

Q1. What is the most appropriate intent for each transformation?

Choose one intent per question only by dragging it inside the 'Intent' box.

If you are **REALLY** doubting between two intents, drag the most likely one in the 'Intent' box and the other in the 'Alternate Intent' box.

Q2. Convert a Class diagram to XMI in order to store the model on the cloud.

Input: Class diagram

Output: XMI schema schema

Items	Intent	Alternate Intent
Abstraction	Serialization 1	
Analysis		
Animation		
Approximation		
Canonicalization		
Migration		
Model Editing		
Model Finding		
Model Generation		
Model Matching		
Model Merging		
Model Refactoring		
Model Synchronization		
Normalization		
Optimization		
Parsing		
Refinement		
Rendering		
Restrictive Query		
Reverse Engineering		
Simulation		
Synthesis		
Translation		
Translational Semantics		

Q3. Flatten a Simulink model.

Input: Simulink model

Output: Simulink model

Items	Intent	Alternate Intent
Abstraction	Normalization 1	

Analysis		
Animation		
Approximation		
Canonicalization		
Migration		
Model Editing		
Model Finding		
Model Generation		
Model Matching		
Model Merging		
Model Refactoring		
Model Synchronization		
Optimization		
Parsing		
Refinement		
Rendering		
Restrictive Query		
Reverse Engineering		
Serialization		
Simulation		
Synthesis		
Translation		
Translational Semantics		

Q4. Transform a model defined in the Kernel Meta Meta Model (KM3) into a model defined in Ecore.

Input: KM3 model

Output: Ecore model

Items	Intent	Alternate Intent
Abstraction	Translation 1	
Analysis		
Animation		
Approximation		
Canonicalization		
Migration		
Model Editing		
Model Finding		
Model Generation		
Model Matching		
Model Merging		
Model Refactoring		
Model Synchronization		
Normalization		
Optimization		

- Parsing
- Refinement
- Rendering
- Restrictive Query
- Reverse Engineering
- Serialization
- Simulation
- Synthesis
- Translational Semantics

Q5. Given a state machine, generate a counter-example for a property.

Input: State machine

Output: Counter-example trace

Items	Intent	Alternate Intent
Abstraction	Model Finding 1	
Analysis		
Animation		
Approximation		
Canonicalization		
Migration		
Model Editing		
Model Generation		
Model Matching		
Model Merging		
Model Refactoring		
Model Synchronization		
Normalization		
Optimization		
Parsing		
Refinement		
Rendering		
Restrictive Query		
Reverse Engineering		
Serialization		
Simulation		
Synthesis		
Translation		
Translational Semantics		

Q6. Given a class diagram, introduce common super-classes for classes that share common attributes.

Input: Class diagram

Output: Class diagram

Items	Intent	Alternate Intent
Abstraction	Model Refactoring 1	
Analysis		
Animation		
Approximation		
Canonicalization		
Migration		
Model Editing		
Model Finding		
Model Generation		
Model Matching		
Model Merging		
Model Synchronization		
Normalization		
Optimization		
Parsing		
Refinement		
Rendering		
Restrictive Query		
Reverse Engineering		
Serialization		
Simulation		
Synthesis		
Translation		
Translational Semantics		

Q7. Generate the Java Doc from a class diagram.

Input: Class diagram

Output: The HTML documentation

Items	Intent	Alternate Intent
Abstraction	Synthesis 1	
Analysis		
Animation		
Approximation		
Canonicalization		
Migration		
Model Editing		
Model Finding		
Model Generation		
Model Matching		
Model Merging		
Model Refactoring		

Model Synchronization
Normalization
Optimization
Parsing
Refinement
Rendering
Restrictive Query
Reverse Engineering
Serialization
Simulation
Translation
Translational Semantics

Q8. Convert a relation model (of a relational database) into its unique third normal form (3NF).
Input: Entity-Relation (ER) model
Output: ER model

Items	Intent	Alternate Intent
Abstraction	Canonicalization 1	
Analysis		
Animation		
Approximation		
Migration		
Model Editing		
Model Finding		
Model Generation		
Model Matching		
Model Merging		
Model Refactoring		
Model Synchronization		
Normalization		
Optimization		
Parsing		
Refinement		
Rendering		
Restrictive Query		
Reverse Engineering		
Serialization		
Simulation		
Synthesis		
Translation		
Translational Semantics		

Q9. Define the actions that can be performed from a Pacman game state to move to its next state.

Input: Pacman configuration model
Output: Pacman configuration model

Items	Intent	Alternate Intent
Abstraction	Simulation 1	
Analysis		
Animation		
Approximation		
Canonicalization		
Migration		
Model Editing		
Model Finding		
Model Generation		
Model Matching		
Model Merging		
Model Refactoring		
Model Synchronization		
Normalization		
Optimization		
Parsing		
Refinement		
Rendering		
Restrictive Query		
Reverse Engineering		
Serialization		
Synthesis		
Translation		
Translational Semantics		

Q10. Given a DSM consisting of processes with some tasks initially assigned, produce an optimal assignment of the set of tasks to the set of processors.

Input: ProcessorTaskAssignment DSM
Output: ProcessorTaskAssignment DSM

Items	Intent	Alternate Intent
Abstraction	Optimization 1	
Analysis		
Animation		
Approximation		
Canonicalization		
Migration		
Model Editing		
Model Finding		
Model Generation		

Model Matching
Model Merging
Model Refactoring
Model Synchronization
Normalization
Parsing
Refinement
Rendering
Restrictive Query
Reverse Engineering
Serialization
Simulation
Synthesis
Translation
Translational Semantics

Q11. Augment a class diagram by adding navigability information on associations, role names, attribute types, methods return types, and method parameter types.

Input: Metamodel in UML class diagram
Output: Metamodel in UML class diagram

Items	Intent	Alternate Intent
Abstraction	Refinement	
Analysis		
Animation		
Approximation		
Canonicalization		
Migration		
Model Editing		
Model Finding		
Model Generation		
Model Matching		
Model Merging		
Model Refactoring		
Model Synchronization		
Normalization		
Optimization		
Parsing		
Rendering		
Restrictive Query		
Reverse Engineering		
Serialization		
Simulation		
Synthesis		

Translation
Translational Semantics

Q12. Check a temporal property on a state machine.

Input: State machine model

Output: Boolean value

Items	Intent	Alternate Intent
Abstraction	Analysis 1	
Animation		
Approximation		
Canonicalization		
Migration		
Model Editing		
Model Finding		
Model Generation		
Model Matching		
Model Merging		
Model Refactoring		
Model Synchronization		
Normalization		
Optimization		
Parsing		
Refinement		
Rendering		
Restrictive Query		
Reverse Engineering		
Serialization		
Simulation		
Synthesis		
Translation		
Translational Semantics		

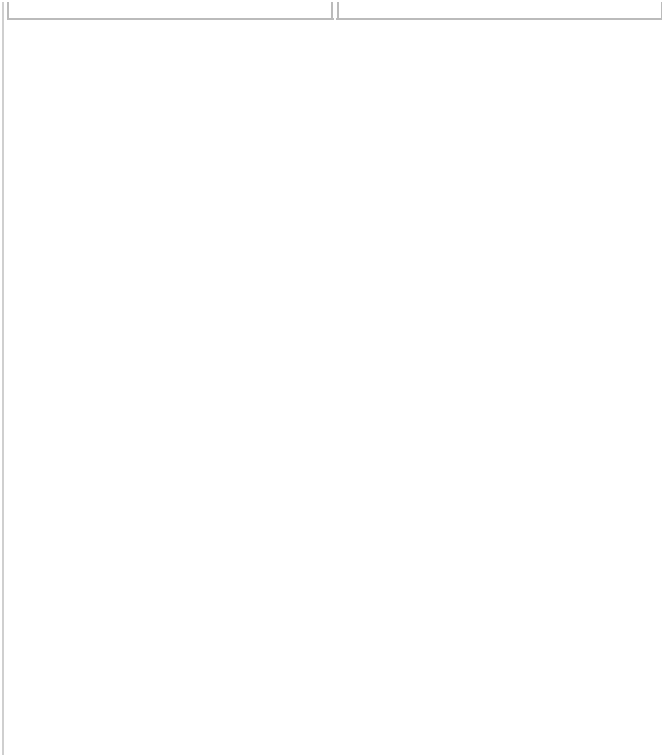
Q13. Extract the classes with no super-classes from a class diagram.

Input: Class diagram

Output: Class diagram

Items	Intent	Alternate Intent
Abstraction	Restrictive Query 1	
Analysis		
Animation		
Approximation		
Canonicalization		

- Migration
- Model Editing
- Model Finding
- Model Generation
- Model Matching
- Model Merging
- Model Refactoring
- Model Synchronization
- Normalization
- Optimization
- Parsing
- Refinement
- Rendering
- Reverse Engineering
- Serialization
- Simulation
- Synthesis
- Translation
- Translational Semantics



Q14. Extract the class hierarchy from a class diagram (with single inheritance) in the form of a directed tree.

Input: Class diagram
Output: Tree

Items	Intent	Alternate Intent
Analysis	Abstraction 1	
Animation		
Approximation		
Canonicalization		
Migration		
Model Editing		
Model Finding		
Model Generation		
Model Matching		
Model Merging		
Model Refactoring		
Model Synchronization		
Normalization		
Optimization		
Parsing		
Refinement		
Rendering		
Restrictive Query		
Reverse Engineering		

Serialization
Simulation
Synthesis
Translation
Translational Semantics

Q15. Clean up a UML class diagram by removing out-of-date annotations on all diagram elements.

Input: Class diagram

Output: Class diagram

Items	Intent	Alternate Intent
Abstraction	Model Editing 1	
Analysis		
Animation		
Approximation		
Canonicalization		
Migration		
Model Finding		
Model Generation		
Model Matching		
Model Merging		
Model Refactoring		
Model Synchronization		
Normalization		
Optimization		
Parsing		
Refinement		
Rendering		
Restrictive Query		
Reverse Engineering		
Serialization		
Simulation		
Synthesis		
Translation		
Translational Semantics		

Q16. Generate a sequence diagram from Python code.

Input: Python program

Output: Sequence diagram

Items	Intent	Alternate Intent
Abstraction	Reverse Engineering 1	
Analysis		

Animation		
Approximation		
Canonicalization		
Migration		
Model Editing		
Model Finding		
Model Generation		
Model Matching		
Model Merging		
Model Refactoring		
Model Synchronization		
Normalization		
Optimization		
Parsing		
Refinement		
Rendering		
Restrictive Query		
Serialization		
Simulation		
Synthesis		
Translation		
Translational Semantics		

Q17. Map a custom DSML for wrist watches to a Statechart model in order to define its behavior.
Input: Watch DSML
Output: Statechart

Items	Intent	Alternate Intent
Abstraction	Translational Semantics 1	
Analysis		
Animation		
Approximation		
Canonicalization		
Migration		
Model Editing		
Model Finding		
Model Generation		
Model Matching		
Model Merging		
Model Refactoring		
Model Synchronization		
Normalization		
Optimization		
Parsing		

- Refinement
- Rendering
- Restrictive Query
- Reverse Engineering
- Serialization
- Simulation
- Synthesis
- Translation

Q18. Move dots representing vehicles through a map of a city.

Input: City traffic DSML

Output: City traffic DSML

Items	Intent	Alternate Intent
Abstraction	Animation 1	
Analysis		
Approximation		
Canonicalization		
Migration		
Model Editing		
Model Finding		
Model Generation		
Model Matching		
Model Merging		
Model Refactoring		
Model Synchronization		
Normalization		
Optimization		
Parsing		
Refinement		
Rendering		
Restrictive Query		
Reverse Engineering		
Serialization		
Simulation		
Synthesis		
Translation		
Translational Semantics		

Q19. Given a Petri Net model that does not satisfy its well-formedness constraints, generate a repair to the model that does satisfy the constraints.

Input: Petri Net

Output: Petri Net

Items	Intent	Alternate Intent
Abstraction	Model Finding	
Analysis		
Animation		
Approximation		
Canonicalization		
Migration		
Model Editing		
Model Generation		
Model Matching		
Model Merging		
Model Refactoring		
Model Synchronization		
Normalization		
Optimization		
Parsing		
Refinement		
Rendering		
Restrictive Query		
Reverse Engineering		
Serialization		
Simulation		
Synthesis		
Translation		
Translational Semantics		

Q20. Convert a class diagram to a human usable textual notation (HUTN) version of it.
Input: Class diagram
Output: HUTN

Items	Intent	Alternate Intent
Abstraction	Rendering	
Analysis		
Animation		
Approximation		
Canonicalization		
Migration		
Model Editing		
Model Finding		
Model Generation		
Model Matching		
Model Merging		
Model Refactoring		
Model Synchronization		

- Normalization
- Optimization
- Parsing
- Refinement
- Restrictive Query
- Reverse Engineering
- Serialization
- Simulation
- Synthesis
- Translation
- Translational Semantics

Q21. Generate a set of mutations of a sequence diagram used for mutation testing.

Input: Sequence diagram metamodel, sequence diagram

Output: Set of sequence diagrams

Items	Intent	Alternate Intent
Abstraction	Model Generation 1	
Analysis		
Animation		
Approximation		
Canonicalization		
Migration		
Model Editing		
Model Finding		
Model Matching		
Model Merging		
Model Refactoring		
Model Synchronization		
Normalization		
Optimization		
Parsing		
Refinement		
Rendering		
Restrictive Query		
Reverse Engineering		
Serialization		
Simulation		
Synthesis		
Translation		
Translational Semantics		

Q22. Upgrade an existing UML model in version 2.0 to the UML version 2.6.

Input: UML model v2.0
Output: UML model v2.6

Items	Intent	Alternate Intent
Abstraction	Migration 1	
Analysis		
Animation		
Approximation		
Canonicalization		
Model Editing		
Model Finding		
Model Generation		
Model Matching		
Model Merging		
Model Refactoring		
Model Synchronization		
Normalization		
Optimization		
Parsing		
Refinement		
Rendering		
Restrictive Query		
Reverse Engineering		
Serialization		
Simulation		
Synthesis		
Translation		
Translational Semantics		

Q23. Combine the Simulink models from different teams working on different parts of the system.
Input: set of Simulink models
Output: Simulink model

Items	Intent	Alternate Intent
Abstraction	Model Merging 1	
Analysis		
Animation		
Approximation		
Canonicalization		
Migration		
Model Editing		
Model Finding		
Model Generation		
Model Matching		
Model Refactoring		

- Model Synchronization
- Normalization
- Optimization
- Parsing
- Refinement
- Rendering
- Restrictive Query
- Reverse Engineering
- Serialization
- Simulation
- Synthesis
- Translation
- Translational Semantics

Q24. Assume two wrist watch companies each have their own design DSMLs. Automatically map the most likely corresponding metamodel elements of the two DSMLs.

Input: set of wrist watch DSMLs

Output: Mapping correspondences

Items	Intent	Alternate Intent
Abstraction	Model Matching 1	
Analysis		
Animation		
Approximation		
Canonicalization		
Migration		
Model Editing		
Model Finding		
Model Generation		
Model Merging		
Model Refactoring		
Model Synchronization		
Normalization		
Optimization		
Parsing		
Refinement		
Rendering		
Restrictive Query		
Reverse Engineering		
Serialization		
Simulation		
Synthesis		
Translation		
Translational Semantics		

Q25. Update the class diagram of A team according to the changes made by another team B.
Input: Class diagram A, Class diagram B
Output: Class diagram A, Class diagram B

Items	Intent	Alternate Intent
Abstraction	Model Synchronization 1	
Analysis		
Animation		
Approximation		
Canonicalization		
Migration		
Model Editing		
Model Finding		
Model Generation		
Model Matching		
Model Merging		
Model Refactoring		
Normalization		
Optimization		
Parsing		
Refinement		
Rendering		
Restrictive Query		
Reverse Engineering		
Serialization		
Simulation		
Synthesis		
Translation		
Translational Semantics		

Q26. Given a class diagram, create another class diagram as similar to it as possible but that does not use multiple inheritance.
Input: Class diagram
Output: Class diagram

Items	Intent	Alternate Intent
Abstraction	Approximation 1	
Analysis		
Animation		
Canonicalization		
Migration		
Model Editing		
Model Finding		
Model Generation		

Model Matching
Model Merging
Model Refactoring
Model Synchronization
Normalization
Optimization
Parsing
Refinement
Rendering
Restrictive Query
Reverse Engineering
Serialization
Simulation
Synthesis
Translation
Translational Semantics

Q27. Deduce the feature diagram model of a wristwatch DSML in textual notation to represent its essential features.

Input: Wrist watch textual model

Output: Feature diagram

Items	Intent	Alternate Intent
Abstraction	Parsing 1	
Analysis		
Animation		
Approximation		
Canonicalization		
Migration		
Model Editing		
Model Finding		
Model Generation		
Model Matching		
Model Merging		
Model Refactoring		
Model Synchronization		
Normalization		
Optimization		
Refinement		
Rendering		
Restrictive Query		
Reverse Engineering		
Serialization		
Simulation		

Synthesis
Translation
Translational Semantics

Q28. Is the catalog self explanatory?

Yes.


Q29. Are there other intents you can think of that are not covered by the catalog?

Q30. Do you have any general remarks on the catalog?

Location Data

Location: [\(33.307205200195, -87.585899353027\)](#)

Source: GeoIP Estimation



The map displays the southeastern United States, specifically Alabama and Georgia. Birmingham, Alabama, is highlighted with a red location pin. Other cities shown include Chattanooga, Huntsville, Marietta, Atlanta, Macon, Columbus, Montgomery, and Jackson. The map also shows the Mississippi River and the state boundaries between Mississippi, Alabama, and Georgia. The Google logo and 'Map data ©2014 Google' are visible at the bottom left and right of the map respectively.