

### 1 UNIFIED CONCEPTUAL MODEL

The unified conceptual model (Figure 1) describes essential concepts for modeling variability of a software system in space (variants) and time (revisions). It follows an open-world assumption (descriptive) instead of a closed-world assumption (prescriptive).

In Table 1, we provide a definition of the involved concepts.

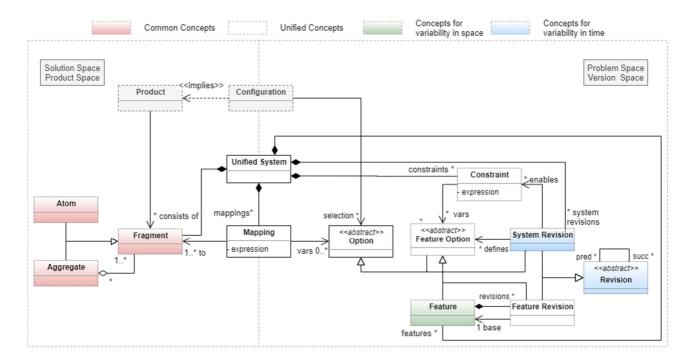


Figure 1: The Conceptual Model with common and unified Concepts for Variability in Space and Time.

Table 1: Definition of concepts in the Conceptual Model.

Concept	Relation to other	Definition	
	Concepts		
Fragment	Product, Unified	Fragments are the essential concept to describe a system on	
	System, Mapping	implementation level. A Fragment can either be an atom or	
		an aggregate, e.g. a single file, character or the node of an	
		AST. We explicitly do not specify the level of granularity for	
		an atom or aggregate to remain as generic as possible. A	
		hierarchical structure of containments is not enforced.	
		Instead, Fragments can be composed to various	
		combinations.	

Product	Configuration,	A <i>Product</i> is implied by a configuration. A <i>Product</i> is not part
	(consists of *)	of the system's state but can be computed from it based on
	Fragment	the configuration.
Unified System	(Contains *) Fragment,	The <i>Unified System</i> represents the unified configurable
_	Mapping,	space regarding spatial and temporal variability. It subsumes
	Configuration,	concepts from both solution and problem space.
	Constraint, Feature,	
	System Revision	
Mapping	Unified System, (has *)	A <i>Mapping</i> is an arbitrary expression (e.g., Boolean formula)
	Option variables,	that consists of <i>Option</i> variables that are mapped to
	(references 1*)	fragments. Therefore, the Mapping connects concepts from
	Fragment	the solution space (fragments) to concepts in the problem
		space (options).
Option	Configuration,	An <i>Option</i> expresses the variability of a system. This can
	Mapping, Feature	either manifest as variability in space (i.e., <i>Feature</i> ) or
	Option, System	variability in time (i.e. System Revision or Feature Revision).
	Revision	
Feature Option	(Extends) Option,	A Feature Option represents the configurable space on
	Constraint, System	feature level.
	Revision, Feature,	
	Feature Revision	
Feature	(Contains *) Feature	" A prominent or distinctive user-visible aspect, quality, or
	Revision	characteristic of a software system or systems [1]"
Revision	(Has *) predecessor	A Revision evolves along the time dimension and is intended
	and successor	to supersede its predecessor by an increment, e.g., due to a
	Revision	bug fix or refactoring. This relation forms a revision graph,
		which is a directed acyclic graph (DAG) with each node
		representing a unique revision.
System	(Extends) Revision,	A System Revision extends the Revision and represents the
Revision	(defines *) Feature	evolutionary state of the entire system at one point in time.
	Option, (enables *)	This state involves the definition of Features and Feature
	Constraint	Revisions (e.g., System Revision 2 involves feature A in
		revision 1 and Feature B in revision 2) along with Constraints
		that are valid for the respective System Revision.
Feature	(has 1 base) Feature,	A Feature Revision extends the Revision and represents an
Revision	(extends) Feature	evolutionary state of one particular <i>Feature</i> at one point in
	Option, (extends)	time.
	Sparsin, (externes)	""" - "

Configuration	(Has a selection of *)	A Configuration implies one particular Product of the Unified
	Options, implies	System and consists of a selection of Option variables. It is
	Product	not part of the system's state.
Constraint	Unified System,	The Constraint is an arbitrary expression (e.g., Boolean
	System Revision, (has	formula) that constrains <i>Feature Options</i> that can be
	*) Feature Option	combined in a Configuration.

# 2 MAPPING

To assess the mapping between concepts and relations of the unified conceptual model regarding the selected tool, each concept and relation is considered separately. For the sake of simplicity, we omit inheritance relationships.

## 2.1 CONCEPTS

For each concept of the conceptual model listed in Table 3, please inspect whether an equivalent construct exists in your tool and complete the form according to the following scheme in Table 2:

Table 2: Exemplary Mapping of ECCO (incomplete).

Concept in	Maps to Construct	Does not map /	Please comment, if concept is only
Model	(Name)	Does not exist	partially reflected
Fragment	Artifact	-	-
Product	-	✓	Because it is not part of the state of the
			system but exists as output in the form
			of files in the file system.
System Revision	-	✓	ECCO considers Feature Revisions
			only.

Table 3: Concept Mapping between Conceptual Model and Tool.

Concept in	Maps to Construct	Does not map /	Please comment, if concept is only
Model	(Name)	does not exist	partially reflected
Fragment	Artifact		
Product	Variant		
Unified System	Repository		
Mapping	Association		
Option (abstract)			

Feature Option		
(abstract)		
Feature	Feature	
Revision	Revision	
(abstract)		
System Revision	-	
Feature Revision	Feature Revision	
Configuration	Configuration	
Constraint	-	
Unmapped		
constructs in tool		
Remarks		

# 2.2 RELATIONS

For each relation of the conceptual model listed in Table 5, please inspect whether an equivalent relation exists in your tool and complete the form according to the following scheme in Table 4:

Table 4: Exemplary Mapping of ECCO (incomplete).

Name of	Maps to Relation	Does not map /	If relation is only partially mapped,
Relation in		Does not exist	please name divergence (source,
Conceptual			target, multiplicity, direction and kind)
Model			
Graph-based	Tree-based	-	Uses strong containment instead of weak
Fragment	Fragment		containment for children of fragments. To
structure	structure with		mitigate this limitation, ECCO uses cross-
	cross-tree		tree references.
	references		
	(subsumed by		
	graph)		
Mapping has 1*	equivalent	-	
Fragments			
System Revision	-	✓	ECCO considers Feature Revisions only.
defines * Feature			
Options			

Table 5: Relation Mapping between Conceptual Model and Tool.

Name of Relation in	Maps to	Does not map /	If relation is only partially mapped,
Conceptual Model	Relation	Does not exist	please name divergence (source, target,
			multiplicity, direction and kind)
Graph-based	Tree-based		
Fragment structure	Fragment		
	structure with		
	cross-tree		
	references		
	(subsumed		
	by graph)		

Fragments  Mapping has 1.* Fragments  Configuration implies Product  Configuration has a selection of * Options  Unified System has * fragments  Unified System has * Constraints  Constraint has * Feature Poption variables  System Revision defines * Feature Options  System Revision enables * Constraints  Revision has * successor (Branching/Forking) and predecessor (Merging) Revisions  Unmapped relations in tool	Product consists of *	equivalent		
Fragments  Configuration implies Product  Configuration has a selection of * Options  Unified System has * equivalent fragments  ECCO considers Feature Revisions only.  In the sequivalent fragments fragme	Fragments			
Configuration implies Product  Configuration has a selection of *Options  Unified System has * equivalent fragments  Unified System has * equivalent Mappings  Unified System has * equivalent  Unified System has * System Revisions  Unified System has * equivalent  Eatures  Unified System has * equivalent  Features  Unified System has * equivalent  Features  Unified System has * equivalent  Feature As * Feature  Revisions  Constraint has * equivalent  Feature Option  variables  System Revision  defines * Feature  Options  System Revision  enables * Constraints  Y ECCO considers Feature Revisions only.  Becco considers Feature Revisions only.  ECCO considers Feature Revisions only.  Becco considers Feature Revisions only.  And DAG supported.  No DAG supported.  Unmapped relations in	Mapping has 1*	equivalent		
Product  Configuration has a selection of *Options  Unified System has * equivalent fragments  Unified System has * equivalent Mappings  Unified System has * constraints  Unified System has * equivalent  Unified System has * equivalent  Features  Unified System has * equivalent  Features  Unified System has * system Revisions  Mapping has *Option variables  Feature Apa * Feature  Constraint has * equivalent  Feature Option  variables  System Revision  System Revision  Gonstraint has * equivalent  Feature Option  variables  System Revision  Gonstraint has * equivalent  Feature Option  variables  System Revision  Gonstraint has * equivalent  Feature Option  variables  System Revision  Gonstraint has * equivalent  Feature Option  variables  System Revision  Gonstraint has * equivalent  Feature Option  variables  No DAG supported.  Unimapped relations in	Fragments			
Configuration has a selection of * Options  Unified System has * equivalent fragments  Unified System has * equivalent Mappings  Unified System has * constraints  Unified System has * equivalent Mappings  Unified System has * equivalent Features  Unified System has * equivalent Features  Unified System has * equivalent Features  Unified System has * equivalent Feature Revisions  Mapping has * Option variables  Feature has * Feature Revisions  Constraint has * equivalent Feature Option variables  System Revision  Goffines * Feature Option variables  System Revision  edfines * Feature Option variables  System Revision  edfines * Feature Option variables  System Revision enables * Constraints  Revision has * successor  (Branching/Forking) and predecessor  (Merging) Revisions  Unmapped relations in	Configuration <i>implies</i>	equivalent		
Selection of * Options Unified System has * equivalent fragments Unified System has * equivalent Mappings Unified System has * constraints Unified System has * equivalent Unified System has * equivalent Features Unified System has * equivalent Features Unified System has * equivalent Feature Revisions Mapping has * Option variables Feature has * Feature Revisions Constraint has * equivalent Revisions Constraint has * equivalent Feature Option variables System Revision System Revision  ### Constraint has * equivalent Feature Option  ### Constraint has * equivalent Feature Option  ### System Revision ### Constraints Revision has * successor (Branching/Forking) and predecessor (Merging) Revisions Unmapped relations in	Product			
Unified System has * fragments	Configuration has a	equivalent		
fragments  Unified System has * equivalent  Unified System has * Constraints  Unified System has * equivalent  Unified System has * Features  Unified System has * System Revisions  Mapping has * Option variables  Feature has * Feature Revisions  Constraint has * equivalent  Revisions  Constraint has * equivalent  Feature Option variables  System Revision  defines * Feature Options  System Revision  enables * Constraints  Revision has * successor (Branching/Forking) and predecessor (Merging) Revisions  Unified System has * equivalent  ECCO considers Feature Revisions only.  ECCO considers Feature Revisions only.  ECCO considers Feature Revisions only.  A Constraint has * ECCO considers Feature Revisions only.  ECCO considers Feature Revisions only.  A Constraint has * Successor (Branching/Forking) and predecessor (Merging) Revisions  Unmapped relations in	selection of * Options			
Unified System has * dequivalent Mappings Unified System has * Constraints Unified System has * Constraints Unified System has * equivalent Features Unified System has * System Revisions Mapping has * Option variables Feature has * Feature Revisions Constraint has * equivalent Revisions  System Revision  Constraint has * equivalent Revisions System Revision  defines * Feature Option System Revision  System Revision  defines * Constraints  Revision has * successor (Branching/Forking) and predecessor (Merging) Revisions Unmapped relations in	Unified System has *	equivalent		
Mappings Unified System has * Constraints Unified System has * Features Unified System has * System Revisions Mapping has *Option variables Feature Option variables System Revision  Constraint has * Feature Option variables  System Revision  System Revision  Constraint has * Feature Option variables  System Revision  System Revision  Gerivalent  Feature Option variables  System Revision  Gerivalent  Feature Options  System Revision  Gerivalent  Fecco considers Feature Revisions only.  ECCO considers Feature Revisions only.  ECCO considers Feature Revisions only.  ECCO considers Feature Revisions only.  Fecco considers Feature Revisions only.	fragments			
Unified System has * Constraints  Unified System has * Features  Unified System has * Features  Unified System has * System Revisions  Mapping has *Option variables  Feature has * Feature Revisions  Constraint has * Feature Option variables  System Revision  Gystem Revision  Constraint has * Feature Option variables  System Revision  System Revision  Gystem Revision  Gystem Revision  Feature Options  System Revision  Gystem Revision  Feature Options  System Revision  Feature Options  System Revision  Gystem Revision  Feature Options  System Revision  Feature Options  System Revision  Gystem Revision  Feature Revisions only.  Feature Revisions only.  Feature Options  System Revision  Feature Revisions only.  Feat	Unified System has *	equivalent		
Constraints  Unified System has * Features  Unified System has * Feature Revisions  Mapping has *Option variables  Feature Revisions  Constraint has * Feature Option variables  System Revision  Gostraint has * Feature Option  System Revision  Gostraint has * System Revision ha	Mappings			
Unified System has * Features  Unified System has * System Revisions  Mapping has *Option variables  Feature has * Feature Revisions  Constraint has * Feature Option variables  System Revision  System Revision  Constraint has * Feature Option variables  System Revision  defines * Feature Options  System Revision  Revision has * System Revision  defines * Constraints  Fecco considers Feature Revisions only.  ECCO considers Feature Revisions only.  ECCO considers Feature Revisions only.  No DAG supported.  No DAG supported.  Unmapped relations in	Unified System has *			
Features  Unified System has * System Revisions  Mapping has * Option variables  Feature has * Feature Revisions  Constraint has * Feature Option variables  System Revision  System Revision  Gotton tariables  Feature Option variables  System Revision  defines * Feature Options  System Revision enables * Constraints  Revision has * successor (Branching/Forking) and predecessor (Merging) Revisions  Unmapped relations in	Constraints			
Unified System has * System Revisions  Mapping has *Option variables  Feature has *Feature Revisions  Constraint has * Feature Option variables  System Revision  defines * Feature Options  System Revision  ECCO considers Feature Revisions only.  BECCO considers Feature Revisions only.  ECCO considers Feature Revisions only.  ECCO considers Feature Revisions only.  BECCO considers Feature Revisions only.  ECCO considers Feature Revisions only.  BECCO considers Feature Revisions only.  ECCO considers Feature Revisions only.  BECCO considers Feature Revisions only.  BECCO considers Feature Revisions only.  ECCO considers Feature Revisions only.  BECCO considers Feature Revisions only.	Unified System has *	equivalent		
System Revisions  Mapping has * Option variables  Feature has * Feature Revisions  Constraint has * Feature Option variables  System Revision  System Revision  System Revision  Gotions  Feature Revisions only.	Features			
Mapping has * Option variables  Feature has * Feature Revisions  Constraint has * equivalent  Feature Option variables  System Revision defines * Feature Options  System Revision enables * Constraints  Revision has * successor (Branching/Forking) and predecessor (Merging) Revisions  Unmapped relations in	Unified System has *		<b>√</b>	ECCO considers Feature Revisions only.
Feature has * Feature Revisions  Constraint has * Feature Option variables  System Revision defines * Feature Options  System Revision enables * Constraints  Revision has * successor (Branching/Forking) and predecessor (Merging) Revisions  Feature equivalent  ECCO considers Feature Revisions only.  ECCO considers Feature Revisions only.  No DAG supported.	System Revisions			
Feature has * Feature Revisions  Constraint has * Feature Option variables  System Revision defines * Feature Options  System Revision enables * Constraints  Revision has * successor (Branching/Forking) and predecessor (Merging) Revisions  Unmapped relations in	Mapping has * Option	equivalent		
Revisions  Constraint has * Feature Option variables  System Revision defines * Feature Options  System Revision enables * Constraints  Revision has * successor (Branching/Forking) and predecessor (Merging) Revisions Unmapped relations in	variables			
Constraint has * Feature Option variables  System Revision defines * Feature Options  System Revision enables * Constraints  Revision has * successor (Branching/Forking) and predecessor (Merging) Revisions  Unmapped relations in	Feature <i>has</i> * Feature	equivalent		
Feature Option variables  System Revision  defines * Feature Options  System Revision enables * Constraints  Revision has * successor (Branching/Forking) and predecessor (Merging) Revisions  Unmapped relations in	Revisions			
variables       System Revision         defines * Feature       ✓         Options       ECCO considers Feature Revisions only.         System Revision       ✓         enables * Constraints       ECCO considers Feature Revisions only.         Revision has *       ✓         successor       No DAG supported.         (Branching/Forking)       and predecessor         (Merging) Revisions       Unmapped relations in	Constraint has *	equivalent		
System Revision  defines * Feature Options  System Revision enables * Constraints  Revision has * successor (Branching/Forking) and predecessor (Merging) Revisions  Unmapped relations in	Feature Option			
defines * Feature       Options         System Revision       ✓         enables * Constraints       ECCO considers Feature Revisions only.         Revision has *       ✓         successor       No DAG supported.         (Branching/Forking)       and predecessor         (Merging) Revisions       Unmapped relations in	variables			
Options  System Revision enables * Constraints  Revision has * successor (Branching/Forking) and predecessor (Merging) Revisions  Unmapped relations in	System Revision		<b>✓</b>	ECCO considers Feature Revisions only.
System Revision  enables * Constraints  Revision has * successor (Branching/Forking) and predecessor (Merging) Revisions  Unmapped relations in	defines * Feature			
enables * Constraints  Revision has * successor (Branching/Forking) and predecessor (Merging) Revisions  Unmapped relations in	Options			
Revision has * successor (Branching/Forking) and predecessor (Merging) Revisions Unmapped relations in	System Revision		<b>✓</b>	ECCO considers Feature Revisions only.
successor (Branching/Forking) and predecessor (Merging) Revisions Unmapped relations in	enables * Constraints			
(Branching/Forking) and predecessor (Merging) Revisions Unmapped relations in	Revision has *		<b>✓</b>	No DAG supported.
and predecessor (Merging) Revisions Unmapped relations in	successor			
(Merging) Revisions Unmapped relations in	(Branching/Forking)			
Unmapped relations in	and predecessor			
	(Merging) Revisions			
tool	Unmapped relations in			1
	tool			

Remarks	

#### A. REFERENCES

- [1] K. Kang, J. Hess W. Novak, and A. Peterson, "Feature-Oriented Domain Analysis (FODA) Feasibility Study.," Carnegie Mellon University, 1990.
- [2] G. Guizzardi, L. F. Pires and M. van Sinderen, "An Ontology-Based Approach for Evaluating the Domain Appropriateness and Comprehensibility Appropriateness of Modeling Languages," *Proceedings of the International Conference on Model Driven Engineering Languages and Systems*, 2005.
- [3] S. Ananieva, T. Kehrer, H. Klare, A. Koziolek, H. Lönn, S. Ramesh, A. Burger, G. Taentzer and B. Westfechtel, "Towards a conceptual model for unifying variability in space and time," *Proceedings of the 2nd International Workshop on Variability and Evolution of Software-Intensive Systems*, 2019.