

Modellalapú szoftvertervezés vizsgatételek 2020

Egy vizsga két részből áll, a I. és a II. blokkból is egy-egy kérdésre kell felelni.

(Nem a formális definíciók visszaadását kérjük, hanem a fogalmak lényegét és értelmezését pl. a házi feladat kontextusában.) Figyelem: előadáson nem feltétlenül ilyen csoportosításban hangzottak el.

I. THEORY AND TECHNIQUES

- 1) Domain-specific languages, metamodeling, EMF
 - a) Metamodel vs instance model, generalization, type conformance
 - b) Metamodeling levels, multi-level metamodeling
 - c) Concrete syntax vs. abstract syntax
 - d) Well-formedness constraints
 - e) Derived features and Views
 - f) Generative vs. interpreted modeling (e.g. architecture of EMF vs. Sirius)
 - g) Core concepts in Ecore metamodels (EClass, EReference, EAttribute)
- 2) Model queries and transformations
 - 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) Rule-based model transformations, graph transformation rules (structure + core semantics)
 - f) Common challenges in rule-based systems, causal dependence in graph transformation
 - g) Model transformations (M2T and M2M) in transformation chains and flows
 - h) Incremental transformations (forward vs. bidirectional, source vs target incrementality)
 - i) Levels/granularity of incrementality in model transformations
- 3) Concrete syntax and code generation
 - a) Human aspects, textual vs. graphical syntax and editors, scaling and layouting issues
 - b) Parser-based ("raw") vs. projectional editing for textual syntax, mixed cases
 - c) The parsing process of textual languages, AST vs. DOM, scoping, linking etc.
 - d) Context-free grammar vs. derivation vs derivation tree, ambiguity
 - e) Language server architecture, motivations
 - f) Architecture of projectional editors and views, notational information, transactions
 - g) Model-driven specification methods of graphical editors and views (see e.g. Sirius)
 - h) Ad hoc vs. template-based code generators vs AST/DOM generation (serializer-based)
 - i) Model-code synchronization, dealing with manually written parts

II. APPLICATIONS AND PRACTICE

- 4) Model management, advanced modeling topics
 - a) Model persistence, model server architectures, challenges (cross-references, roundtripping)
 - b) Model versioning, model comparison / model differencing vs model merge
 - c) Evolution: Model vs. metamodel vs. transformation
 - d) Megamodels, global model management;
 - e) Back-annotation, forms of traceability for transformations
 - f) Collaborative modelling: offline/online connectivity, access control and locking
- 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) Basic ideas of Product Line Engineering and Variability Management
 - d) Simulations, Functional Mock-up
 - e) Model generators, design space exploration: criteria/goals and example applications
 - f) Partial modelling, MAVO, 3-valued logic