

Mit jelentenek az alábbi fogalmak? (Nem a formális definíciók visszaadását kérem, hanem a fogalmak lényegét és értelmezését pl. a házi feladat kontextusában)

- 1) Domain-specific languages, metamodeling, EMF
  - a) Metamodel vs instance model
  - b) Metamodeling levels, multi-level metamodeling, deep instantiation
  - c) Concrete syntax vs. abstract syntax
  - d) Well-formedness constraints
  - e) Derived features and Views
  - f) Static vs. dynamic (behavioural) modeling languages
  - g) Operational vs. denotational semantics (dynamic)
  - h) Generative vs. interpreted modeling (e.g. architecture of EMF vs. Sirius)
  - i) Core concepts in Ecore metamodels (EClass, EReference, EAttribute)
- 2) Model queries, transformation and code generation
  - a) Applications of model queries
  - b) Model queries with OCL: core language concepts
  - c) Model queries with graph patterns (e.g. VQL): core ideas and language elements
  - d) Local search vs. incremental graph pattern evaluation
  - e) Graph transformation rules (structure + core semantics)
  - f) Causal dependence vs. conflicts in graph transformation:
  - g) Model transformations (M2T and M2M) vs Model transformation chains
  - h) Incremental model transformations (forward vs. backward vs change-driven)
  - i) Levels of incrementality in model transformations
  - j) Reactive transformations
- 3) Concrete syntax and code generation
  - a) Textual vs. graphical syntax and editors
  - b) Parser-based vs. projectional editing for textual syntax
  - c) Lexer vs. parser, AST vs. DOM
  - d) Grammar vs. derivation vs derivation tree
  - e) Architecture of graphical editors and views
  - f) Specification methods of graphical editors and views
  - g) Ad hoc / dedicated vs. templated based code generators
  - h) Direct source code generation vs. AST generation
  - i) Model-code synchronization, dealing with manually written parts
- 4) Model management, advanced modeling topics
  - a) Standard model serialization in XMI (e.g. in EMF)
  - b) Model comparison / model differencing vs model merge
  - c) Evolution: Model vs. metamodel vs. transformation
  - d) Megamodels, global model management
  - e) Back-annotation, Traceability of transformations
  - f) Offline vs. online collaborative modeling
- 5) Applications
  - a) Models and transformations in Critical Systems, V/Y Development processes
  - b) MDA: CIM vs PIM vs PSM, role and benefit of PIM-PSM mappings
  - c) Refactoring, Reverse engineering
  - d) Simulations, Functional Mock-up
  - e) Design space exploration