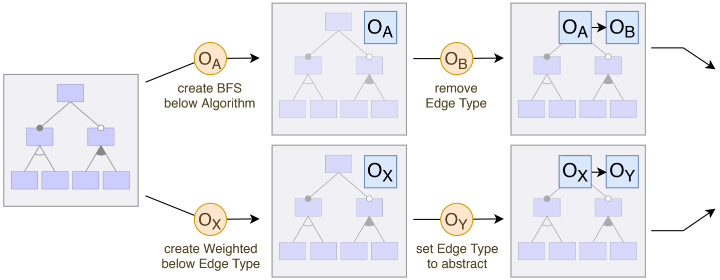


$O_A \otimes O_X$? **No.**

$O_A \otimes O_Y$, based on O_X? **No.**

$O_X \otimes O_B$, based on O_A?

Feature Modeling Conflicts

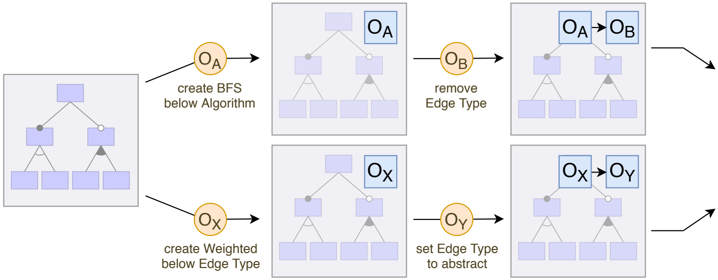


$O_A \otimes O_X$? **No.**

$O_A \otimes O_Y$, based on O_X? **No.**

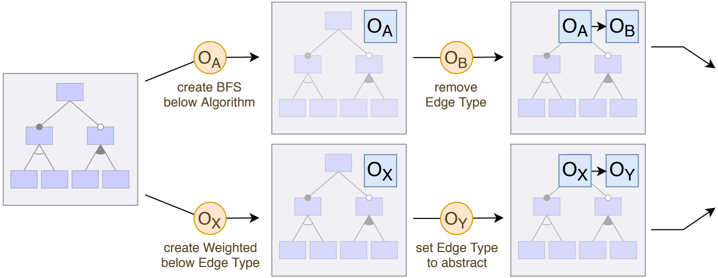
$O_X \otimes O_B$, based on O_A? **Yes.**

Feature Modeling Conflicts



$O_A \otimes O_X$? **No.**
$O_A \otimes O_Y$, based on O_X? **No.**
$O_X \otimes O_B$, based on O_A? **Yes.**
Thus, also $O_Y \otimes O_B$.

Feature Modeling Conflicts



$O_A \otimes O_X$? **No.**

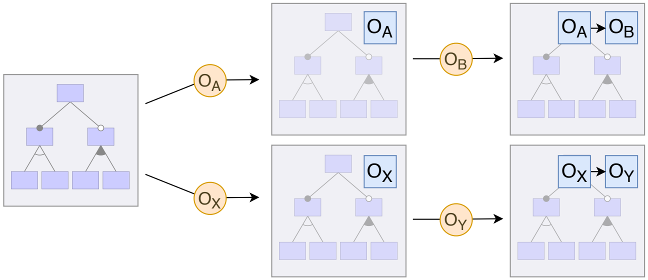
$O_A \otimes O_Y$, based on O_X? **No.**

$O_X \otimes O_B$, based on O_A? **Yes.**

Thus, also $O_Y \otimes O_B$.

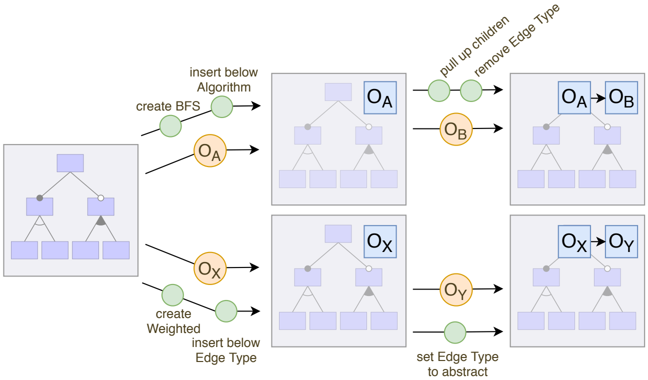
$\{\{O_A, O_B\}, \{O_A, O_X, O_Y\}\}$

Feature Modeling Conflicts



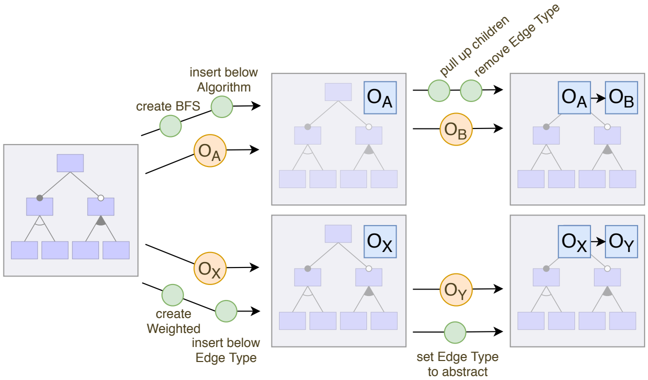
Decompose into Primitive Operations.

Feature Modeling Conflicts



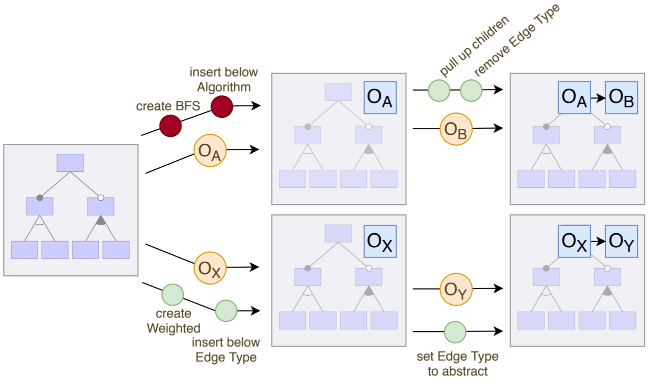
Decompose into Primitive Operations.

Feature Modeling Conflicts



Decompose into Primitive Operations.
$O_X \otimes O_B$ because:

Feature Modeling Conflicts

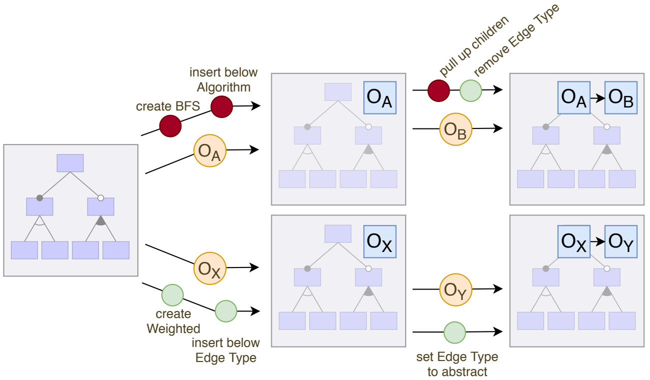


Decompose into Primitive Operations.

$O_X \otimes O_B$ because:

Based on O_A,

Feature Modeling Conflicts

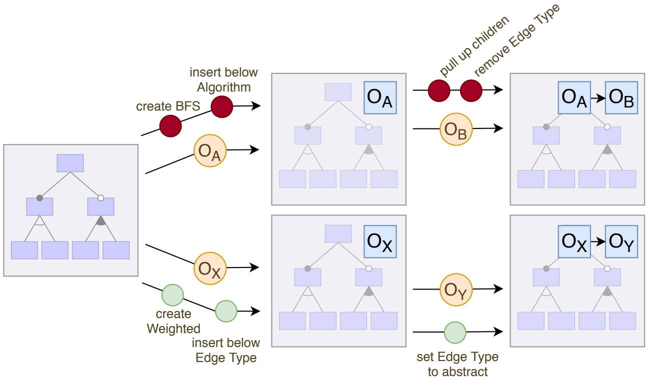


Decompose into Primitive Operations.

$O_X \otimes O_B$ because:

Based on O_A, apply O_B.

Feature Modeling Conflicts

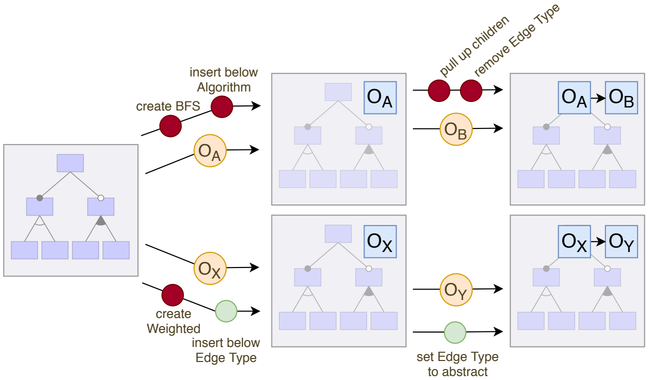


Decompose into Primitive Operations.

$O_X \otimes O_B$ because:

Based on O_A, apply O_B.

Feature Modeling Conflicts



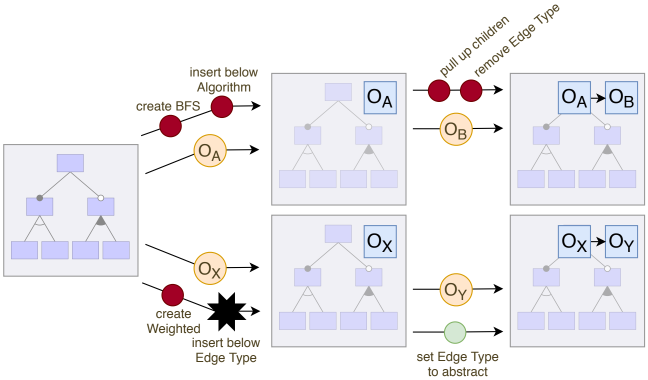
Decompose into Primitive Operations.

$O_X \otimes O_B$ because:

Based on O_A, apply O_B.

Now apply O_X ...

Feature Modeling Conflicts



Decompose into Primitive Operations.

$O_X \otimes O_B$ because:

Based on O_A, apply O_B.

Now apply O_X ...

...but a conflict rule applies.

Conflict Detection Rules

- No writes to removed features.

Conflict Detection Rules

- No writes to removed features.
- No writes to the same feature attribute.

Conflict Detection Rules

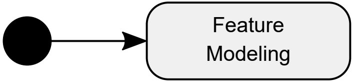
- No writes to removed features.
- No writes to the same feature attribute.
- May not introduce cycles.

Conflict Detection Rules

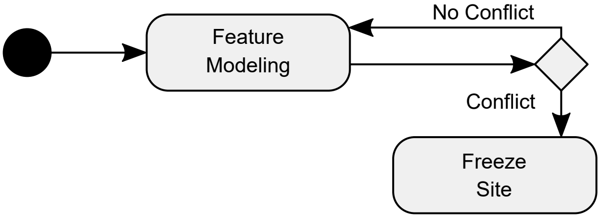
- No writes to removed features.
- No writes to the same feature attribute.
- May not introduce cycles.
- No mandatory/optional writes to group children.

Conflict Detection Rules

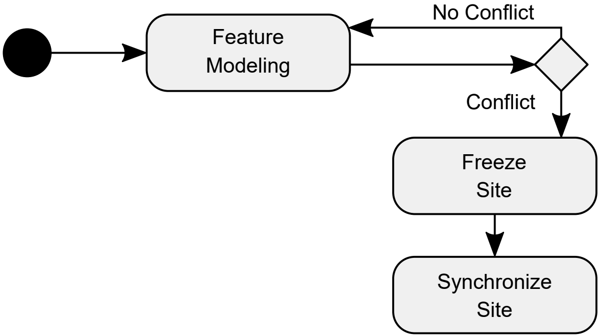
- No writes to removed features.
- No writes to the same feature attribute.
- May not introduce cycles.
- No mandatory/optional writes to group children.
- ... (cross-tree constraints, semantic properties)


Feature
Modeling

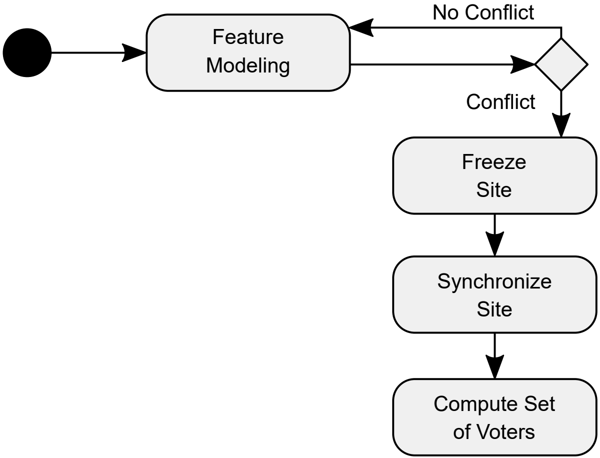
Resolution Process



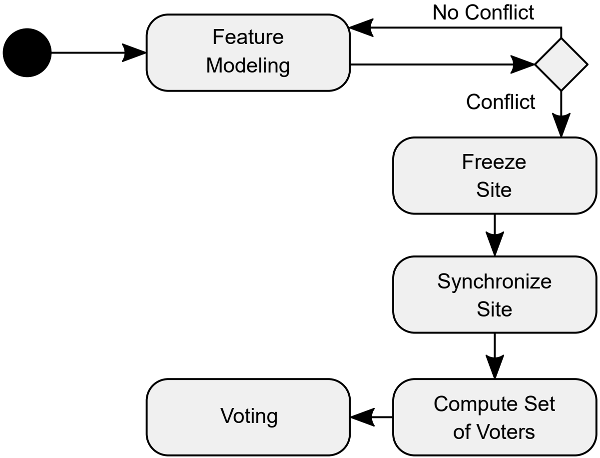
Resolution Process



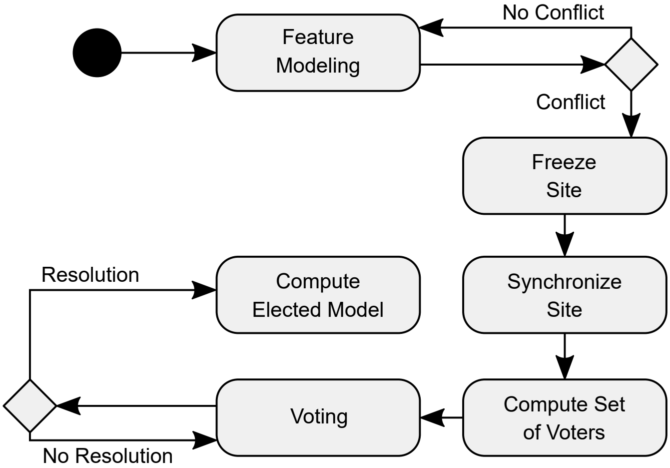
Resolution Process



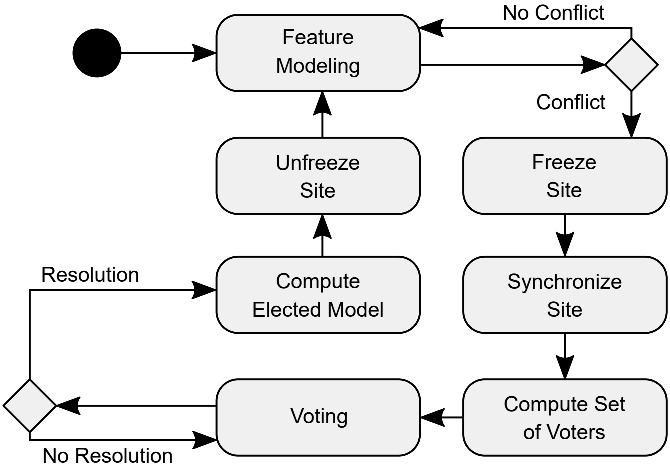
Resolution Process



Resolution Process



Resolution Process



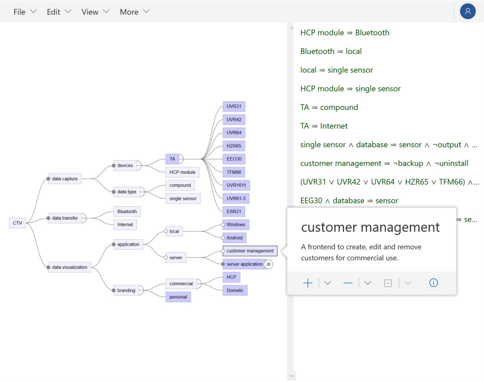
Correctness

- Convergence
- Causality Preservation
- Intention Preservation

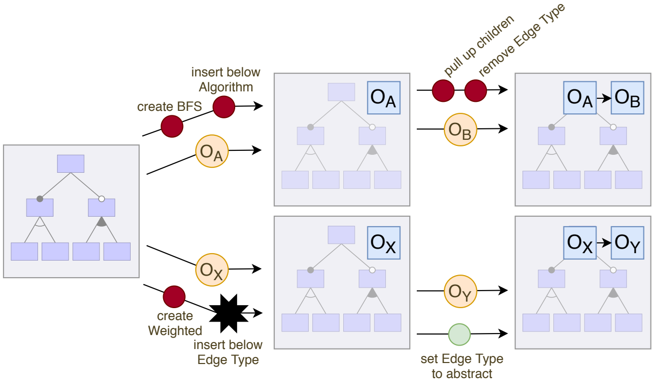
Correctness

- Convergence
- Causality Preservation
- Intention Preservation

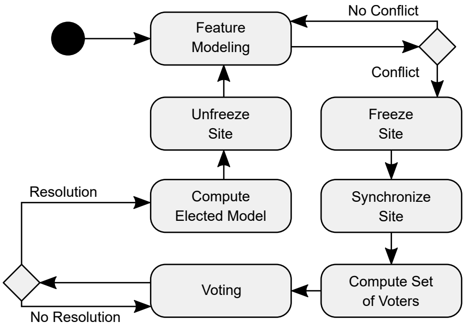
Prototype + Tests



Conclusion

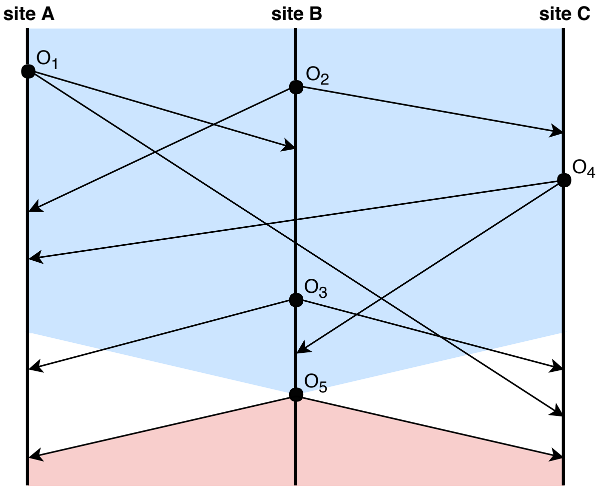


Conflict Detection



Conflict Resolution

CCI Model



- Convergence
- Causality Preservation
- Intention Preservation

Concurrency Control

	Turn-Taking	Locking	CRDTs	Serialization	OT	MVSD	MVMD
Concurrency	○	◐	●	●	●	●	●
Optimism	◐	○	●	●	●	●	●
Intention Preservation	●	○	○	○	○	◐	●
Flexibility	●	○	○	○	○	○	◐
Correctness	●	◐	◐	●	○	○	◐

Architecture (Demo)