



Invited talk at
GEODES Research Day and Symposium
Montreal, 2022

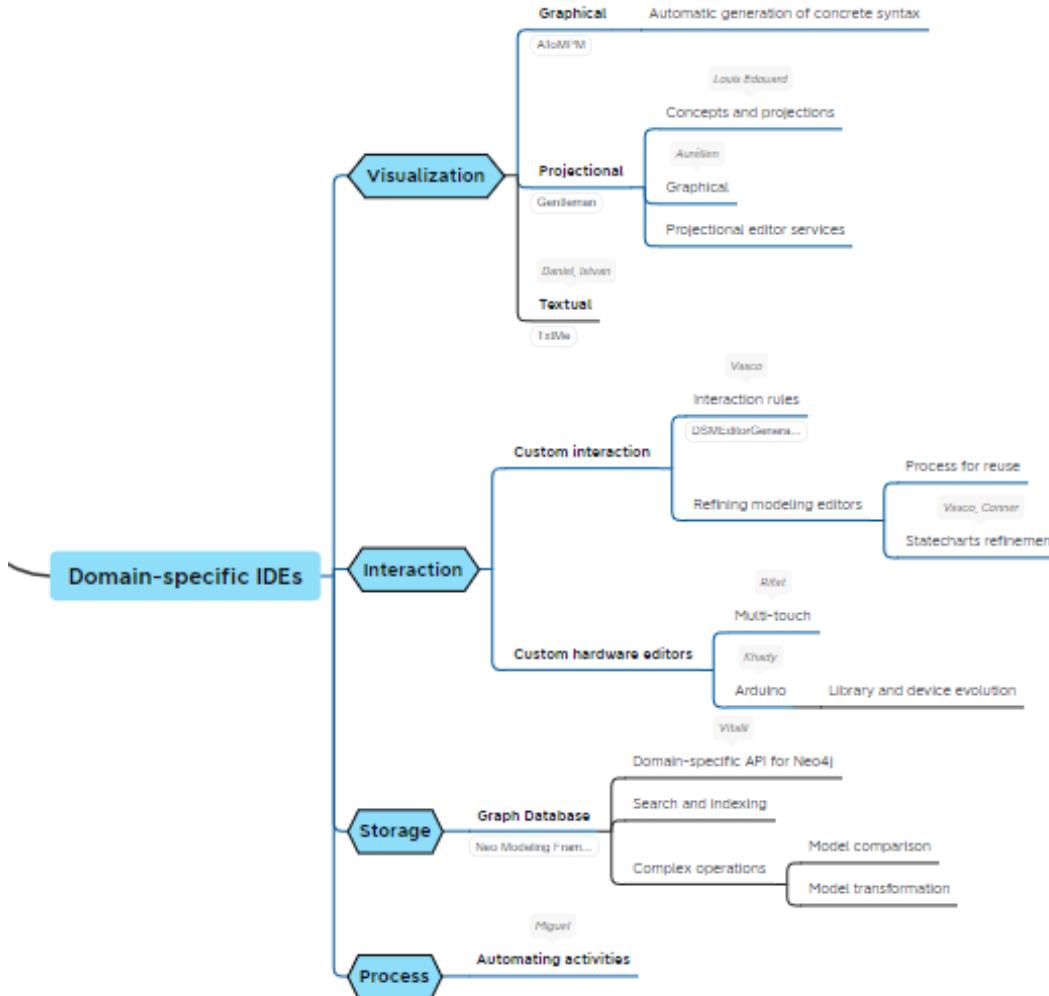
Overview of our research in software modeling and simulation

Prof. Eugene Syriani

Outline



Outline



Domain-specific IDEs – Editors

Projectional editing

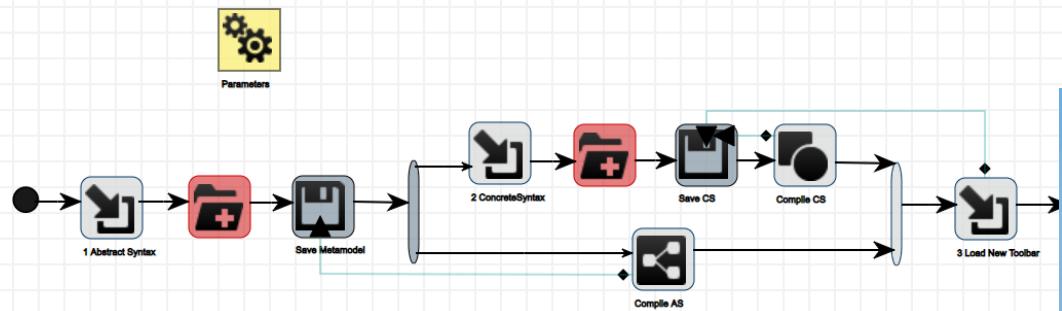


AToMPM

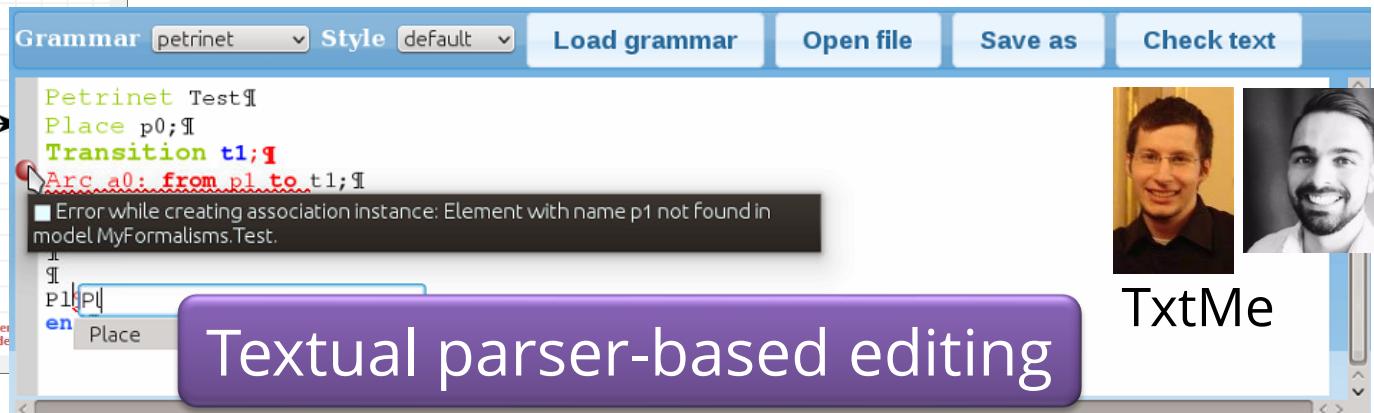
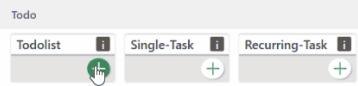
THE UNIVERSITY OF ALABAMA



localhost:8124/atompm



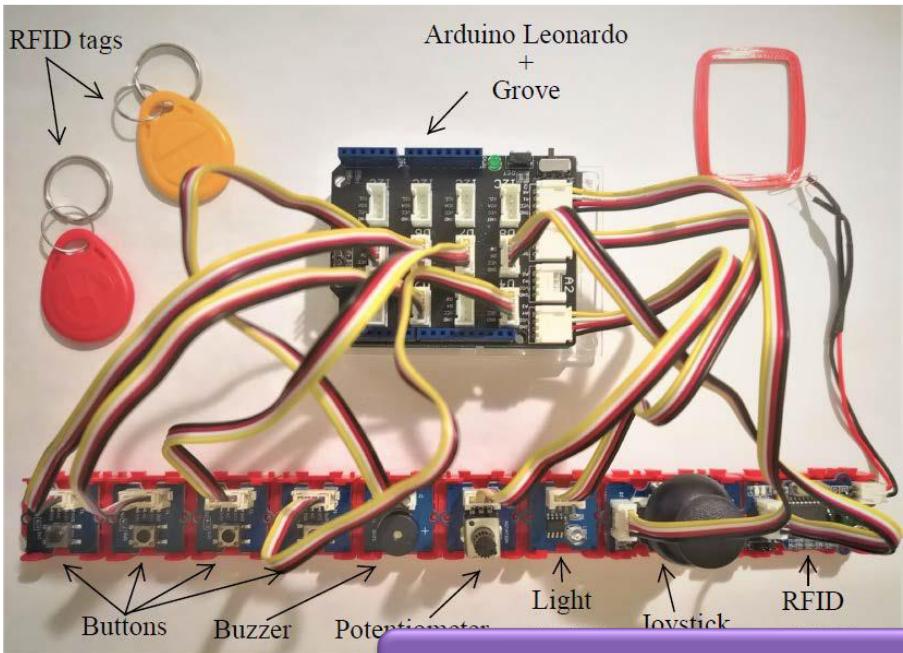
Graphical syntax-directed editing



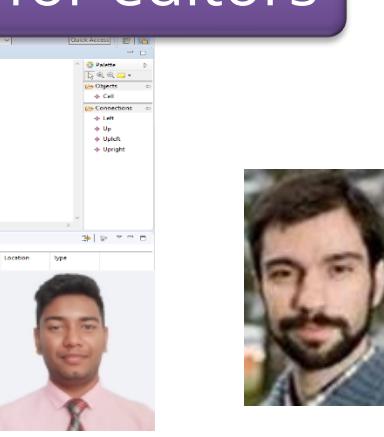
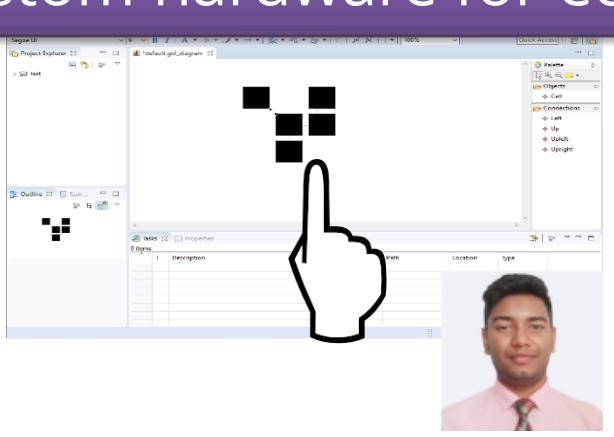
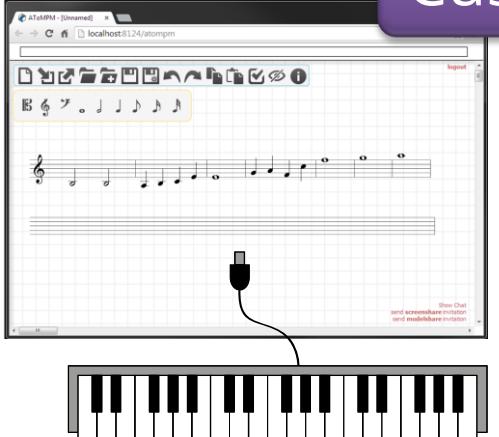
TxtMe

Textual parser-based editing

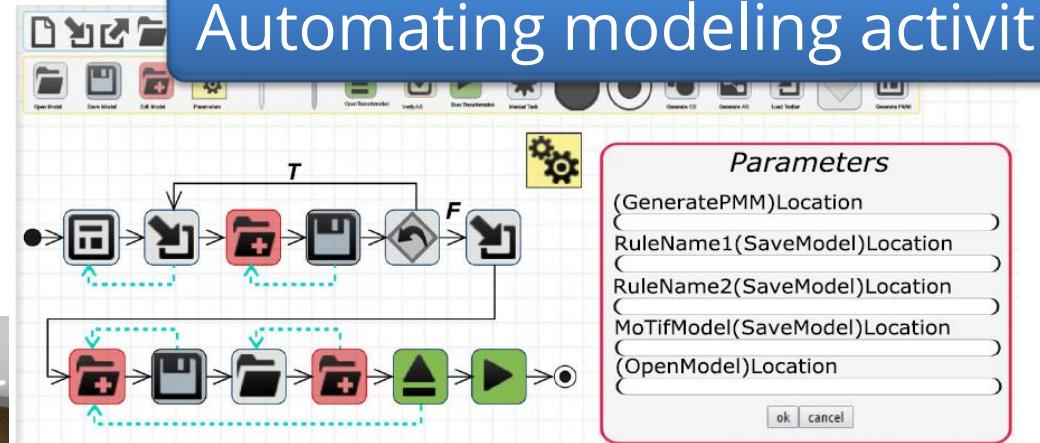
Domain-specific IDEs – Interaction



Custom hardware for editors



Automating modeling activities



```
InteractionRule StartPlay
  Condition {
    focus Interface playModelButton {}
  }
  --- press ---
  Effect {
    Interface light {value = "running"}
    Operation runGoL {}
    Interface playModelButton {value = "active"}
  }
```

```
InteractionRule EndPlay
  Condition {
    Interface playModelButton {value = "active"}
  }
  --- _finish ---
  Effect {
    Interface light {value = "fin"}
    Interface playModelButton {value = "default"}
  }
```

```
InteractionRule CreateCell
  Condition { focus Canvas {} } --- select --> Effect { Lang Cell {op = add} }
```

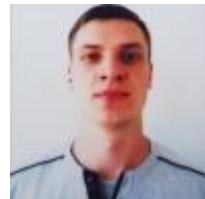
```
InteractionRule RemoveCell
  Condition { focus Lang Cell {} } --- select --> Effect { Lang Cell {op = rem} }
```

```
InteractionRule TurnLightOff
  Condition {
    focus Interface
      lightButton {}
  }
  --- press ---
  Effect {
    Interface light {
      value = "off"
    }
  }
```

Custom interactions

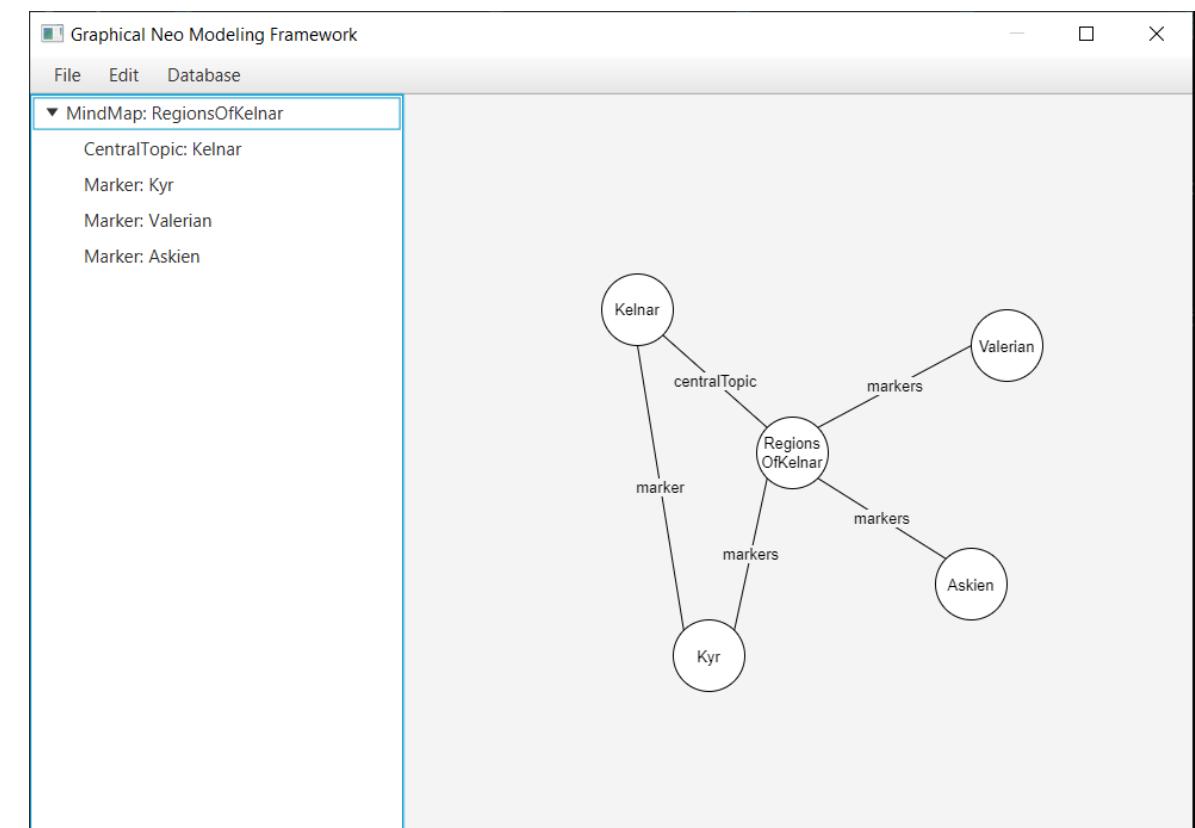
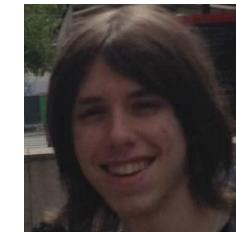
Domain-specific IDEs – Storage

- Storing models in graph databases
- NeoModelingFramework

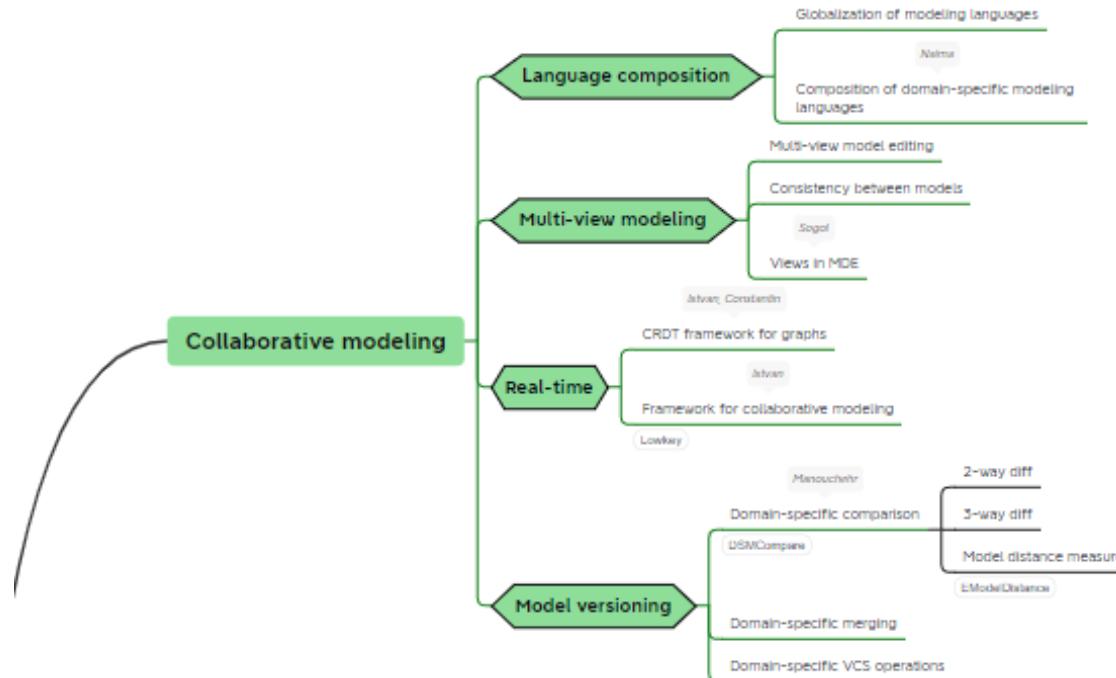


```
1 val manager = GrafModelManager(dbUri, username, password)
2 val graph = manager.createGraph()
3 graph.setName("G1")
4 val v0 = graph.addVertices(VertexType.CompositeVertex) as CompositeVertex
5 val v1 = v0.addSub_vertices(VertexType.Vertex)
6 val v2 = v0.addSub_vertices(VertexType.Vertex)
7 v0.setDefault_vertex(v1)
8 manager.saveChanges()
9 manager.close()
```

Domain-specific API



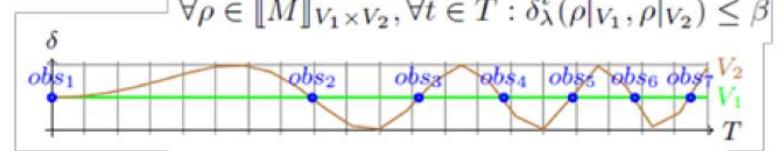
Outline



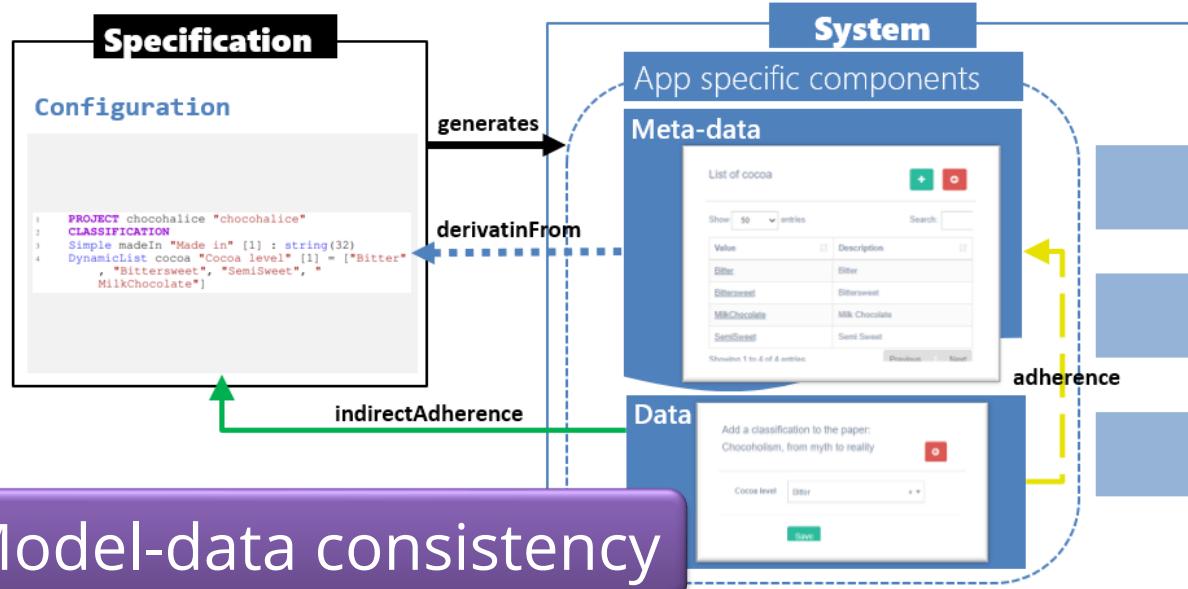
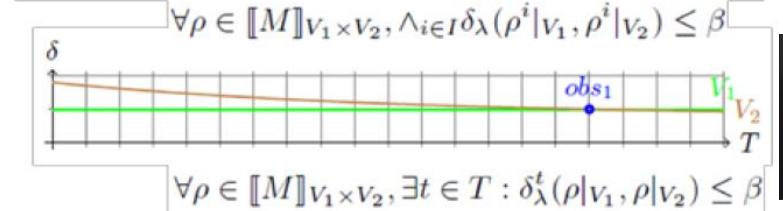
Collaborative modeling – Multi-view consistency



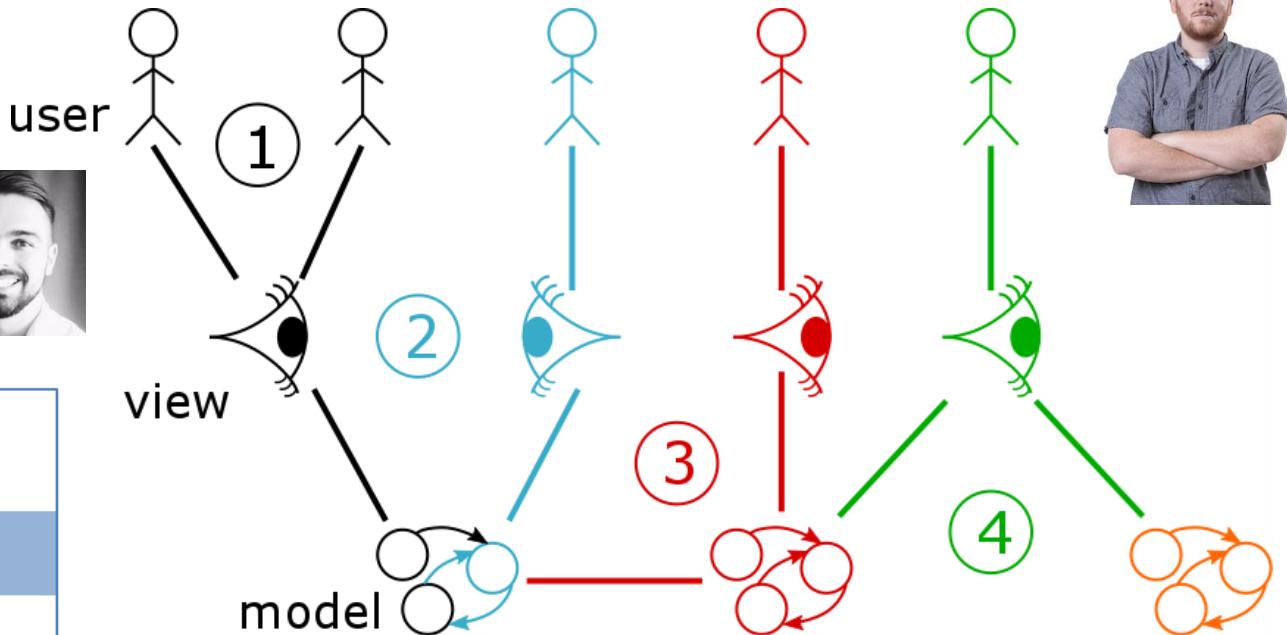
Repetitive



Eventual



Tolerating inconsistency



Multi-view modeling

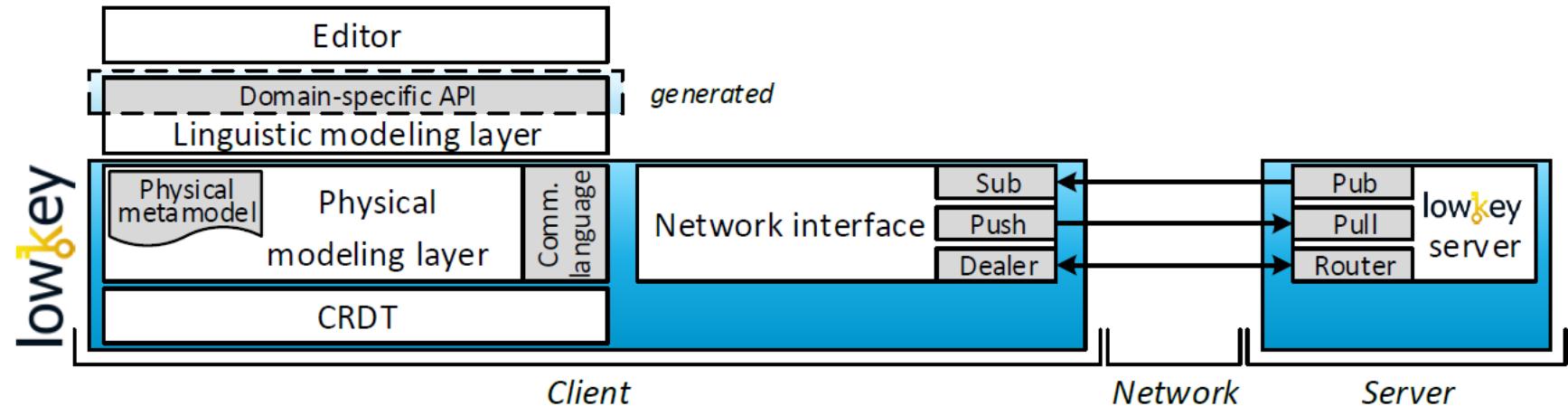
Model-data consistency

Collaborative modeling – Real-time collaboration

Linguistic well-formedness rules

Uniform object representation

Collaboration / persistence



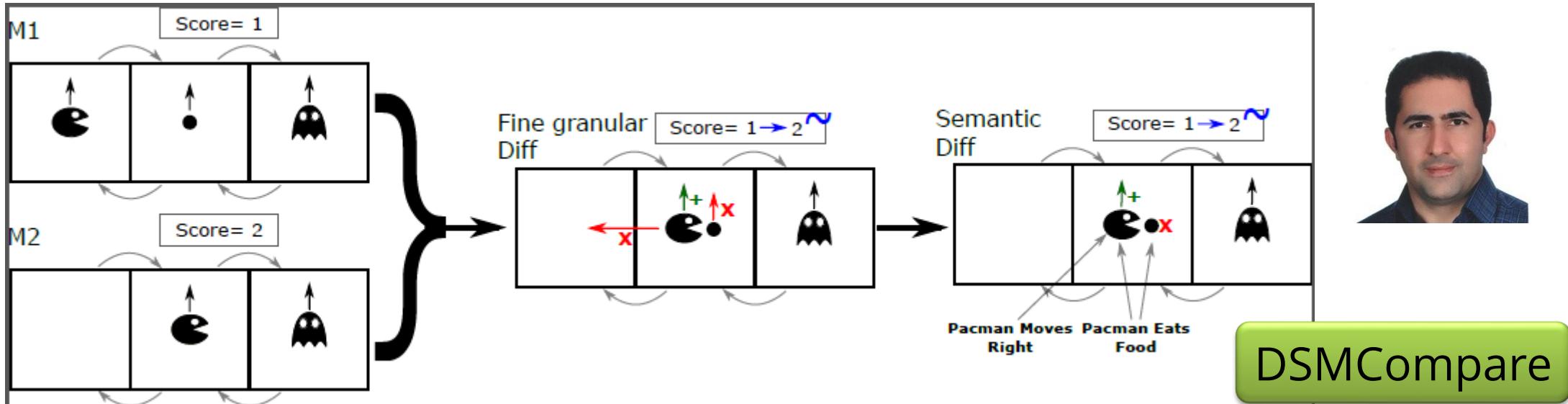
CollabServer

CRDT

lowkey



Collaborative modeling – Model versioning



Move $\Delta_M(G_1, G_2) = \sum_{m_1 \in Mov_1, m_2 \in Mov_2} \delta_M(m_1, p(m_2)), \text{ where } l(m_1) = l(m_2)$

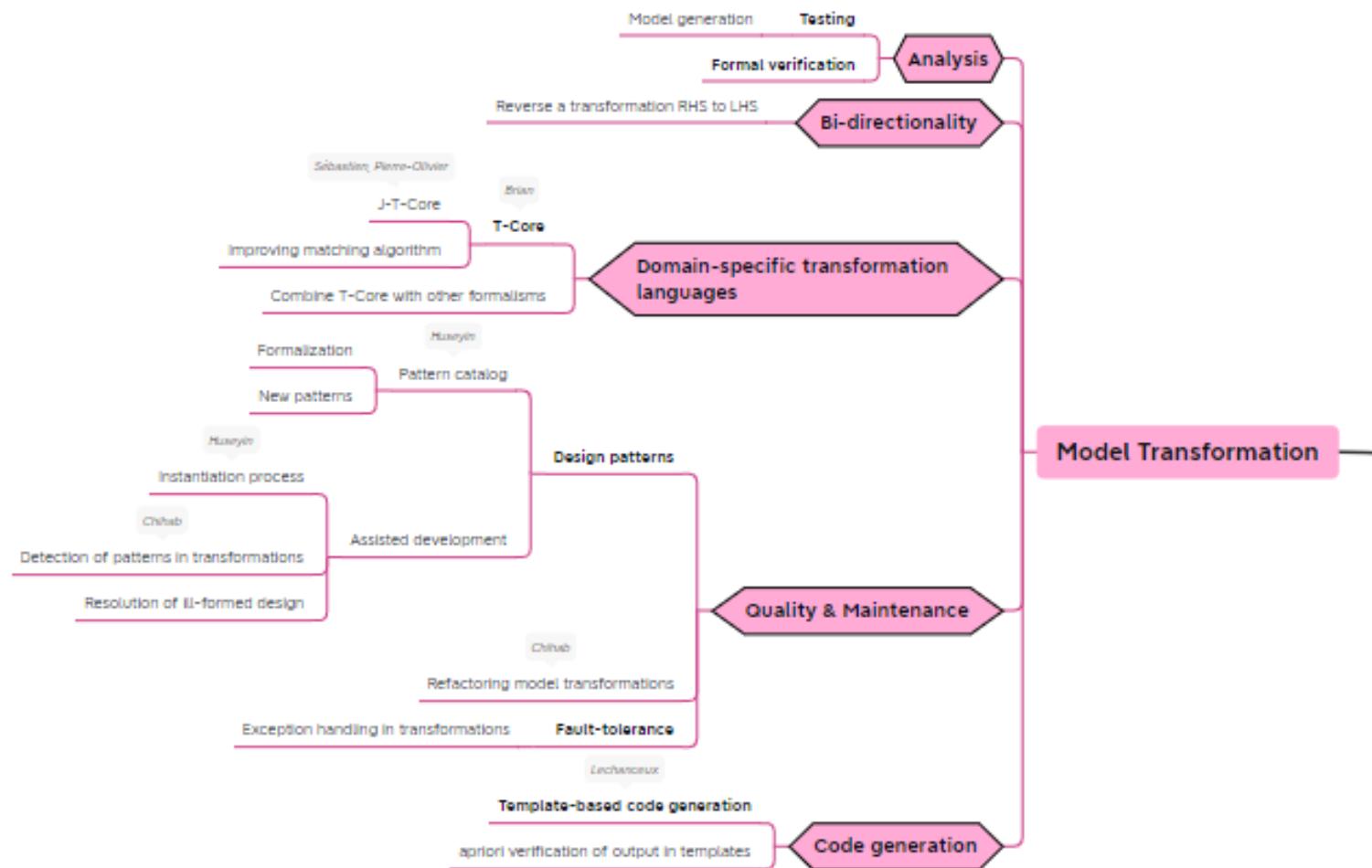


Element $\Delta_E(G_1, G_2) = \frac{|\{v_1 \in V_1 | \nexists v_2 \in V_2, l(v_2) = l(v_1)\}| + |\{v_2 \in V_2 | \nexists v_1 \in V_1, l(v_1) = l(v_2)\}|}{|V_1| + |V_2|}$

Value $\delta_V(v, x) = \begin{cases} |a(v_1, x)| & \text{if } a(v, x) = 0 \\ |a(v, x) - a(v_1, x)| / a(v, x) & \text{otherwise} \end{cases}$

Domain-specific distance

Outline

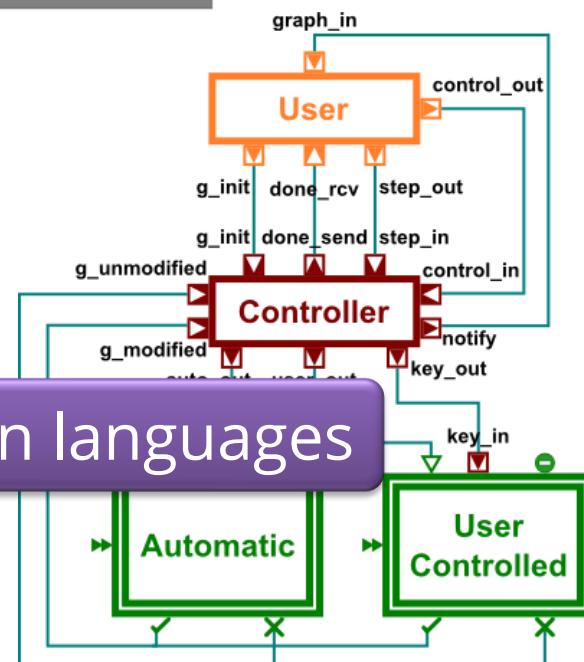


Model transformation – Domain-specific transformation languages

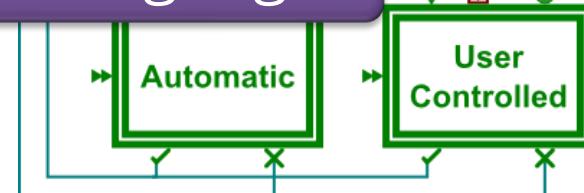
```
class ARule(Composer):
    def __init__(self, LHS, RHS):
        super(ARule, self).__init__()
        self.M = Matcher(condition=LHS, max=1)
        self.I = Iterator(max_iterations=1)
        self.W = Rewriter(condition=RHS)

    def packet_in(self, packet):
        self.is_success = False
        packet = self.M.packet_in(packet)
        if not self.M.is_success: return packet
        packet = self.I.packet_in(packet)
        if not self.I.is_success: return packet
        packet = self.W.packet_in(packet)
        if not self.W.is_success: return packet
        self.is_success = True
        return packet
```

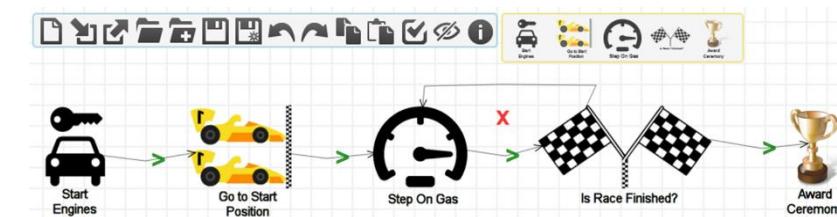
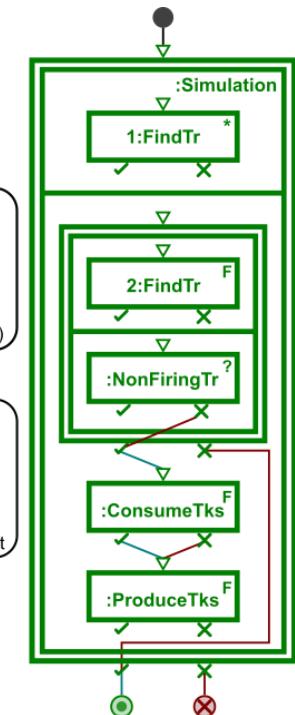
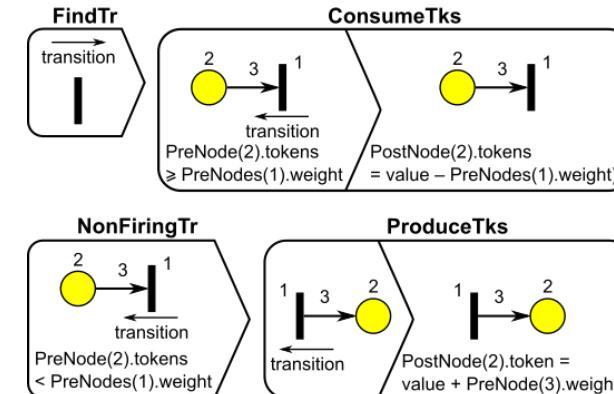
T-Core



Custom transformation languages

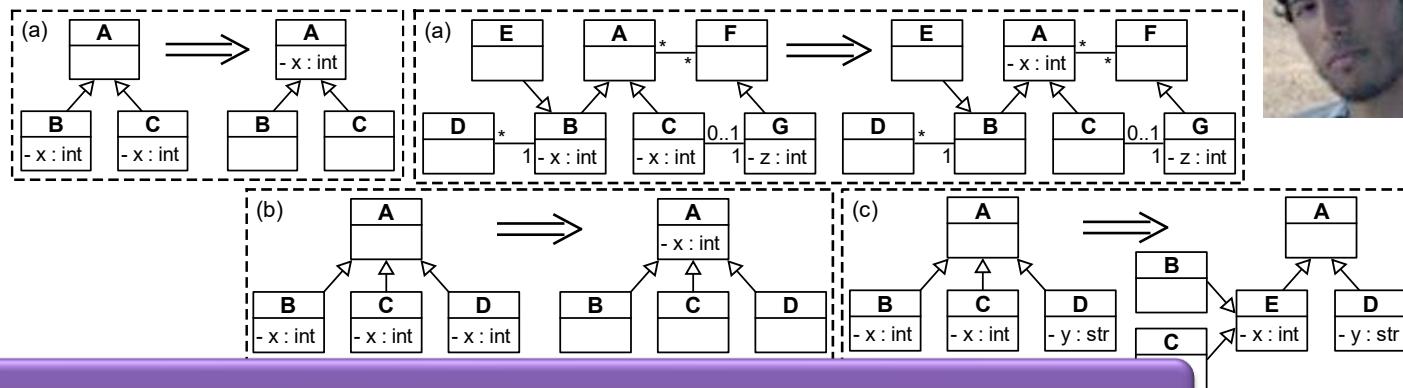
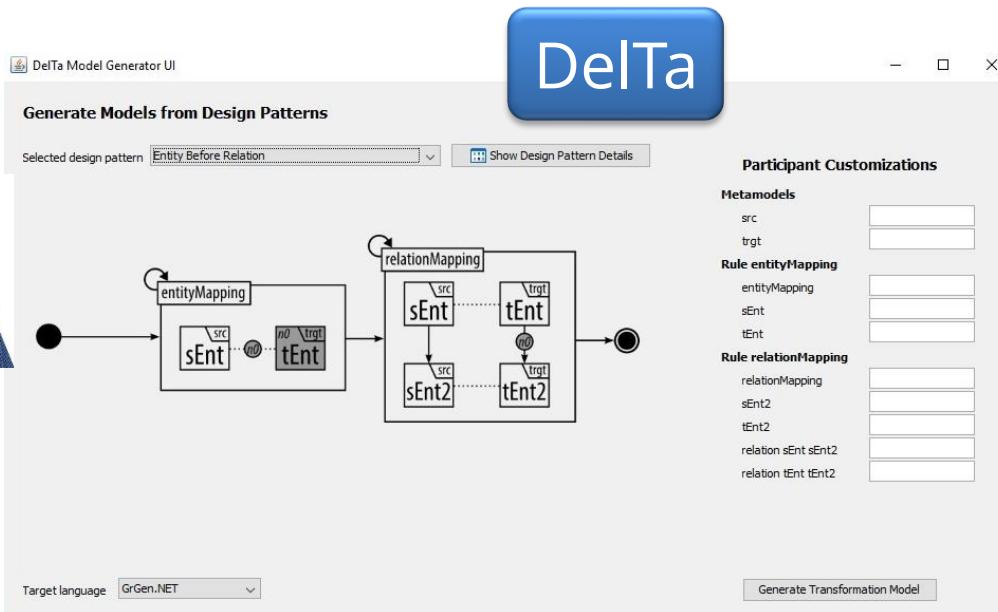


MoTif



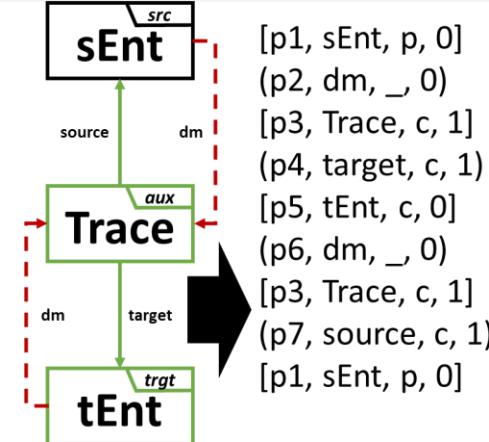
Higher-order transformations

Model transformation – Design patterns

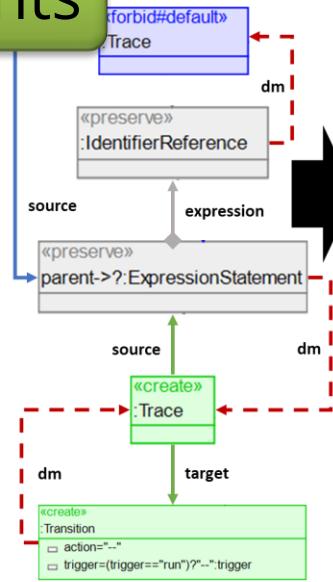


Refactoring transformations by-example

Detection of variants



- [p1, sEnt, p, 0]
- (p2, dm, _, 0)
- [p3, Trace, c, 1]
- (p4, target, c, 1)
- [p5, tEnt, c, 0]
- (p6, dm, _, 0)
- [p3, Trace, c, 1]
- (p7, source, c, 1)
- [p1, sEnt, p, 0]



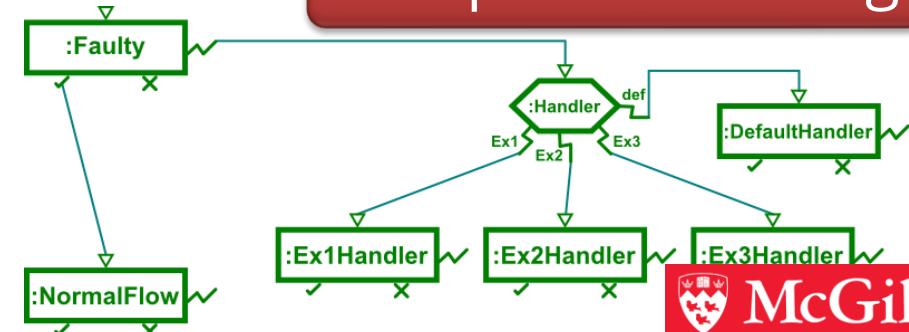
- [t10, Trace, f]
- (t11, source, c)
- [t1, ExprStmt, p]
- (t2, dm, _)
- [t3, Trace, c]
- (t4, target, c)
- [t5, Transition, c]
- (t6, dm, _)
- [t3, Trace, c]
- (t7, source, c)
- [t1, ExprStmt, p]
- (t8, expression, c)
- [t9, IdentifRef, p]
- (t12, dm, _)
- [t10, Trace, f]

(b)

(c)

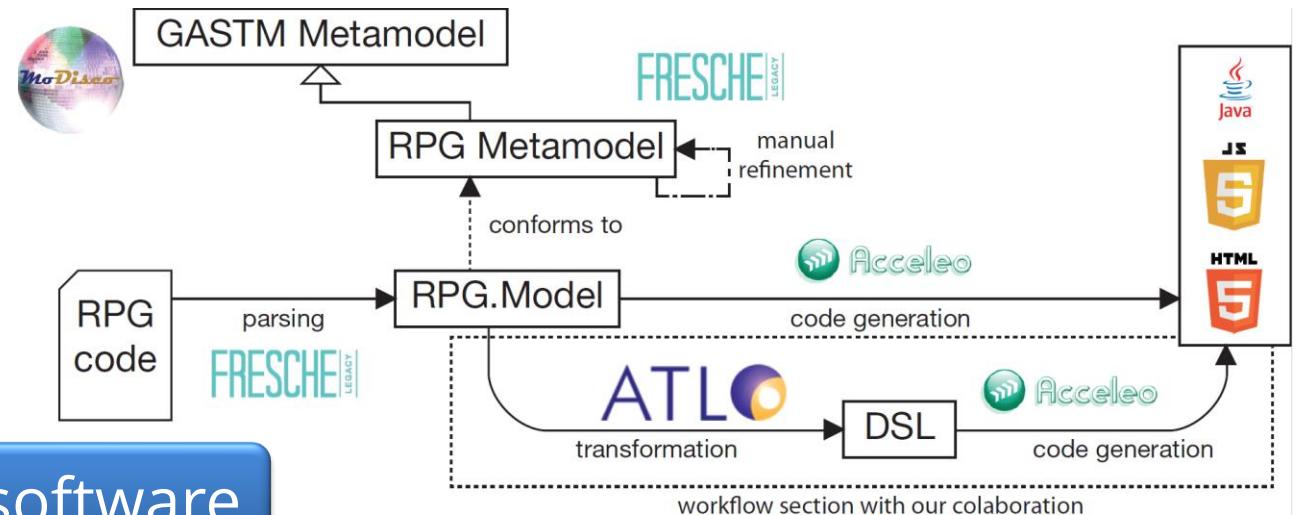
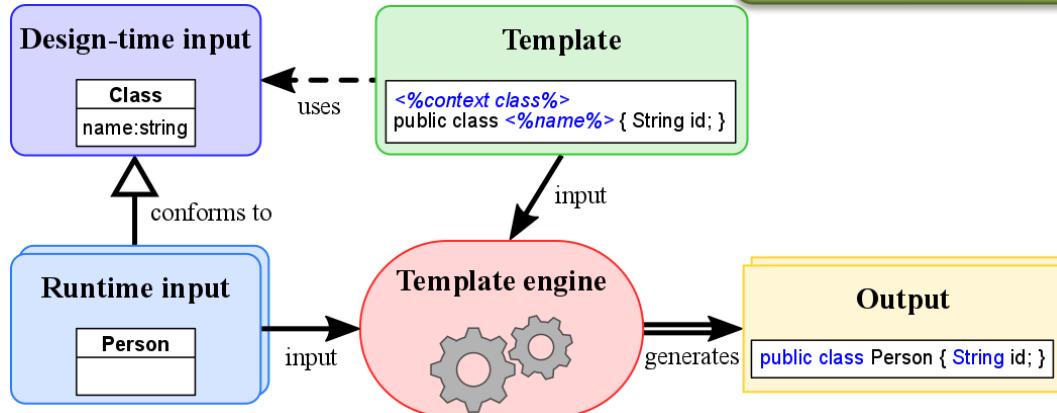
(d)

Exception handling



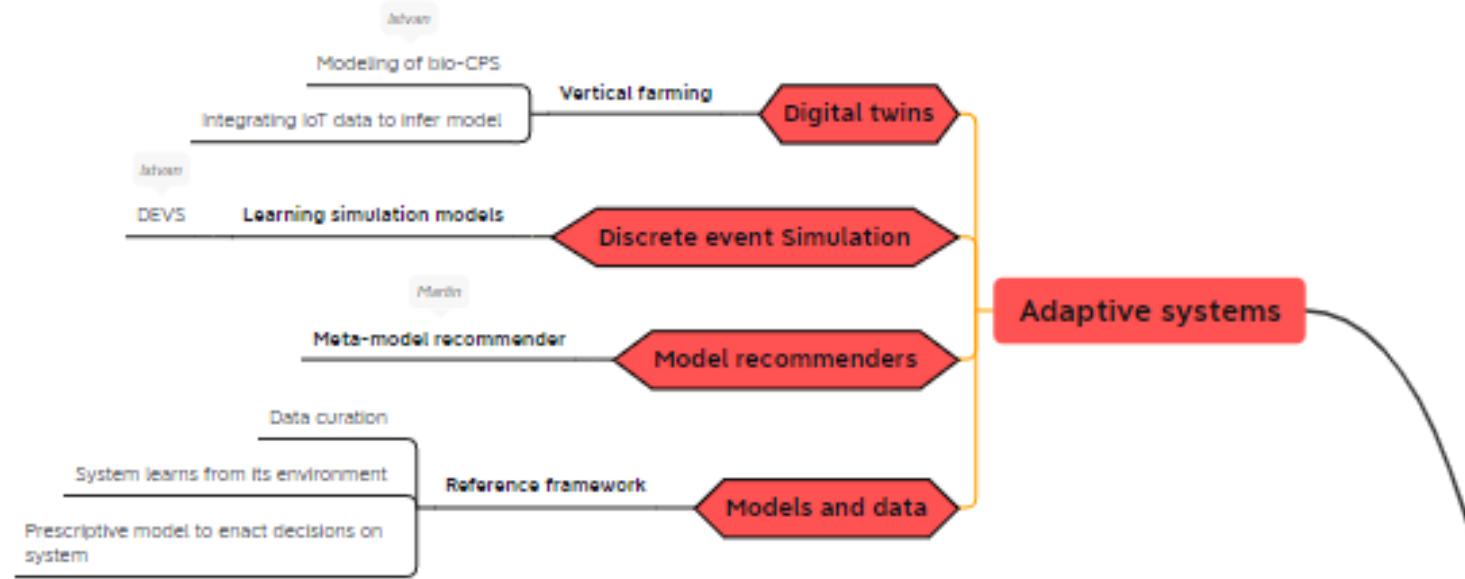
Model transformation – Code generation

Template-based code generation



Modernization of legacy software

Outline

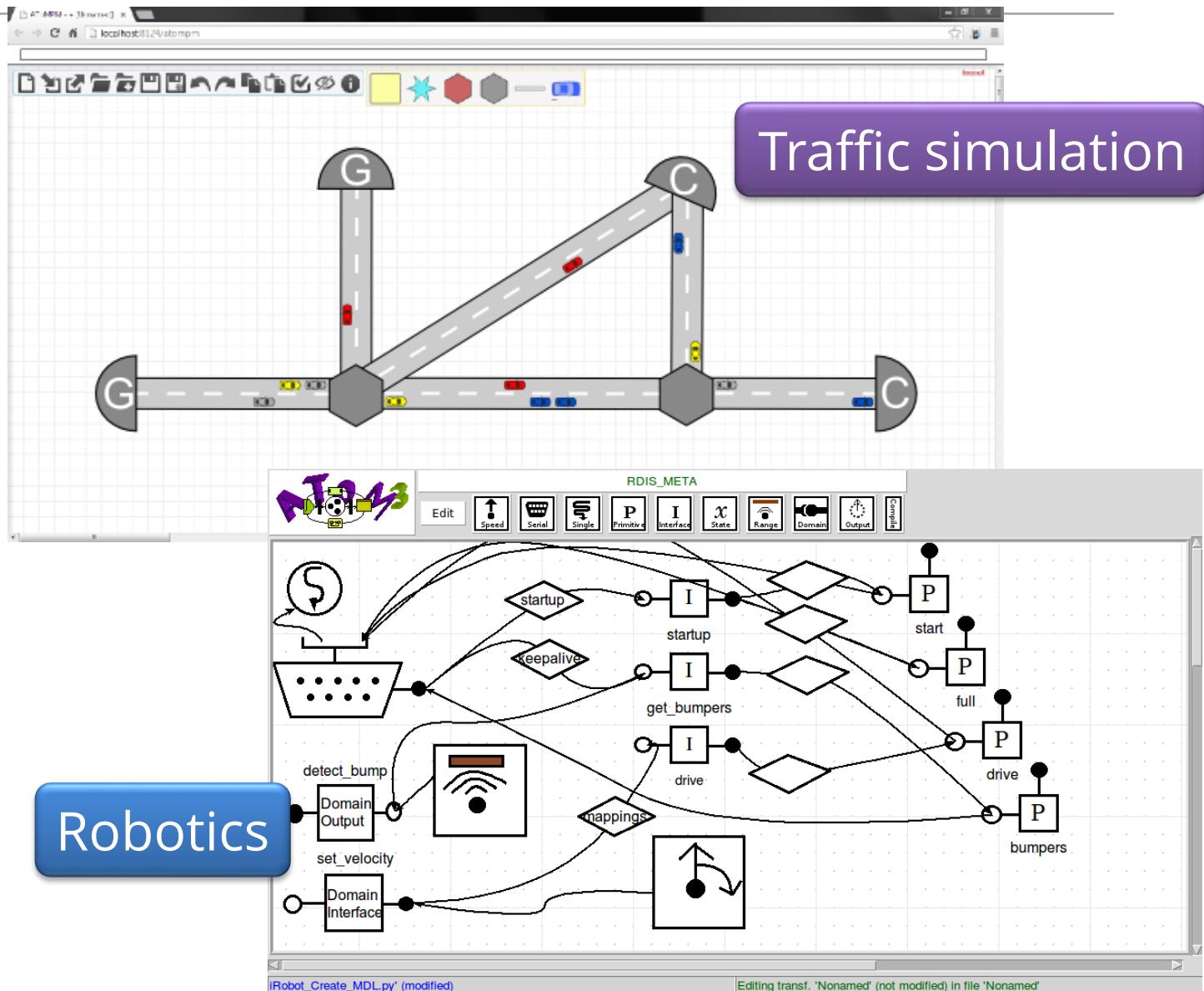
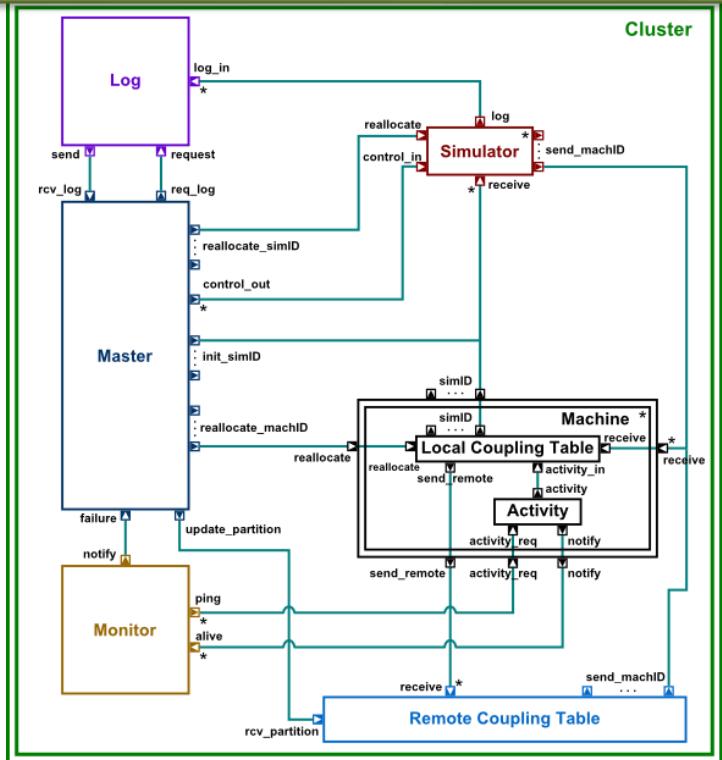


Adaptive systems – Simulation

THE UNIVERSITY OF ALABAMA

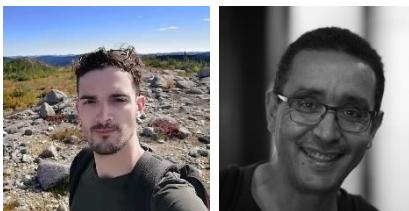


Distributed DEVS simulators

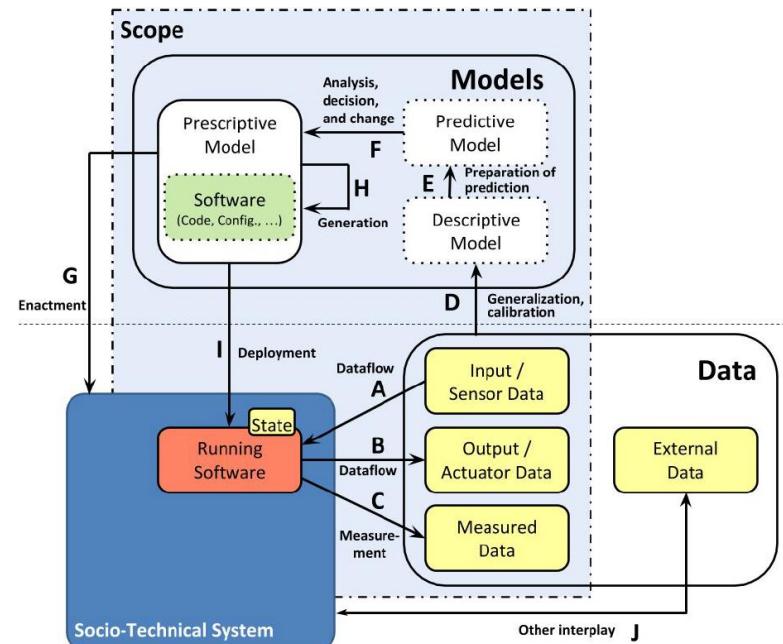
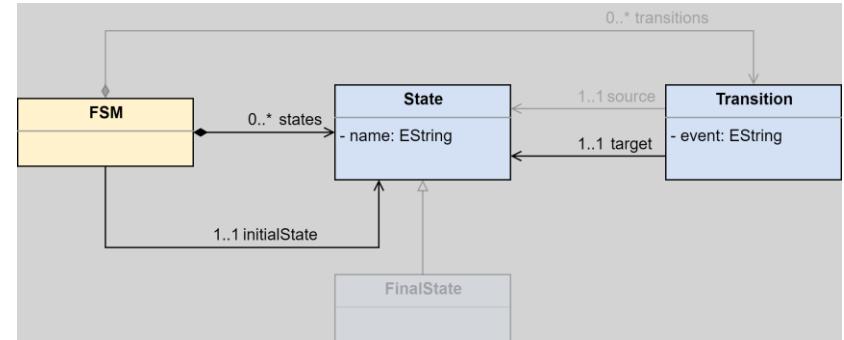


Adaptive systems – Data and machine learning

B. Combemale, J. Kienzle, G. Mussbacher, H. Ali, D. Amyot, M. Bagherzadeh, E. Batot, N. Bencomo, B. Benni, J. Bruel, J. Cabot, B. Cheng, P. Collet, G. Engels, R. Heinrich, J. Jezequel, A. Koziolek, S. Mosser, R. Reussner, H. Sahraoui, R. Saini, J. Sallou, S. Stinckwich, E. Syriani, and M. Wimmer. A Hitchhiker's Guide to Model-Driven Engineering for Data-Centric Systems. *IEEE Software*: 38(4), pp. 71–84 (2020).

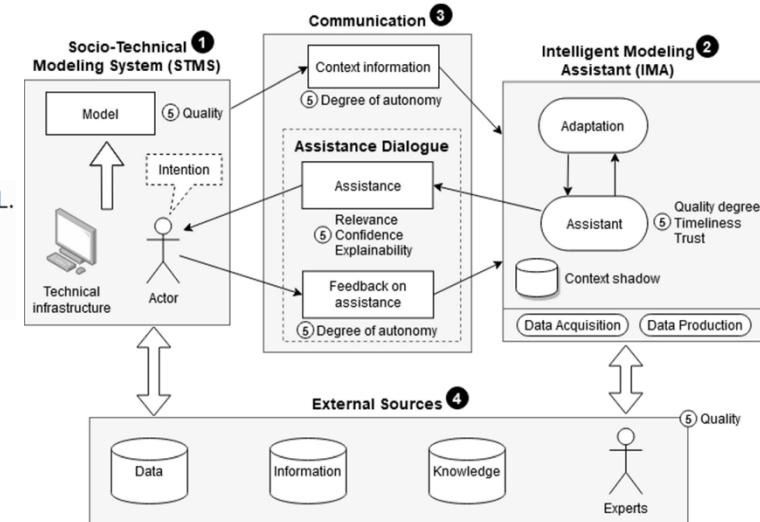


Meta-model recommenders



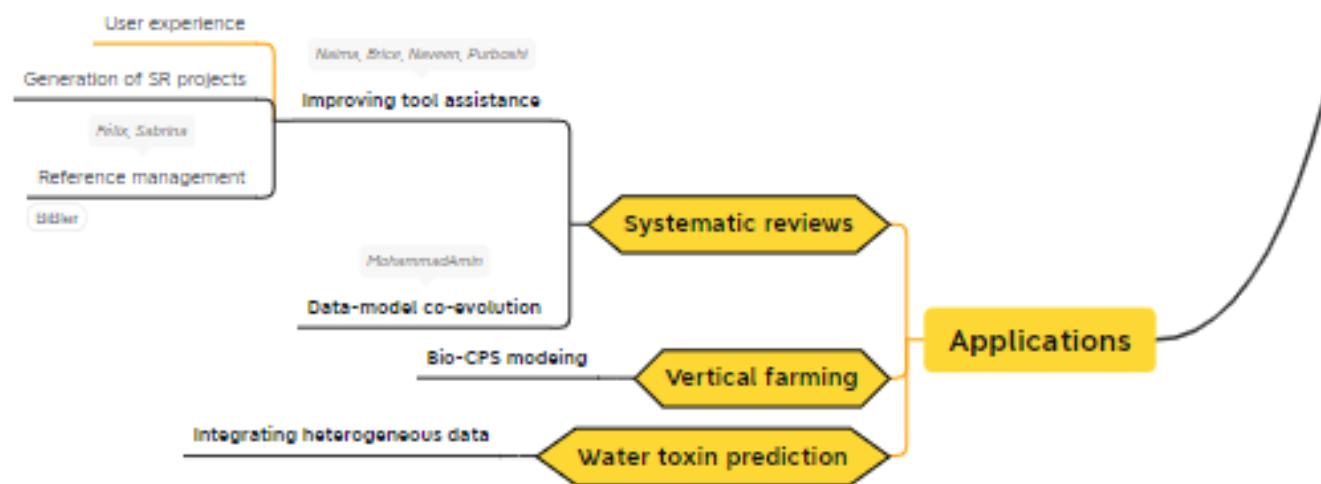
G. Mussbacher, B. Combemale, J. Kienzle, S. Abrahão, H. Ali, N. Bencomo, M. Búr, L. Burgueño, G. Engels, P. Jeanjean, J. Jézéquel, T. Kühn, S. Mosser, H. Sahraoui, E. Syriani, D. Varró, and M. Weyssow. Opportunities in Intelligent Modeling Assistance. *Software and Systems Modeling*: 19, pp. 1045–1053. (2020).

Models and data reference framework



Intelligent modeling assistance

Outline



Systematic reviews

Model transformation

ReLiS editor

DSL Forge Workbench v0.9.1

File Edit Help

*mt.relis

```

PROJECT mt "Model transformation"
1 SCREENING
2 Review
3 Conflict on Criteria resolved_by Unanimity
4 Criteria: "Title", "Abstract", "Keywords", "Less than 4 pages", "Not using model transformation"
5 Validation 20% Normal
6 Min_score 5
7 Phase 1 "Screen per title[Fields>Title"
8 Phase 2 "Screen per title and abstract" [FieldTitle, Abstract, Paper]
9 QA
10 Questions ["Does the study have validation?", "Are RQs clearly stated?"]
11 Answers["Yes", "Partially", "No"]
12 Min_score 5
13 DATA EXTRACTION
14 Transformation_name "Transformation name" [1] string(100)
15 DynamicList trans_language "Transformation Language" [1] "Model", "Henshing", "QVT"
16 List multiple bidirectional [1] "Exogenous", "Endogenous", "Outgoing"
17 DynamicList intent [1] "Transformation", "Simulation", "Migration" []
18 DynamicList intent_relation "Intent relation" [0] "Relation" = ["Sequence", "Inverse"] []
19 DynamicList intent_1 "Intent 1" depends_on intent
20 DynamicList intent_2 "Intent 2" depends_on intent
21 note
22 note
23 SYNTESIS
24 1dChart scope "Scope" on scope charts(Pie, Bar)
25 2dChart year_scope "Scope per year" on scope per year charts(Line)
26 3dChart year_scope "Scope per year" on scope per year charts(Line)

```

Paper * Paper_21 - SW-14 pattern

Transformation Language Model

Scope Outplace

Bidirectional

Intent

Simulation

Translation

Note

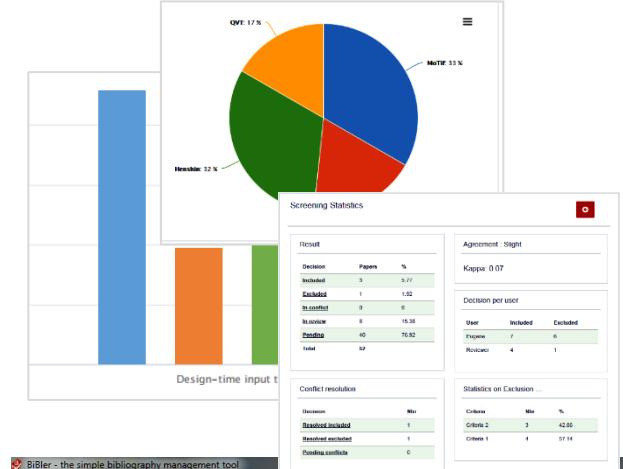
Save

Decision

Include Excluded

Excluded criteria Note

Save and Next



#	P	Type	Author	Title	Year	Key
1		BOOK	V(O)Ous et al.	(P)riniciples of distributed database systems	2011	Agrawa2011
2		INPROCEEDINGS	(A)kerebis et al.	(W)ebs (A)pplications (D)evelopment with (W)ebs(ML) and (W)ebs(R)atio 5.0	2008	Acerbi2008
3		INPROCEEDINGS	(A)lgerro et al.	(H)ow to preserve the benefits of (D)eign (P)atterns	1998	Agerbo1998
4		INPROCEEDINGS	(A)grawal	(M)etamodel based model transformation language	2003	Agrawa2003
5		INPROCEEDINGS	(A)grawal	(R)eusable (I)ldoms and (P)atterns in (G)raph (T)ransformation (L)anguages	2005	Agrawa2005
6		ARTICLE	(A)grawal et al.	(T)he (D)esign of a (L)anguage for (M)odel (T)ransformations	2006	Agrawa2006
7		ARTICLE	(A)grawal et al.	(G)raph (T)ransformation on (D)omain-specific (M)odels	2003	Agrawa2003a
8		INPROCEEDINGS	(A)lbed et al.	(A)spect-(O)riented (M)odeling and (I)nformation (H)iding	2009	AlAbd2009
9		INPROCEEDINGS	(A)lamri et al.	(T)owards (C)loud-(B)ased (S)oftware (P)rocess (M)odelling and (E)nactment	2014	Alamri2014
10		INPROCEEDINGS	(A)lbini-(A)miet et al.	(I)nstantiating and (D)etecting (D)eign (P)atterns (Placing (B)its and (P)ieces (T)ogether	2001	Albin
11		BOOK	(A)lexander et al.	(A)(P)attern (L)anguage (T)owns, (B)uildings, (C)onstruction	1977	Alexander1977
12		ARTICLE	(A)lfaro et al.	(I)nterface automata	2001	Alfar2001
13		TECHREPORT	(A)lgers et al.	(SMARTES) (D)eliverable 3: (R)eview of (M)icro (S)imulation (M)odels	1997	Algiers1997
14		INBOOK	(A)lonso et al.	(U)ser-(I)nterface (M)odelling for (B)lind (U)users	2008	Alonso2008
15		ARTICLE	(A)lur et al.	(A)(T)heory of (T)imed (A)utomata	1994	Alur1994
16		INPROCEEDINGS	(A)melunen et al.	(MOFLON): (A)(S)tandard-(C)ompliant (M)etamodeling (F)ramework with (G)raph (T)ransformations	2006	Amelunen2006
17		INPROCEEDINGS	(A)mmann et al.	(U)sing (M)odel (C)hecking to (G)enerate (T)ests from (S)pecifications	1998	Ammann1998
18		INPROCEEDINGS	(A)mmann et al.	(T)owards a (M)odel (T)ransformation (I)nent (C)atalog	2012	Amran2012
19		INPROCEEDINGS	(A)mmann et al.	(A)(T)ridimensional (A)pproach for (S)tudying the (F)ormal (V)erification of (M)odel (T)ransformations	2012	Amran2012a
20		INPROCEEDINGS	(A)mmann et al.	(M)etrics for (M)odel (T)ransformations	2010	Amst2010
21		INPROCEEDINGS	(A)mmann et al.	(P)erformance in (M)odel (T)ransformations: (E)xperiments with (ATL) and (QVT)	2011	Amst2011
22		INPROCEEDINGS	(A)mmann et al.	(P)erformance in (M)odel (T)ransformations: (E)xperiments with (ATL) and (QVT)	2011	Amst2011a
23		INPROCEEDINGS	(A)ndries et al.	(M)odel (T)ransformation (A)nalysis: (S)taying (A)head of the (M)aintenance (N)ightmare	2011	Andrie2011
24		INPROCEEDINGS	(A)ndries et al.	(M)etrics for (A)nalyzing the (Q)uality of (M)odel (T)ransformations	2008	Andrie2008
25		ARTICLE	(A)ndries et al.	(G)raph (T)ransformation for (S)pecification and (P)rogramming	1999	Andrie2009
26		TECHREPORT	(A)nquetti et al.	igraph (D)atabase (Project)	2009	Anquetin2009

Edit fields

BibTeX

ARTICLE(Agrawal2006,
author = {(A)grawal, (A)ditya and (K)arasi, (G)abor and (K)almar, (Z)solt and (N)eema, (S)andeep and (S)hi, (F)eng and (V)izhanyo, (A)ttila},
journal = {(S)oftware \& (S)ystems (M)odeling},
title = {(T)he (D)esign of a (L)anguage for (M)odel (T)ransformations},
year = {2006},
number = {3},
pages = {261--288},
volume = {5}

Preview reference

ARTICLE (Agrawal2006)

A. Agrawal, G. Karsai, S. Kalmar, S. Neema, F. Shi, and A. Vizhanyo. The Design of a Language for Model Transformations. *Software & Systems Modeling*, 5(3), pp. 261–288 (2006).

Total: 757

ReLiS



Bibler

Upcoming projects

Bio-Cyber-Physical System modeling and simulation

Vertical farming



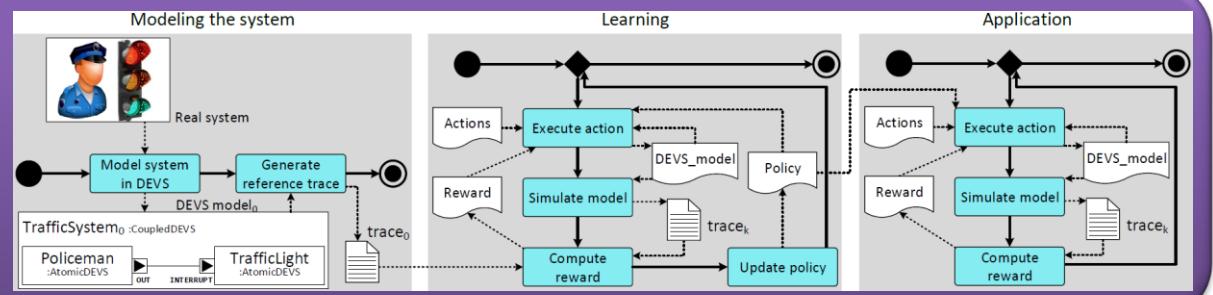
Integration of heterogeneous data for AI model validation

Water toxin prediction



Learning DEVS simulation models

Digital twins



Overview of our research in software modeling & simulation

The collage consists of 12 panels arranged in a grid:

- Top Row:**
 - AToMPM:** A screenshot of a software interface showing a process flow diagram with various nodes and connections.
 - Gentleman:** A screenshot of a software interface showing a dashboard with icons representing different systems or components.
 - afModelManager:** A snippet of Java code for managing a graph database.
 - Hardware Setup:** A photograph of an Arduino Leonardo connected to various sensors (RFID tags, Buttons, Buzzer, Potentiometer, Light sensor, Joystick, RFID sensor) and actuators.
 - InteractionRule:** A snippet of Interaction Rule language defining a rule to turn off a light when a button is pressed.
- Middle Row:**
 - Human Interaction Model:** A diagram showing four types of interactions between a user and a model: 1. Direct, 2. Indirect, 3. Mediated, and 4. Observed.
 - System Architecture:** A diagram of a layered system architecture: Application layer, System layer, Editor layer, Physical modeling layer, Comm. language, Network layer, and CRDT layer.
 - Client:** A diagram of a client system showing a fine-grained difference engine comparing two states.
 - Controller:** A detailed block diagram of a controller system with various inputs and outputs.
- Bottom Row:**
 - Metamodel:** A screenshot of a software interface showing a metamodel definition for the RPG Metamodel.
 - FRESCHETM:** A diagram of a software engineering framework involving Prescriptive Model, Predictive Model, Descriptive Model, and Software.
 - ATL Transformation:** A screenshot of a software interface showing a transformation workflow section.
 - Simulation:** A screenshot of a software interface showing a simulation of a road network with traffic and infrastructure.
 - Decision:** A screenshot of a software interface showing a decision-making interface with various buttons and status indicators.