

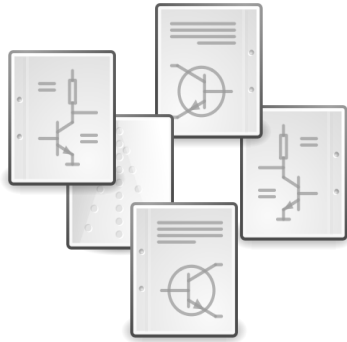
Managing Design-Time Uncertainty

Michalis Famelis, Marsha Chechik

MODELS 2017

Austin TX, USA

Uncertainty in Software Development



Many design alternatives



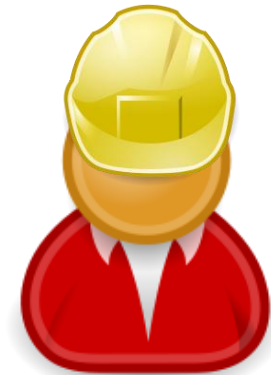
Incomplete information



Conflicting stakeholder opinions



Uncertainty during the design of software.



Uncertainty in:



Environment

What conditions will the system operate in?

Main concern:
adapting to change

Mitigated by uncertainty-aware
software



Design-time


What should the system be like?

Main concern:
making design decisions

Mitigated by uncertainty-aware
software development methodology

Management of Design-Time Uncertainty

Key development goals:

★ Quality  Speed (time to market)

What can developers do?

Make a **provisional** decision and “run with it” ★ 

Wait until uncertainty gets resolved ★ 

Fork and maintain a set of solutions  ★

We propose:

★  **Defer resolution** of uncertainty but incorporate uncertainty handling into the development process to allow progress

Outline



Articulation of uncertainty

- Partial Models:
 - Semantics
 - Notation



Deferral of decisions

- Lifting:
 - Verification
 - Diagnosis
 - Transformation

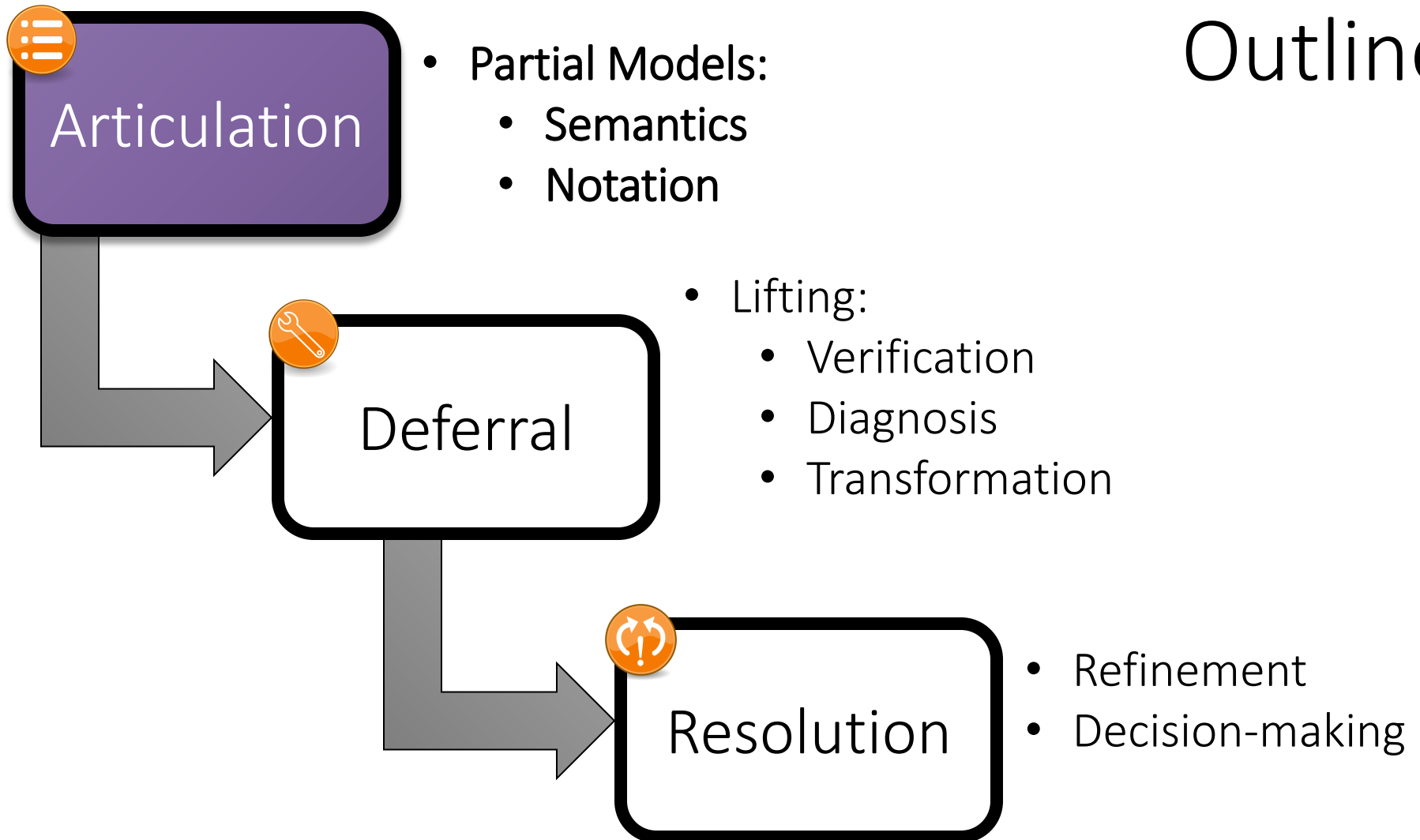


Resolution of uncertainty

- Refinement
- Decision-making

- Methodology and Tool Support
- Worked-out Examples
- Conclusion, Future Work

Outline



- Methodology and Tool Support
- Worked-out Examples
- Conclusion, Future Work

Design-time Uncertainty

Known
Knowns

Design decisions
assumed known

Alternative solutions
assumed elicited

Possibilities

Known
Unknowns

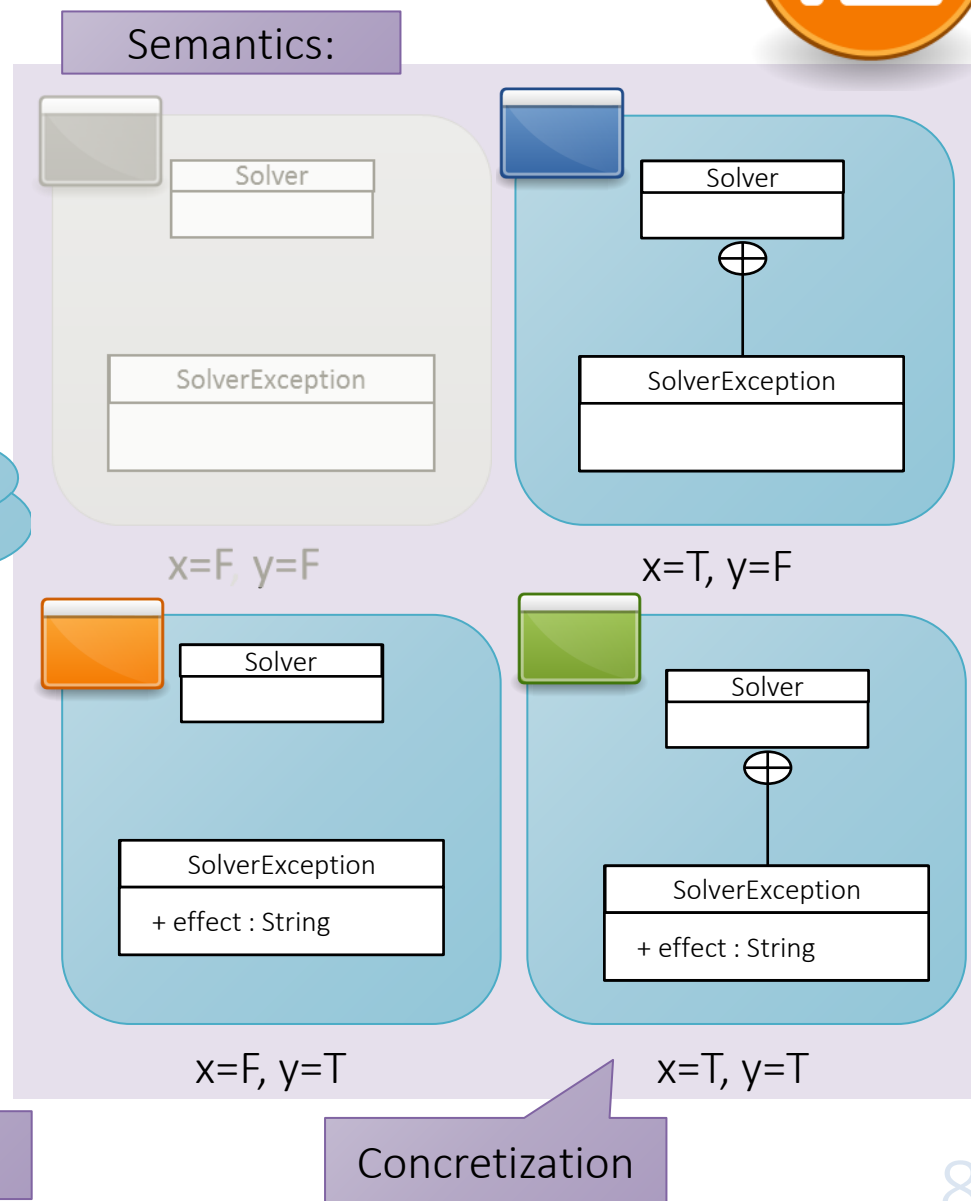
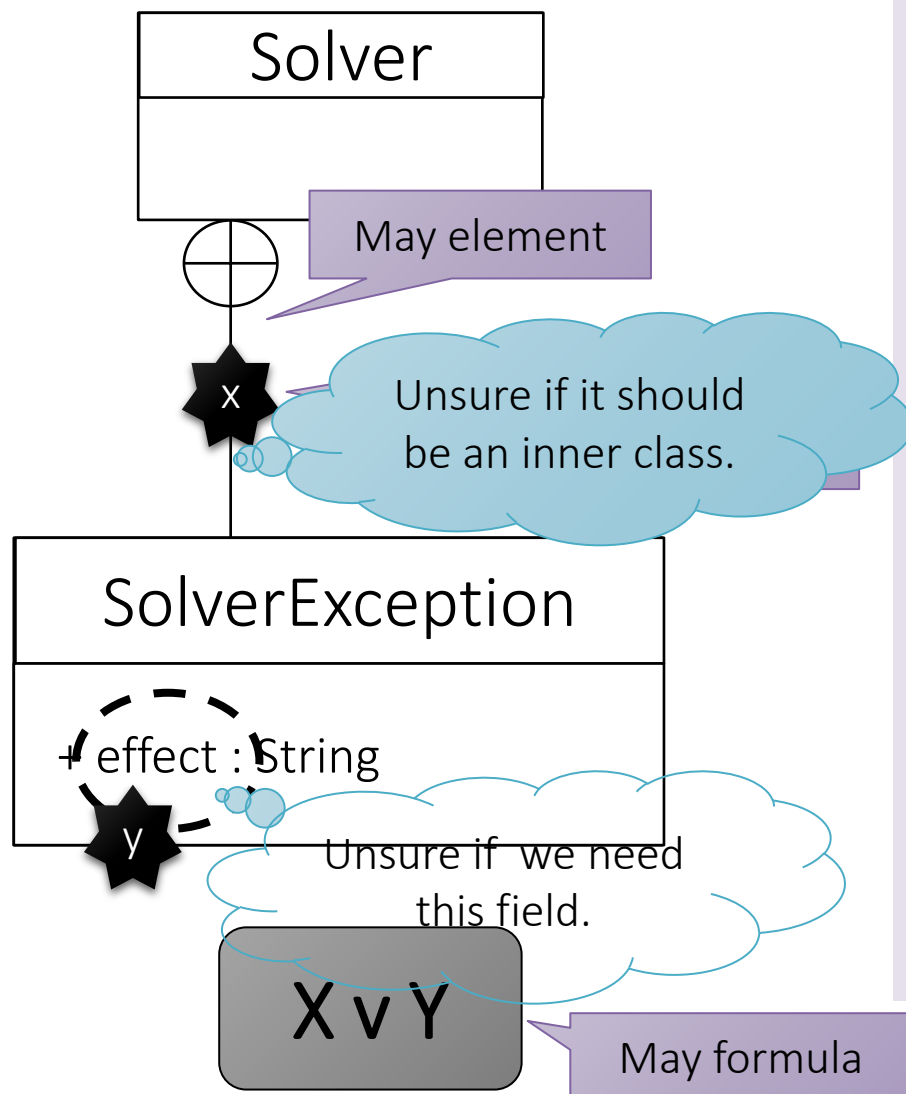
Unknown
Unknowns



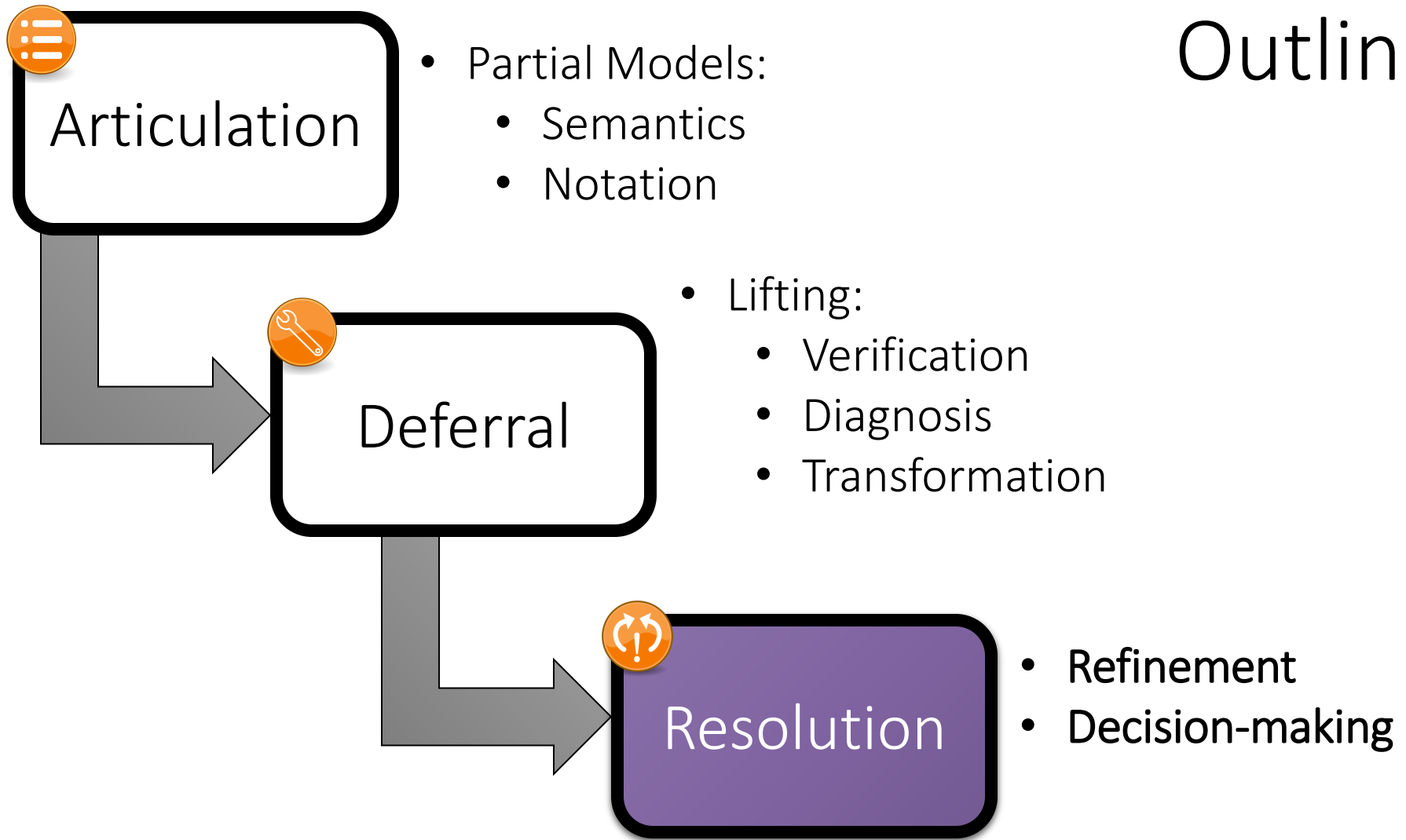
US Secretary of Defense, Donald Rumsfeld
discusses Iraqi WMDs, February 12, 2002



Representing Uncertainty with Partial Models

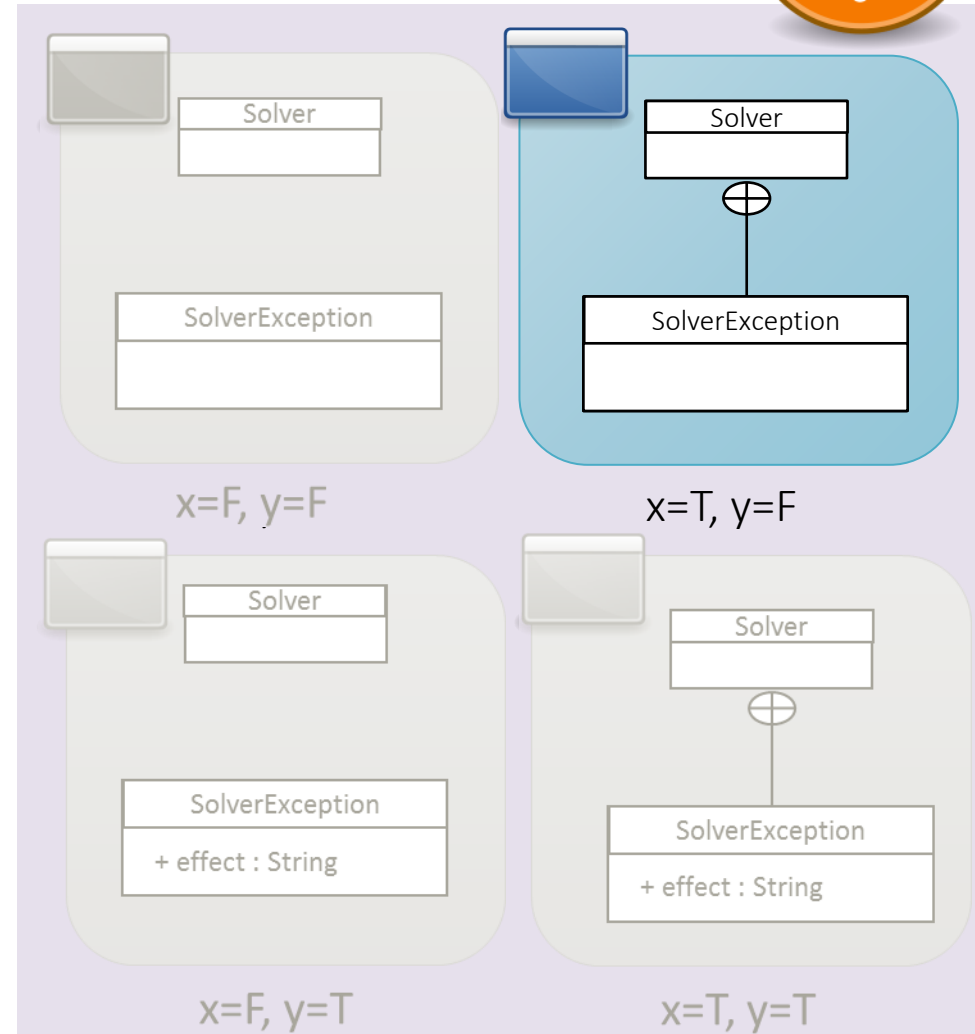
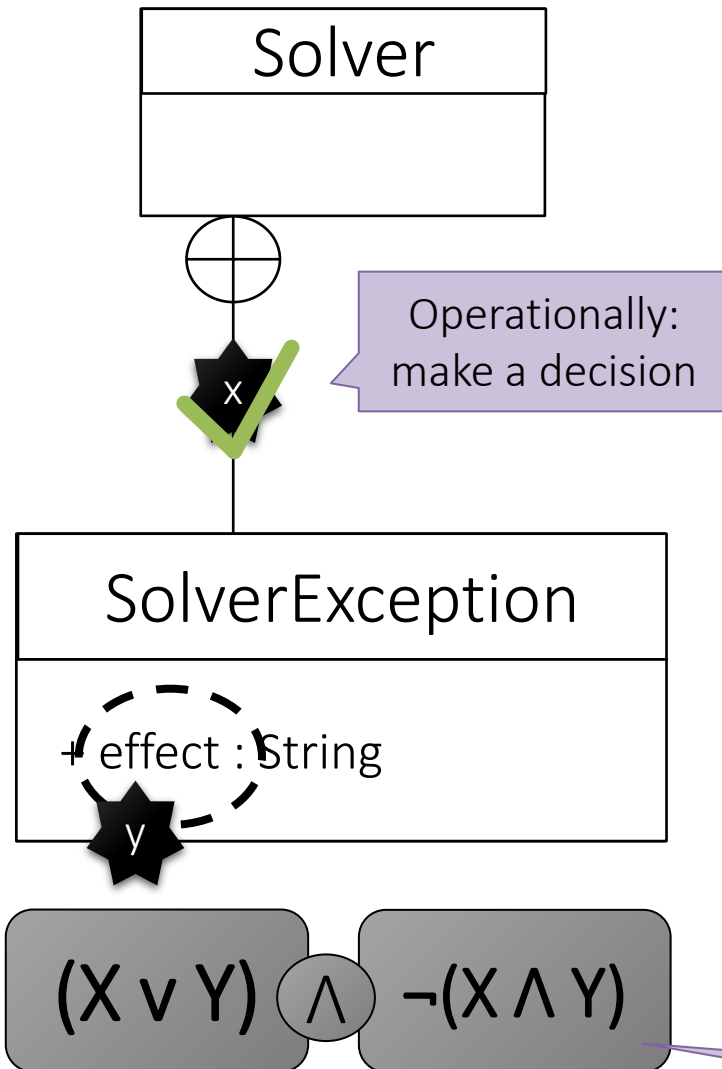


Outline



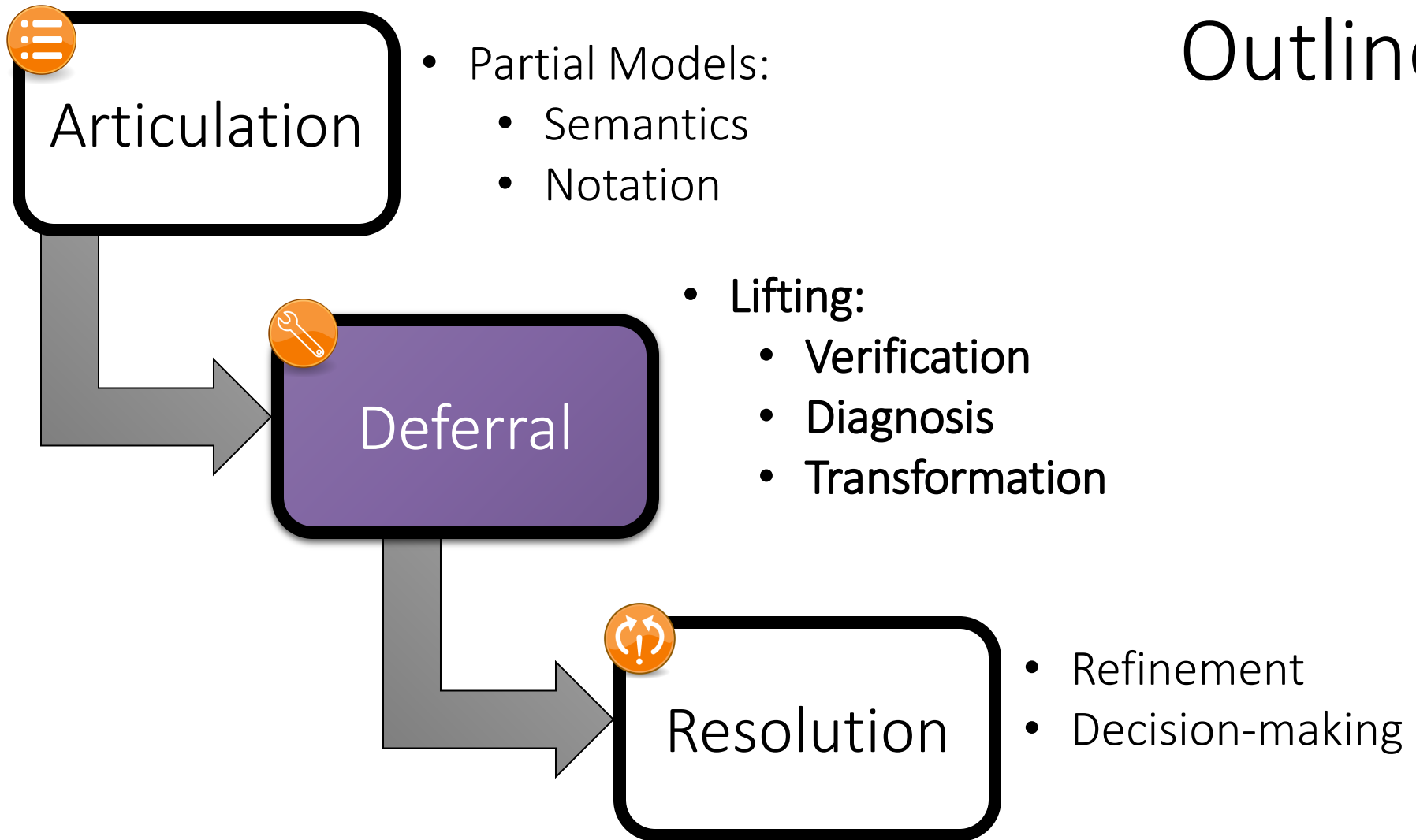
- Methodology and Tool Support
- Worked-out Examples
- Conclusion, Future Work

Refinement: Reduce the Set



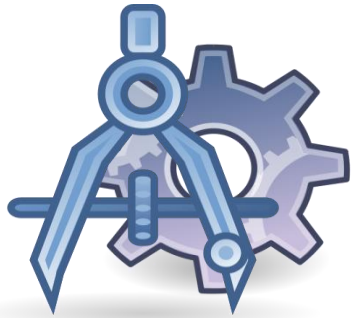
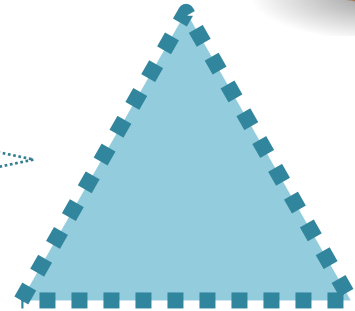
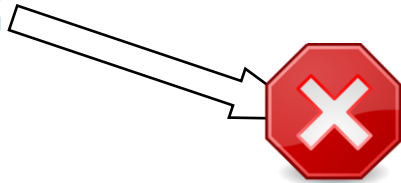
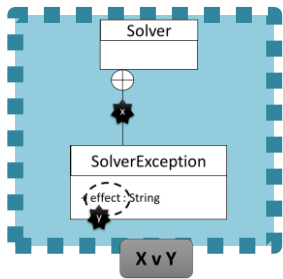
Declaratively: with a property

Outline

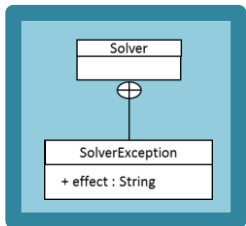


- Methodology and Tool Support
- Worked-out Examples
- Conclusion, Future Work

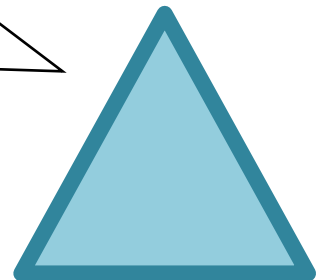
Deferring Uncertainty Resolution



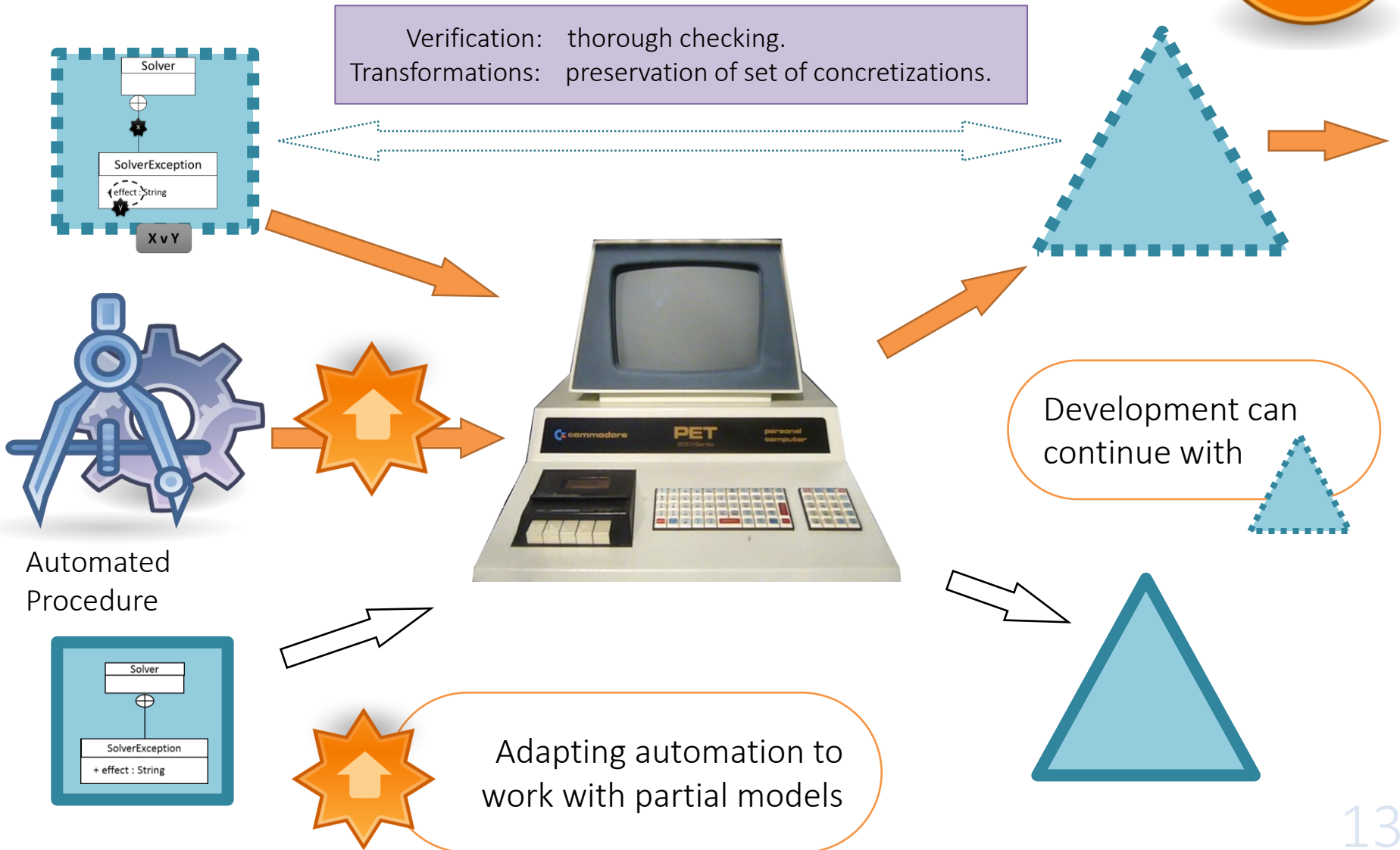
Automated
Procedure



DOES NOT COMPUTE
MAKE DECISIONS FIRST



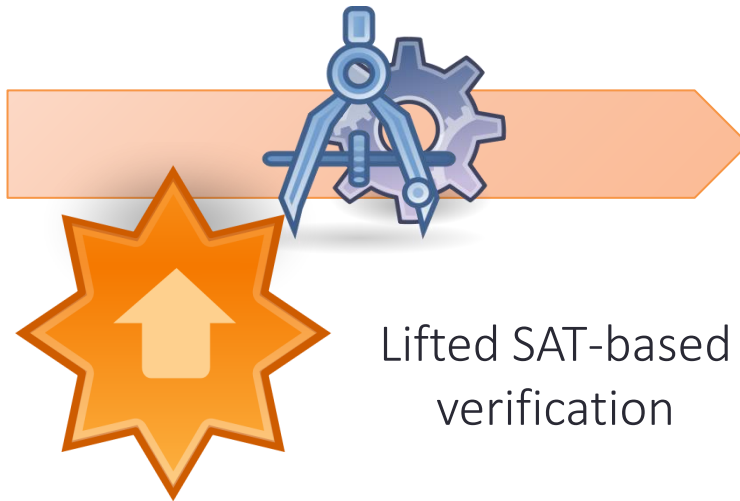
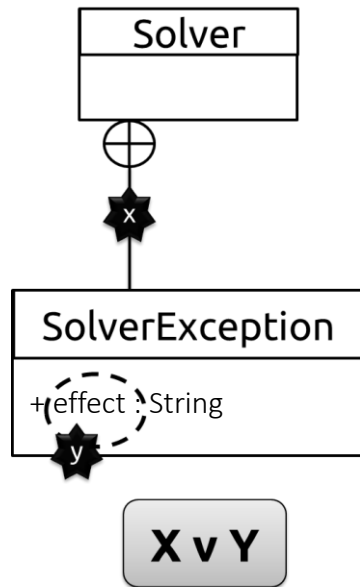
Deferral Through “Lifting”



Lifting Verification

Example property:

“Every inner class has at least one attribute”



Property holds for...



...all
concretizations



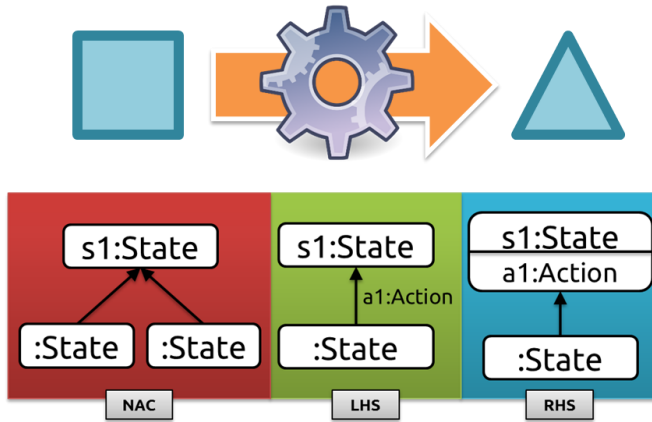
...some but
not all



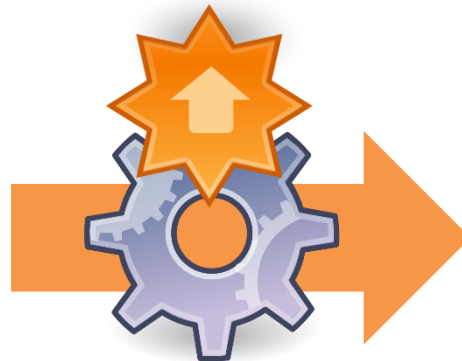
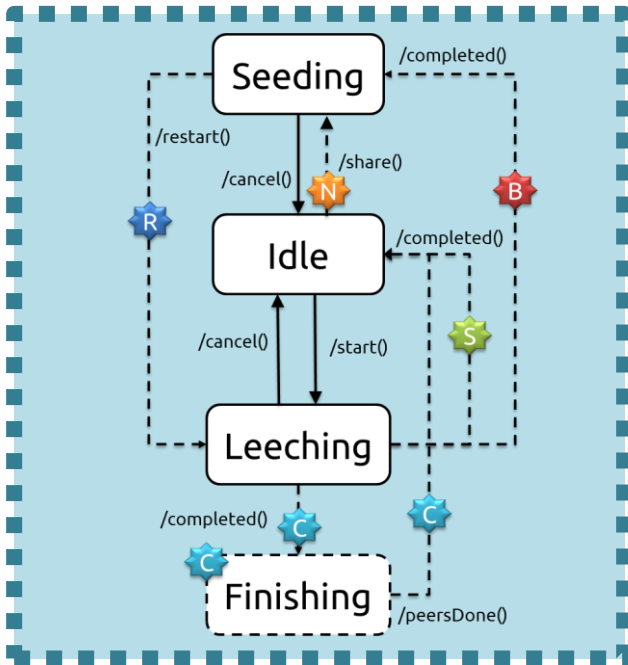
...none

- Applies directly to the partial model
- Does **not** enumerate concretizations
- Computes result using three-valued logic

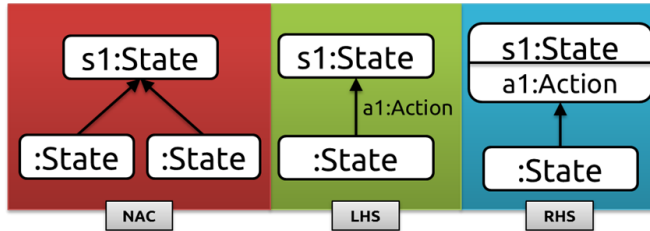
Lifting Transformations



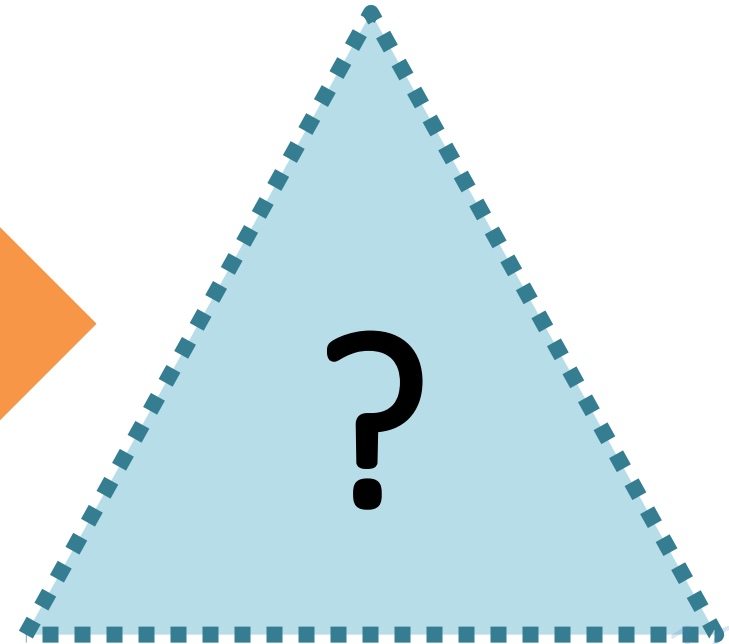
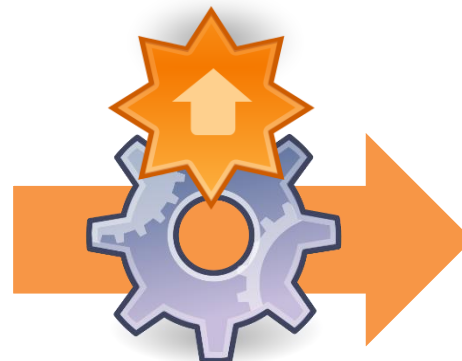
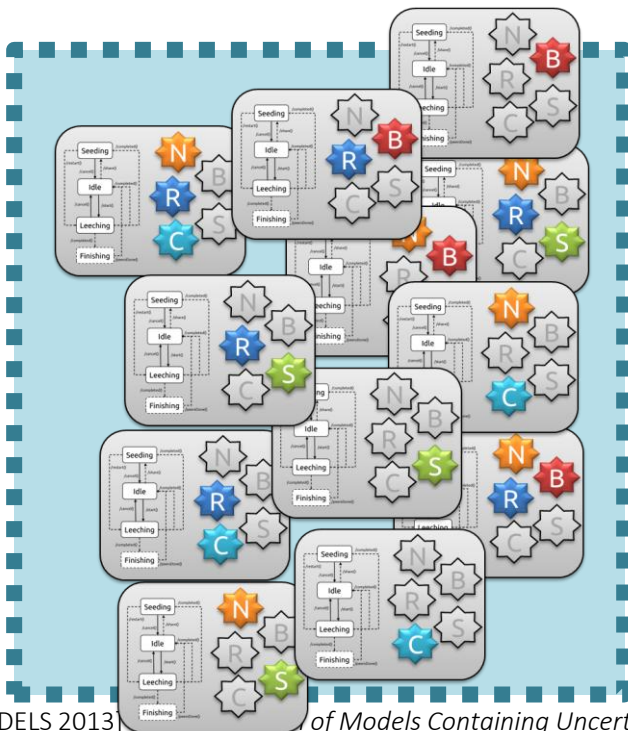
- ? Can users initiate seeding? 
- ? Can users restart downloads? 
- ? What happens when a download is completed?
 - "Benevolent" 
 - "Selfish" 
 - "Compromise" 



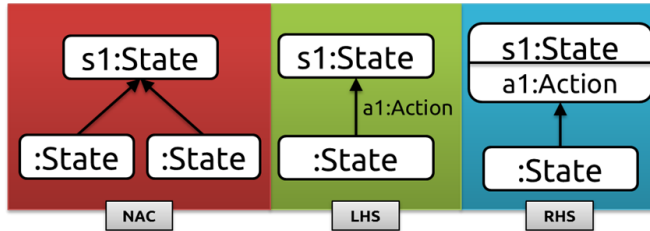
Lifting Transformations



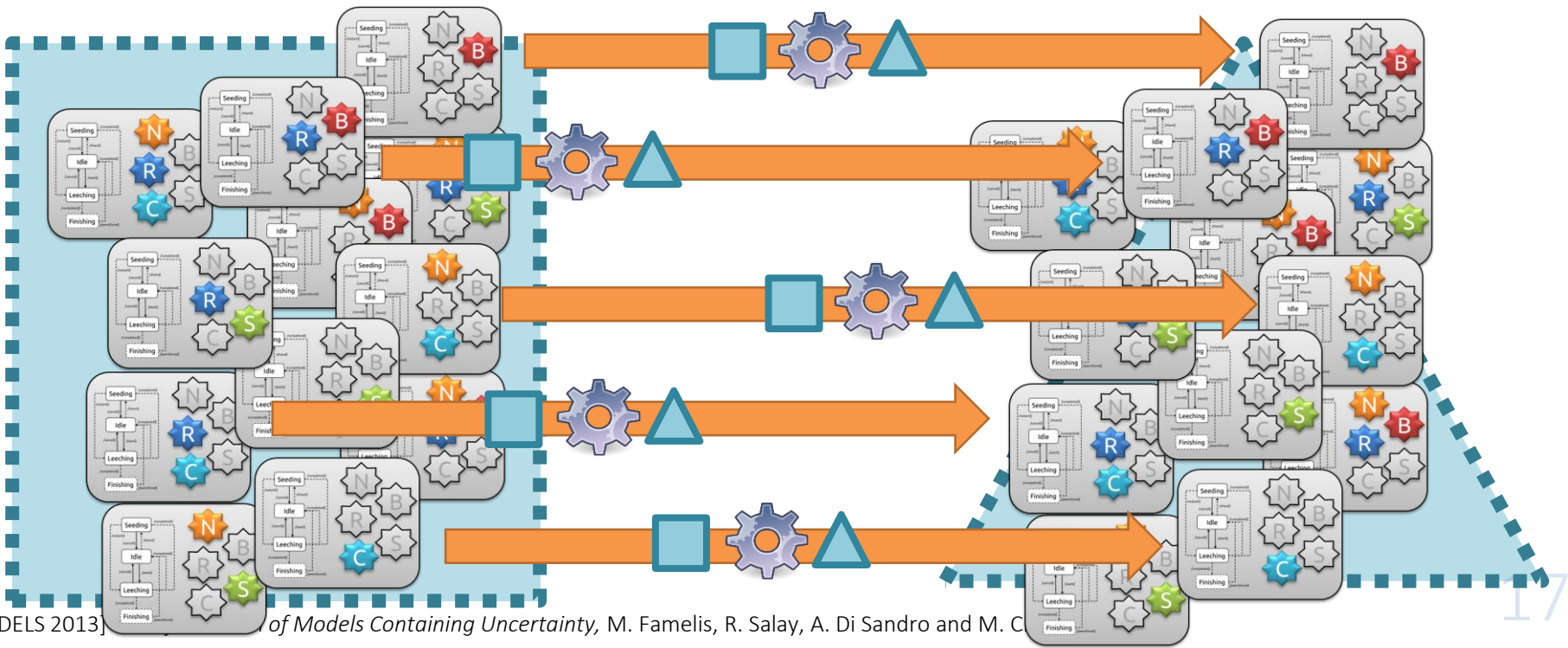
- ? Can users initiate seeding? H
- ? Can users restart downloads? R
- ? What happens when a download is completed?
 - "Benevolent" B
 - "Selfish" S
 - "Compromise" C



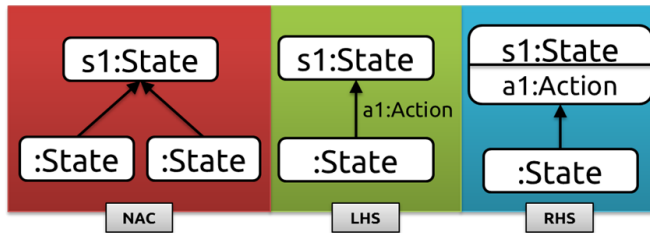
Lifting Transformations



- ? Can users initiate seeding? **H**
- ? Can users restart downloads? **R**
- ? What happens when a download is completed?
 - "Benevolent" **B**
 - "Selfish" **S**
 - "Compromise" **C**



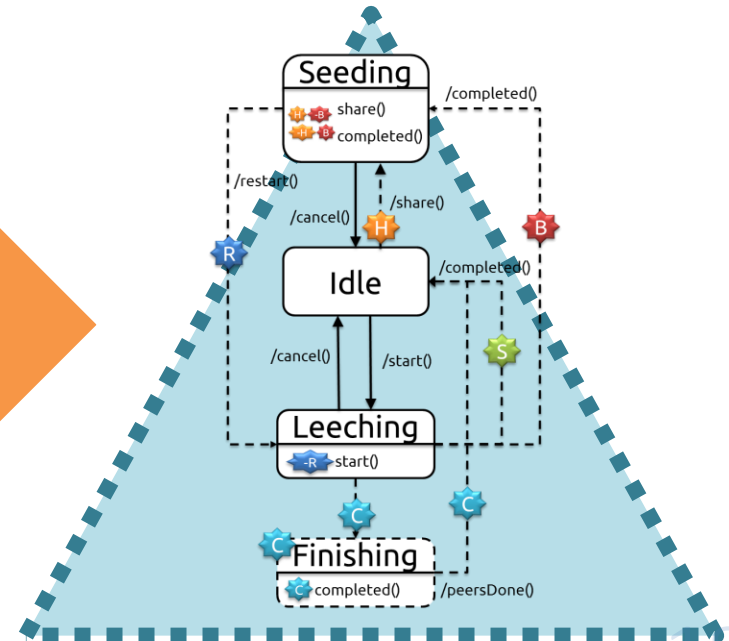
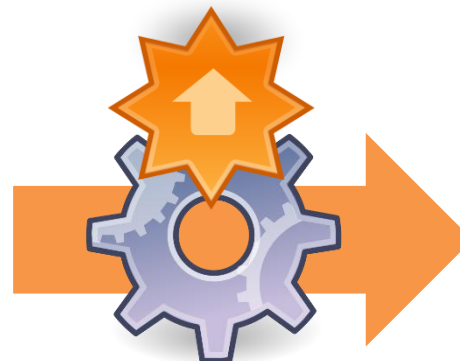
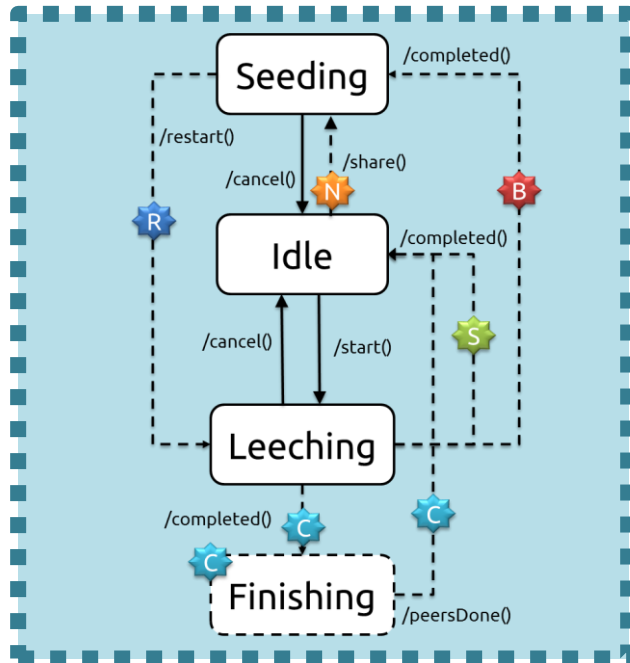
Lifting Transformations



Step 1:
Determine applicability

Step 2:
Transform graph

Step 3:
Transform constraints



Design decisions
not affected

Neither is the
transformation!

- ? Can users initiate seeding?
- ? Can users restart downloads?
- ? What happens when a download is completed?
 - "Benevolent"
 - "Selfish"
 - "Compromise"

Outline



Articulation

- Partial Models:
 - Semantics
 - Notation



Deferral

- Lifting:
 - Verification
 - Diagnosis
 - Transformation

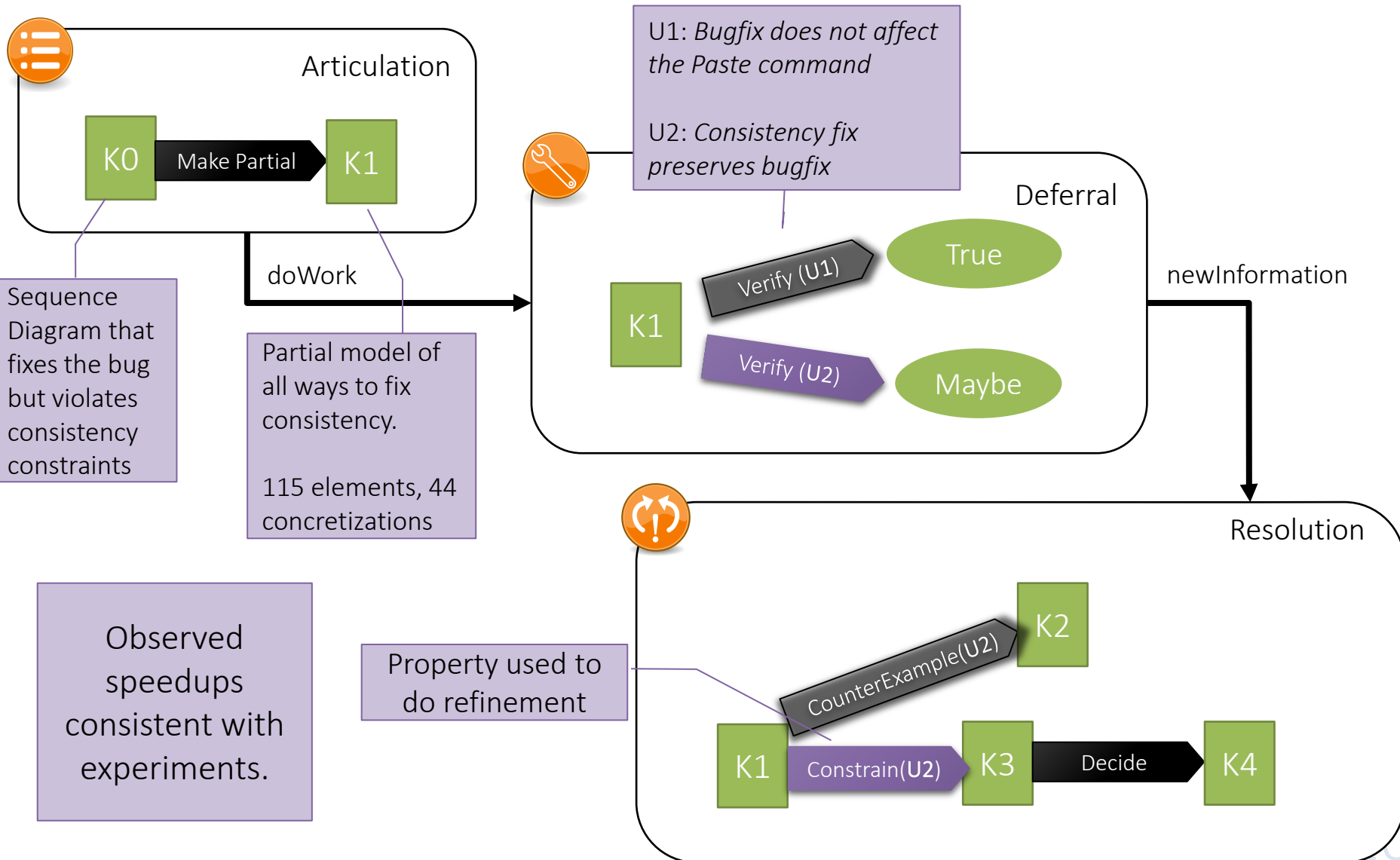


Resolution

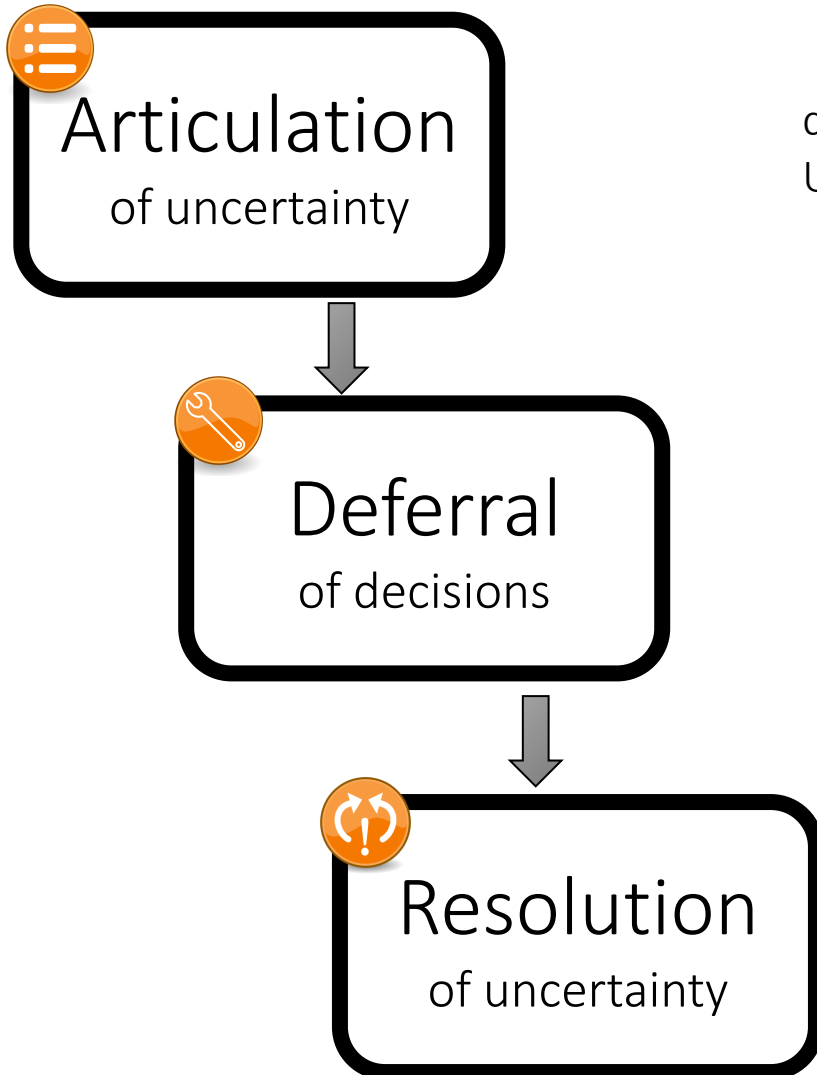
- Refinement
- Decision-making

- **Methodology and Tool Support**
- Worked-out Examples
- Conclusion, Future Work

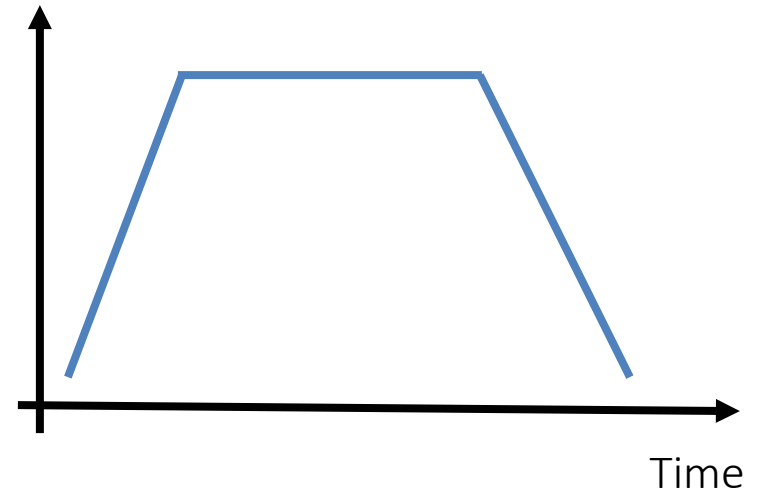
UMLet Bug #10



Uncertainty Lifecycle Management

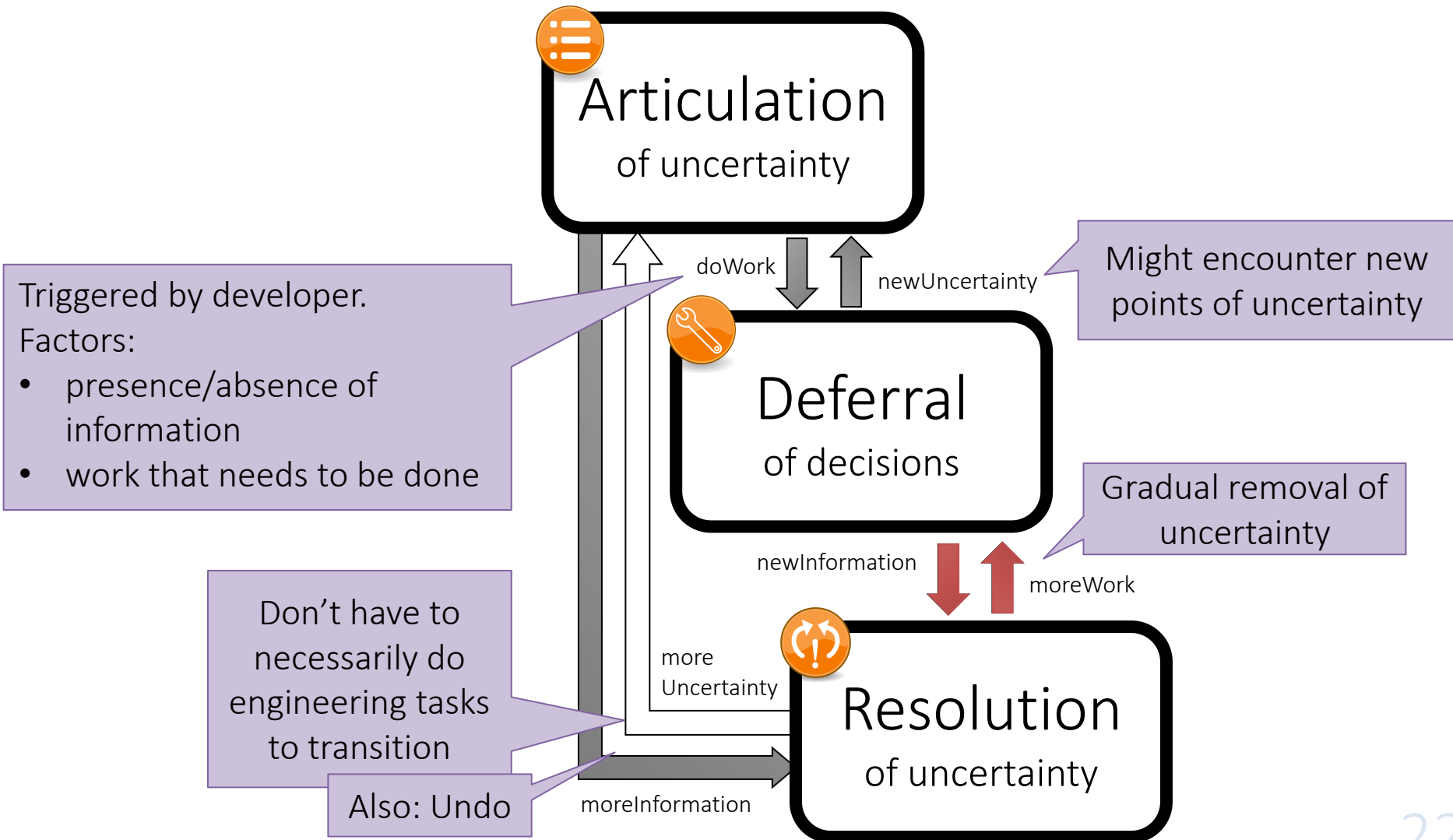


degree of
Uncertainty

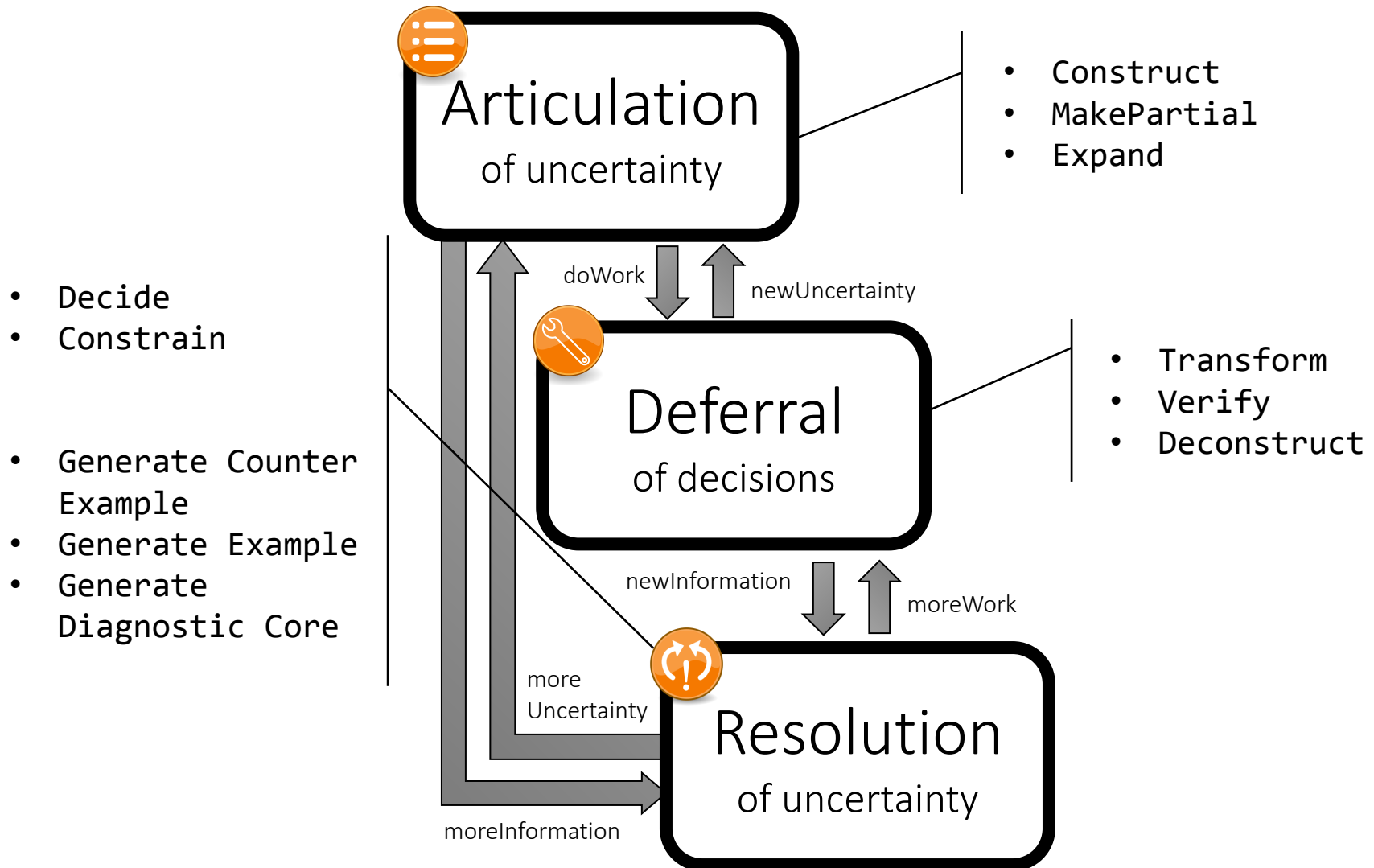


- Degree of uncertainty: size of the set of concretizations
- Ultimately, a single concrete model: all uncertainty resolved

Design-Time Uncertainty Management (DeTUM) model



Uncertainty Management Operators



Example Operator Specification

Name	Construct
Description	
Inputs	
Outputs	
Usage context	
Preconditions	
Postconditions	
Limitations	
Implementation	



MU-MMINT

(pronounced “moomin”)

Partial Model
Editor

Decision Tree
Editor

Dashboard &
Traceability

Verification &
Refinement
Support

Lifted
Transformations



MMINT: “Model Management INteractive”

Eclipse

Z3 SMT Solver

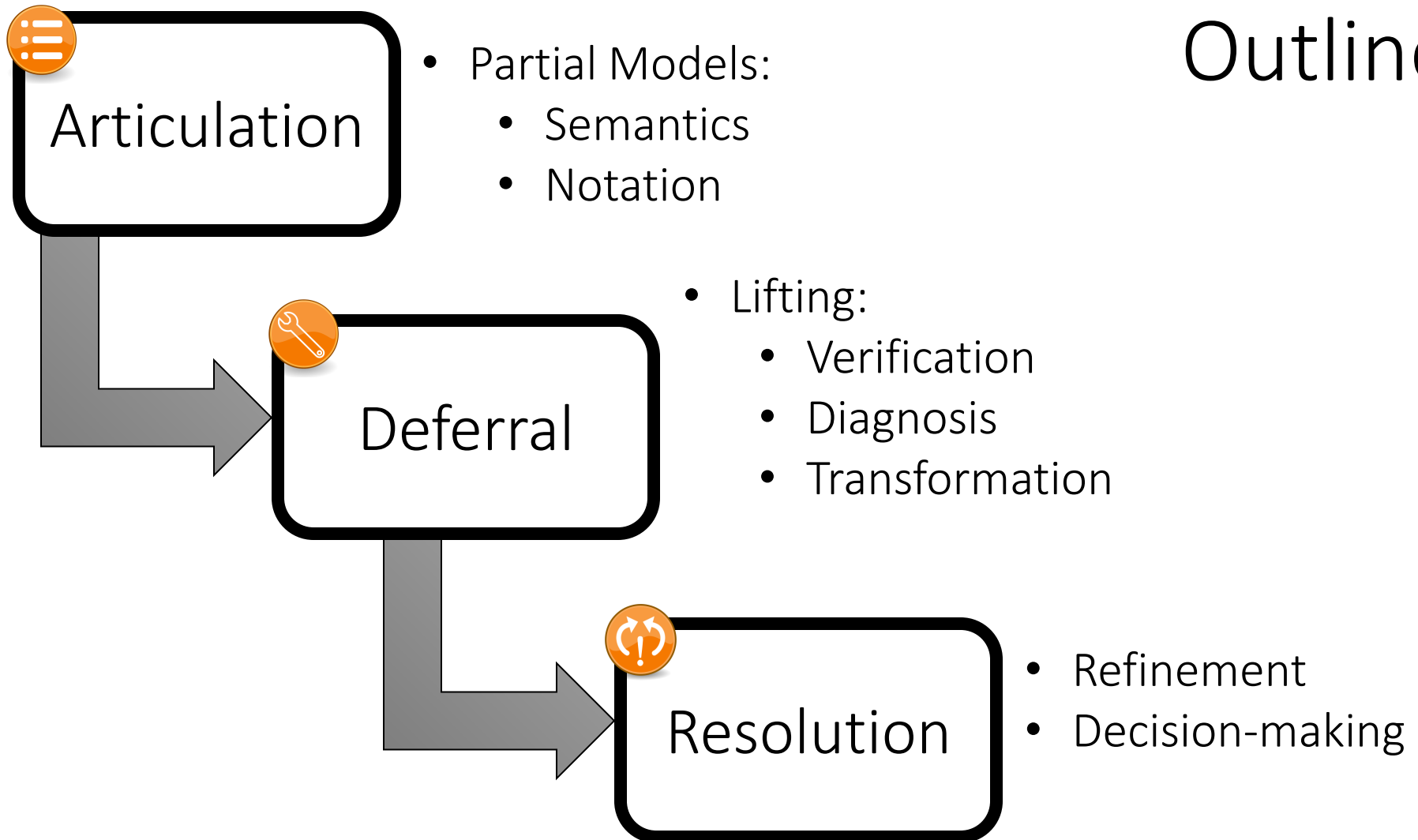
Henshin Graph
Transformation Engine

MU-MMINT demo: <https://youtu.be/kAWUm-iFatM>

MMINT demo: <https://youtu.be/7B7YuV-Jvrc>

Available at <https://github.com/adisandro/MMINT>

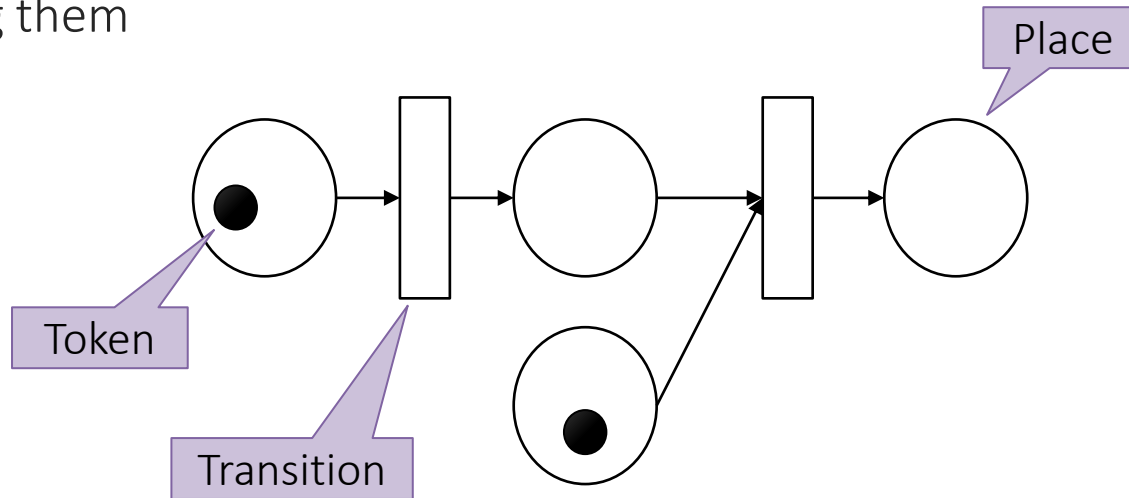
Outline



- Methodology and Tool Support
- **Worked-out Examples**
- Conclusion, Future Work

Metamodel to Relational Schema

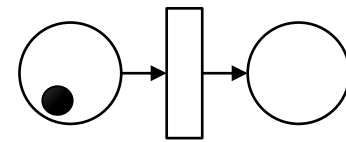
Scenario: create a metamodel for Petri nets, then create a schema for storing them



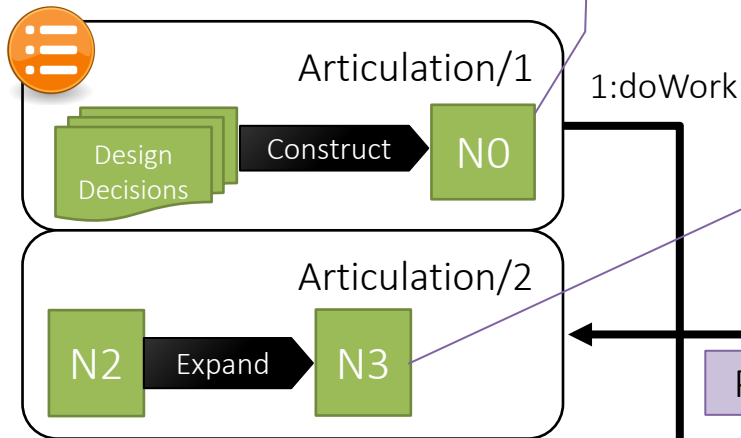
- Atlas Metamodel zoo: 8 different designs / 5 design decisions
- Partial model NO created using MU-MMINT
 - Demo partial model editor
 - Demo Verification and Diagnosis
 - Demo Transformation



Petri Net Metamodel



76 elements, 18 concretizations



Additional uncertainty:
*Which domain-specific extensions
should the metamodel support?*

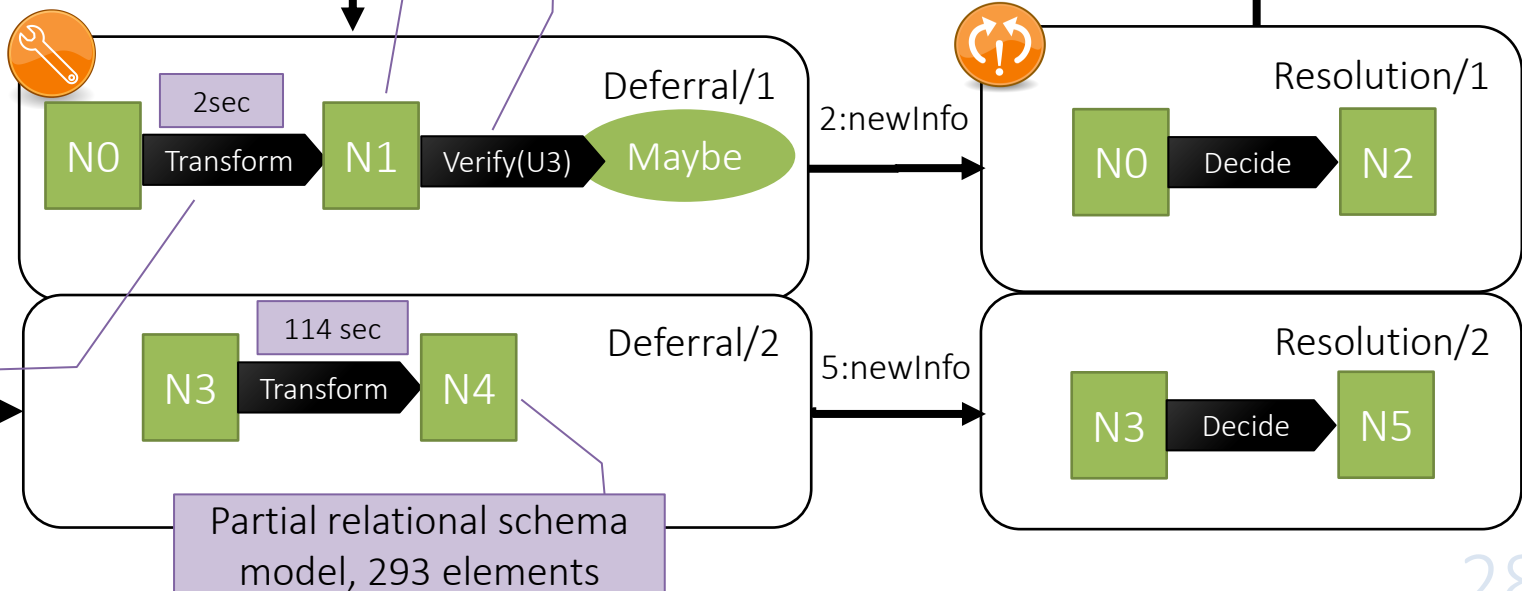
New partial model with 117
elements, 360 concretizations.

3:moreUncertainty

Partial relational schema model, 192 elements

U3: *Diagram element locations are stored*

4:doWork



Object-
Relational
Mapping
transformation
with 5 layered
Henshin rules

Partial relational schema
model, 293 elements

Lessons Learned from Worked Examples

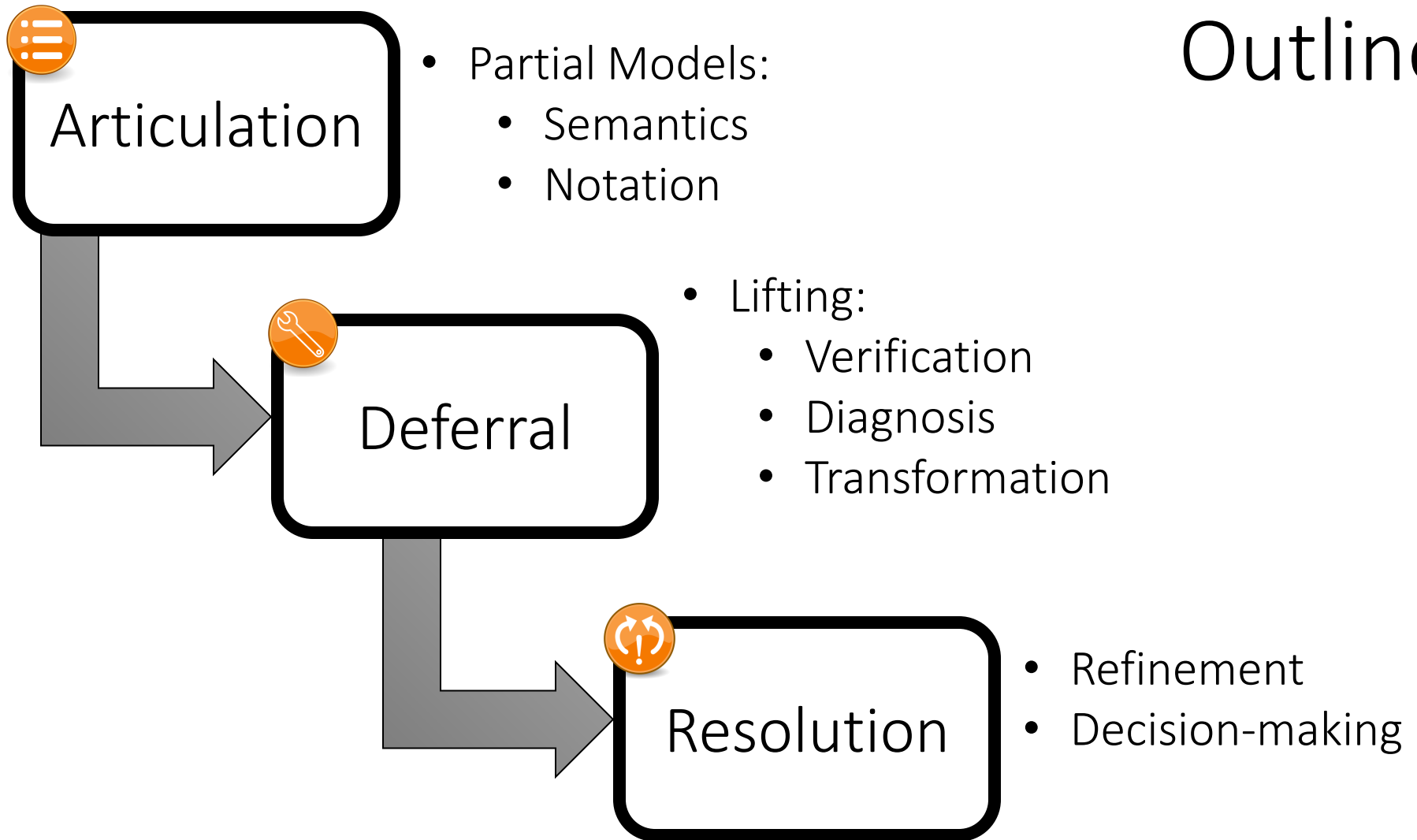
Must better support *Articulation* with automation

Stages of DETUM not rigid (Verification/Diagnosis)

May formula makes engineering of lifting hard

Changing modality of properties may be more appropriate
response to bad verification result

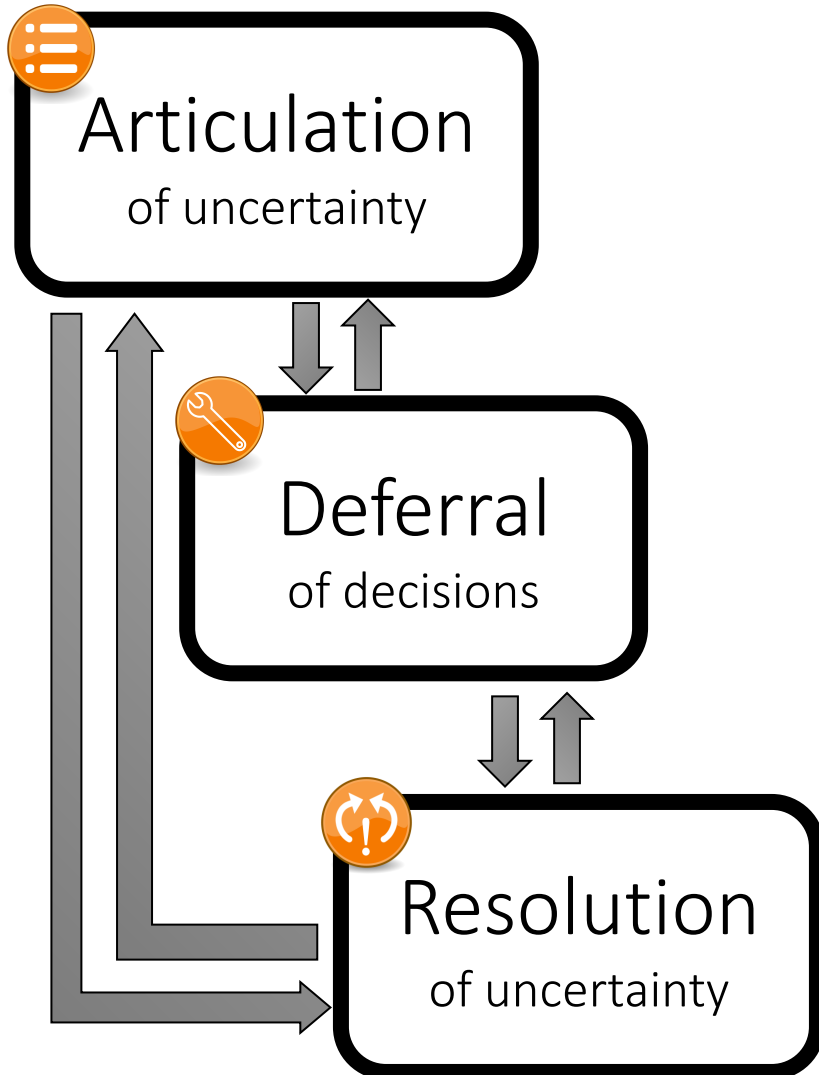
Outline



- Methodology and Tool Support
- Worked-out Examples
- Conclusion, Future Work

Managing of Design-Time Uncertainty

Defer resolution of uncertainty but incorporate uncertainty handling into the development process to allow progress



- Partial Models:
 - Semantics
 - Notation
- Lifting:
 - Verification
 - Diagnosis
 - Transformation
- Refinement
- Decision-making
- DETUM model
- Uncertainty Management Ops
- MU-MMINT

Future Work

- Relax underlying assumptions

 - Design decisions known; alternatives elicited

- Better support uncertainty articulation

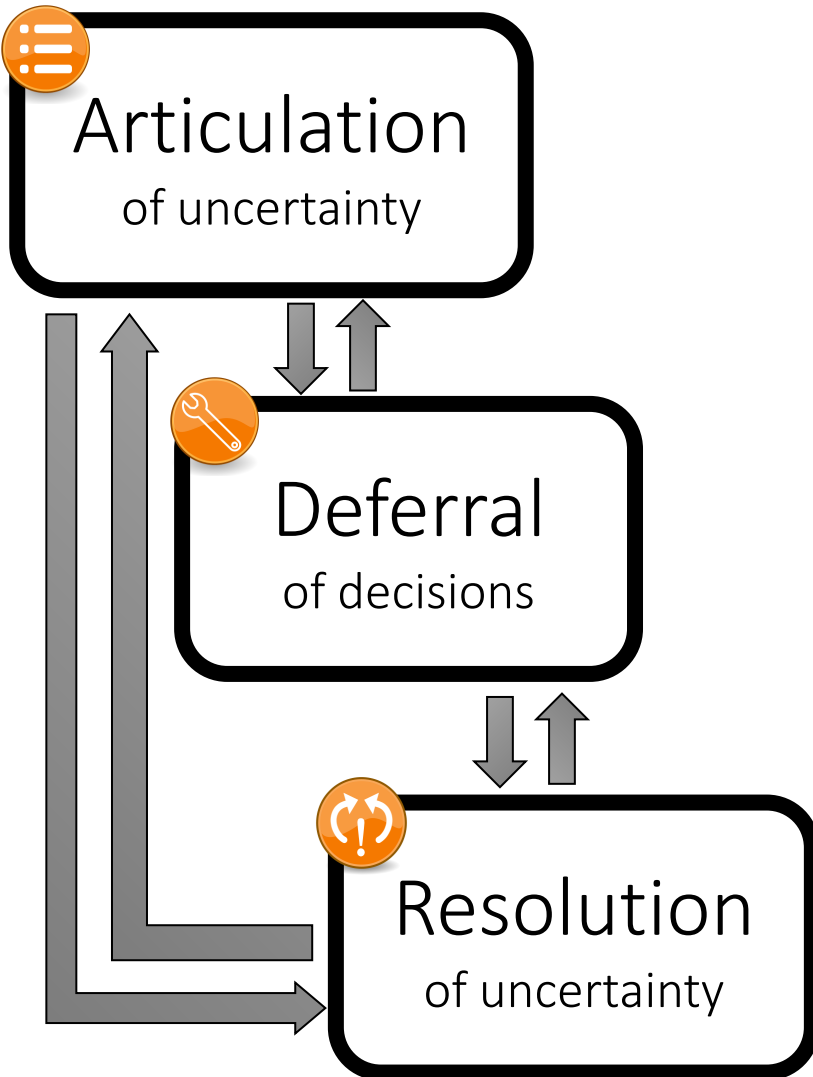
 - Leverage development context

- Systematically elicit design options

- Combine with existing methodologies (e.g. Scrum, Kanban)

Managing of Design-Time Uncertainty

Michalis Famelis, Marsha Chechik



- Partial Models:
 - Semantics
 - Notation
- Lifting:
 - Verification
 - Diagnosis
 - Transformation
- Refinement
- Decision-making
- Methodology and Tool Support
- Worked-out Examples
- Discussion, Future Work

Questions?