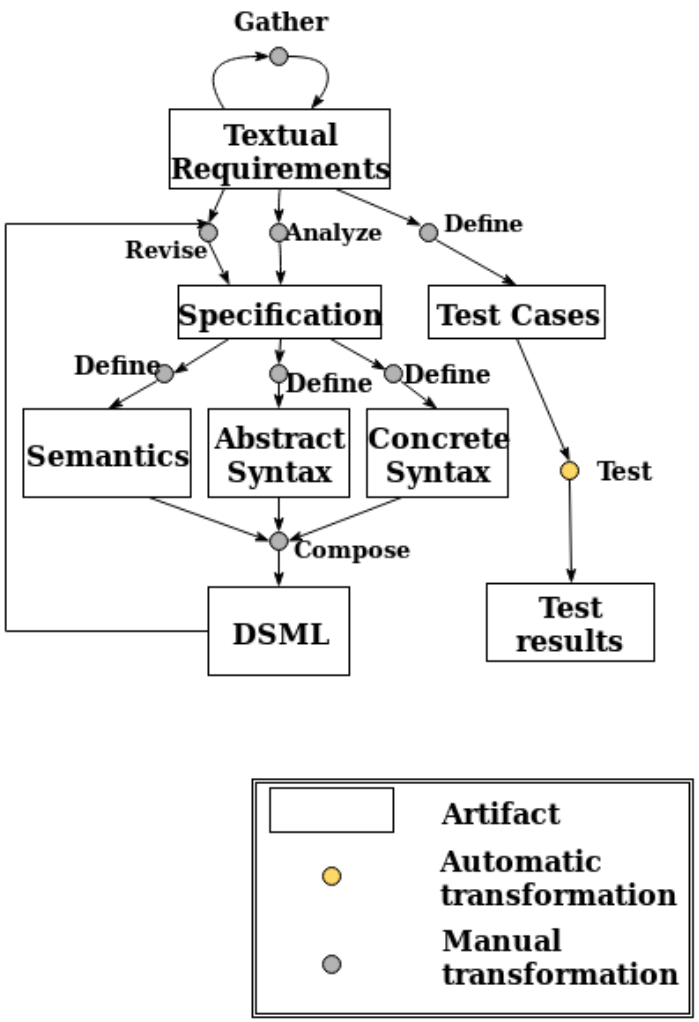
Sketch-based Metamodel Construction

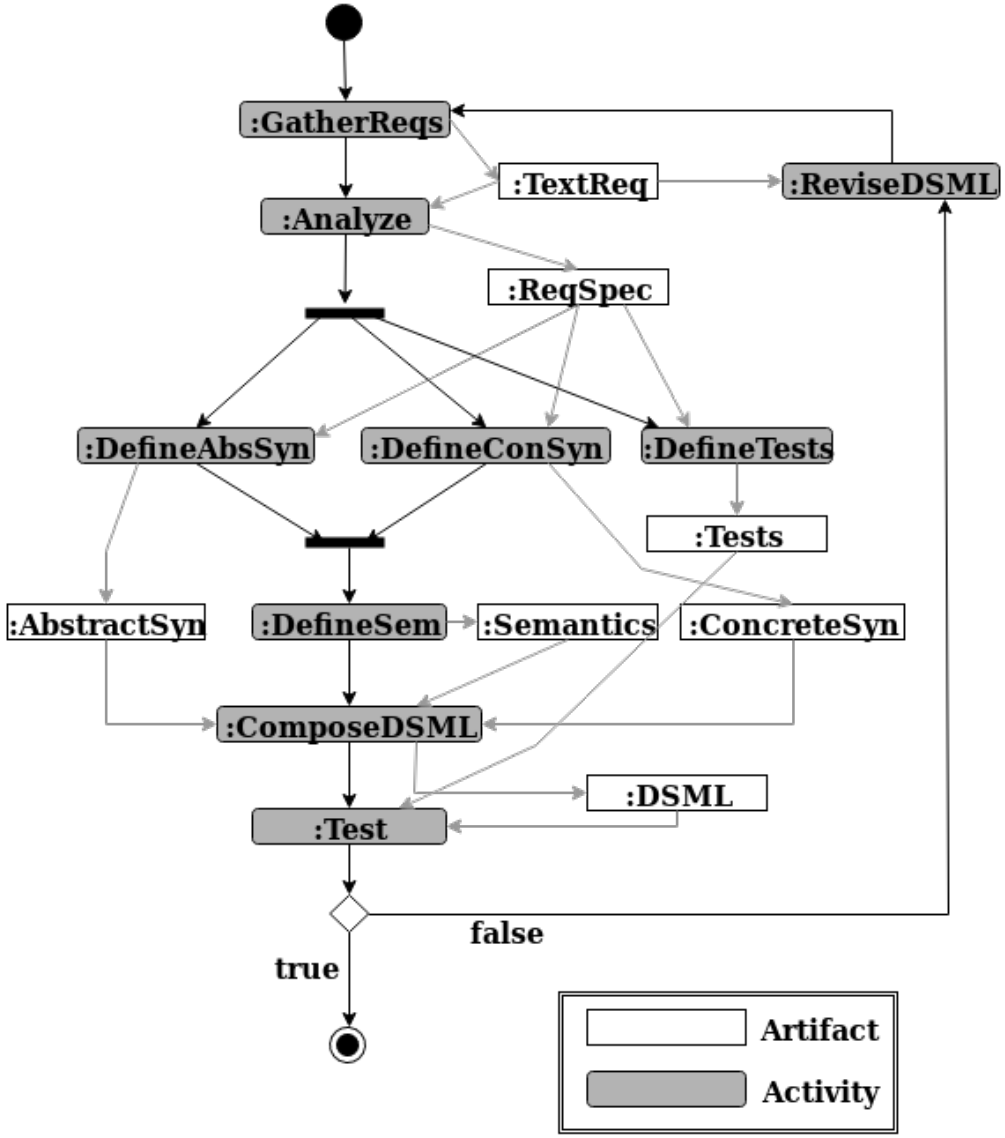
Research Internship II

Lucas Heer
lucas.heer@student.uantwerpen.be

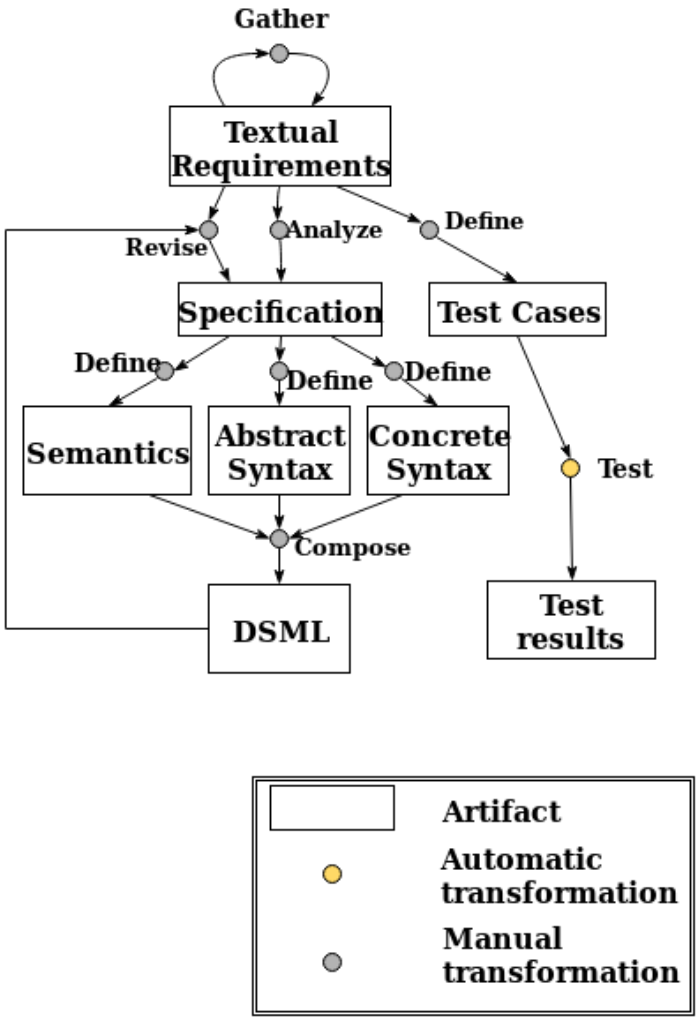
31.01.2018

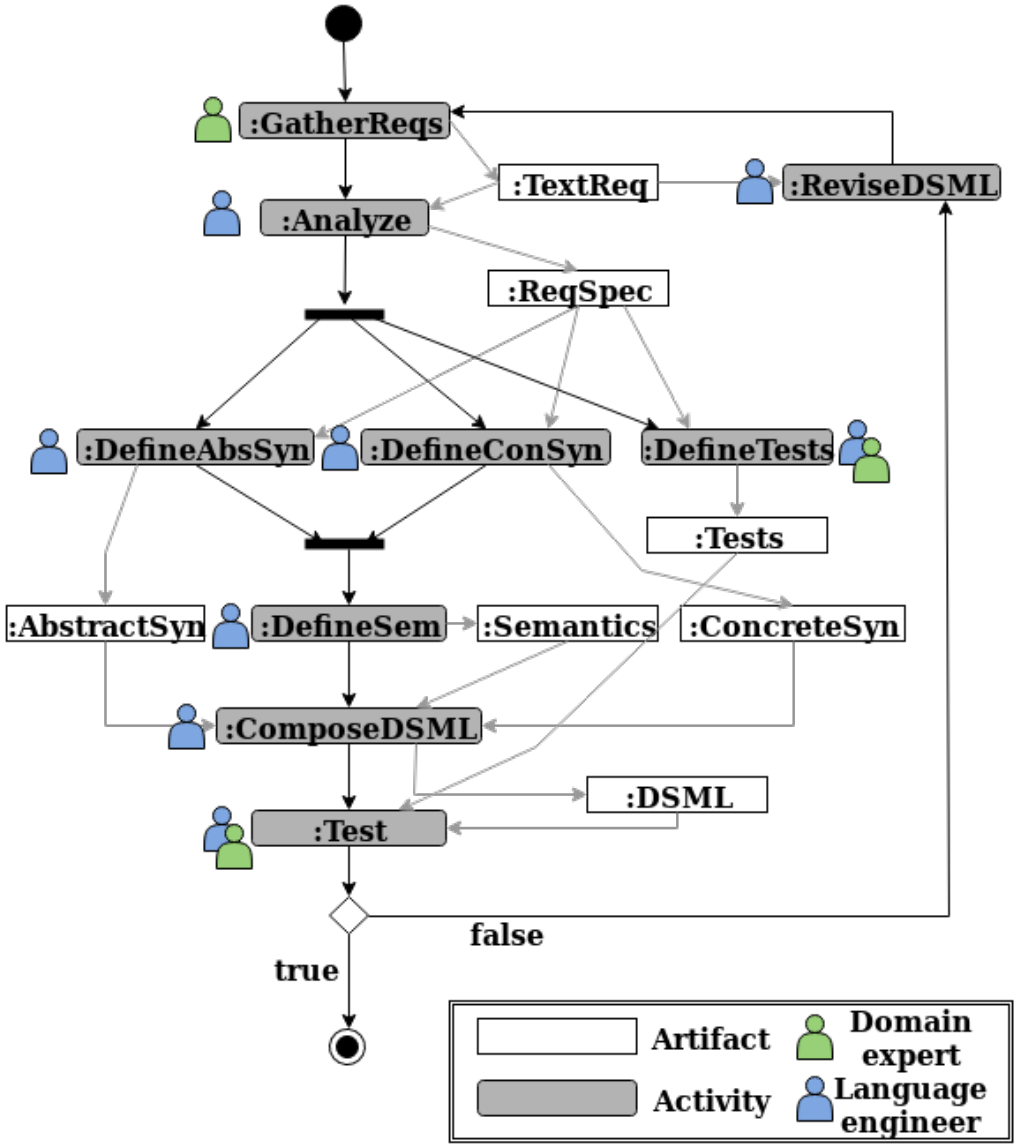
Motivation


Gather
Textual Requirements
Revise
Analyze
Define
Specification
Test Cases
Define
Define
Define
Semantics
Abstract Syntax
Concrete Syntax
Test
Compose
DSML
Test results
Artifact
Automatic transformation
Manual transformation


:GatherReqs
:TextReq
:ReviseDSML
:Analyze
:ReqSpec
:DefineAbsSyn
:DefineConSyn
:DefineTests
:Tests
:AbstractSyn
:DefineSem
:Semantics
:ConcreteSyn
:ComposeDSML
:DSML
:Test
false
true
Artifact
Activity

Motivation

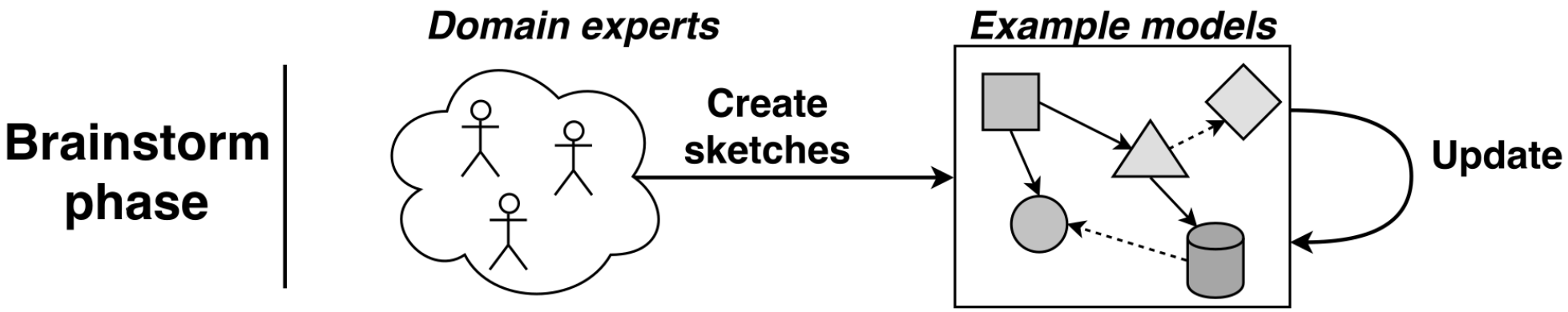

Gather
Textual Requirements
Revise
Analyze
Define
Specification
Test Cases
Define
Define
Define
Semantics
Abstract Syntax
Concrete Syntax
Test
Compose
DSML
Test results
Artifact
Automatic transformation
Manual transformation


:GatherReqs
:TextReq
:ReviseDSML
:Analyze
:ReqSpec
:DefineAbsSyn
:DefineConSyn
:DefineTests
:Tests
:AbstractSyn
:DefineSem
:Semantics
:ConcreteSyn
:ComposeDSML
:DSML
:Test
false
true
Artifact
Activity
Domain expert
Language engineer

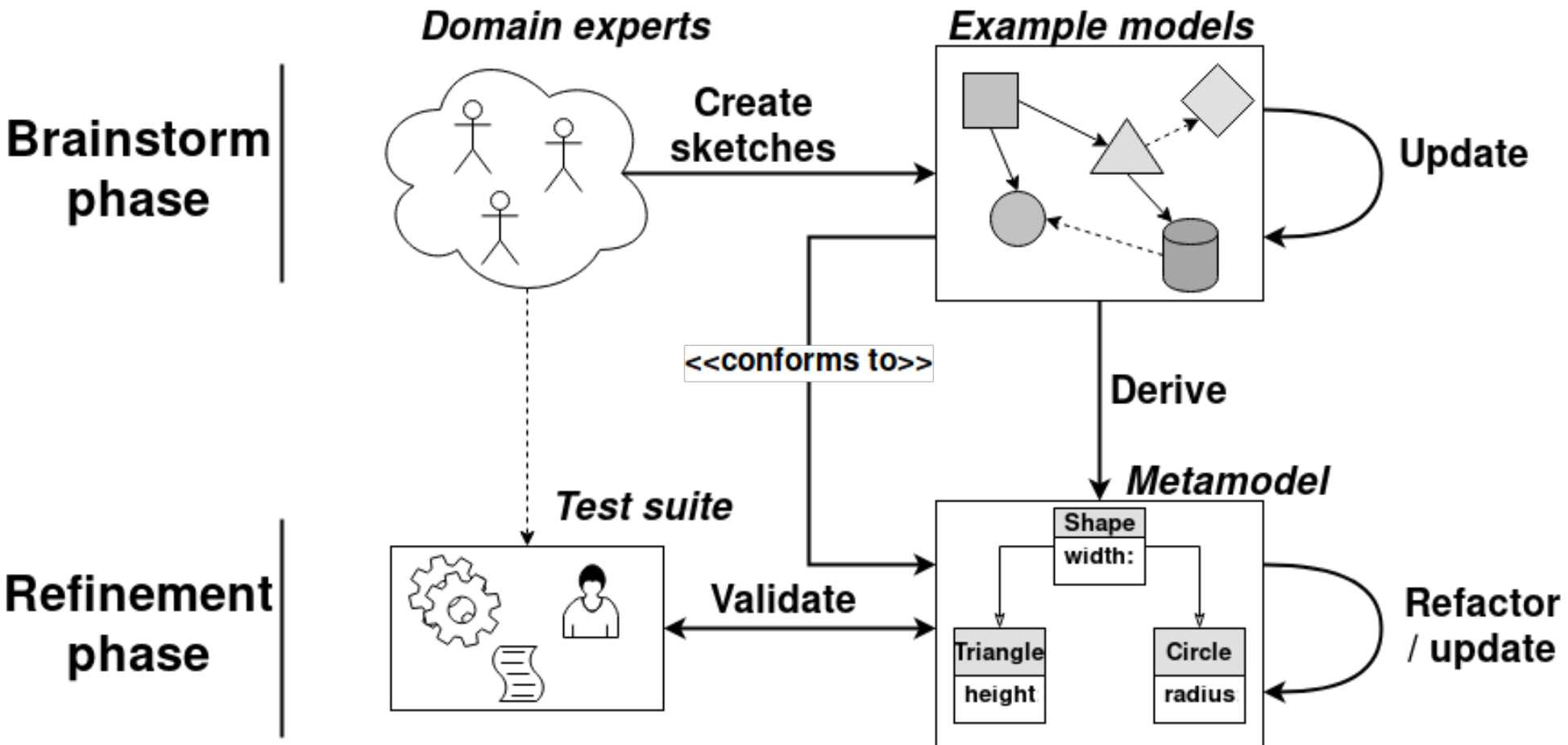
Solution

What if we start from instance models?

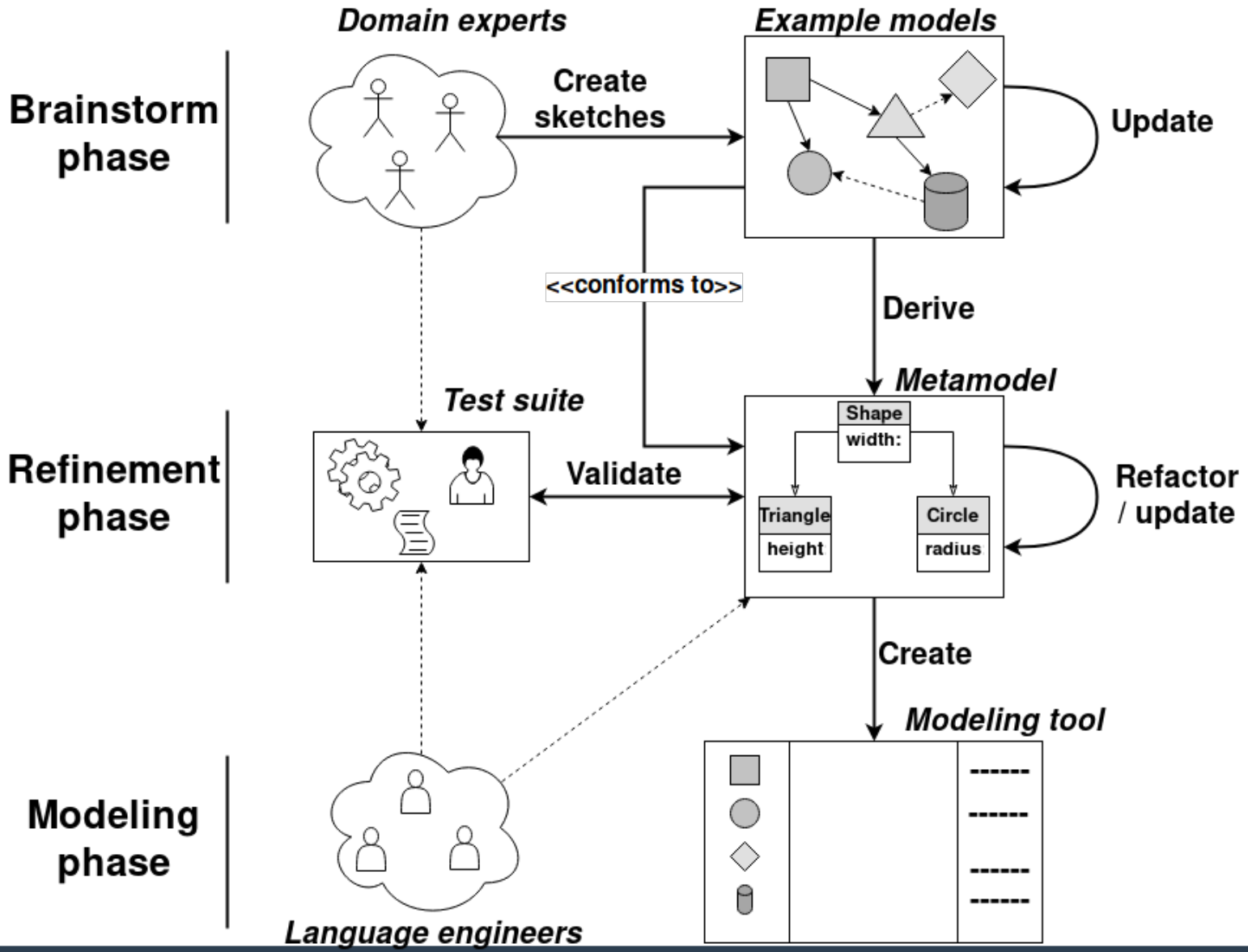
Solution


Brainstorm phase
Domain experts
Create sketches
Example models
Update

Solution


Brainstorm phase
Refinement phase
Domain experts
Create sketches
Example models
Update
<<conforms to>>
Derive
Metamodel
Test suite
Validate
Shape
width:
Triangle
height
Circle
radius
Refactor / update

Solution


Domain experts
Example models
Brainstorm phase
Create sketches
Update
<<conforms to>>
Derive
Metamodel
Test suite
Refinement phase
Validate
Shape
width:
Triangle
height
Circle
radius
Refactor / update
Create
Modeling tool
Modeling phase
Language engineers

Research goal

- Examine current state of research
- Identify challenges
- Analyze existing solutions using concrete test cases
- Compare solutions and find open questions

Existing approaches

Name	Year	Implemented	Readily available
Scribbler	2013	Yes (Standalone)	No
MLCBD	2012	Yes (Visio plugin)	No
metaBup	2013-2017	Yes (EMF plugin)	Yes
FlexiSketch	2012-2015	Yes (Android App)	Yes
Model Workbench	2013-2014	Yes (Standalone)	No

Resources (1)

Scribbler:

[1] Vogel, Martin, Tim Warnecke, and Christian Bartelt. "Scribbler: From Collaborative Sketching to Formal Domain Specific Models and Back Again." Demos/Posters/StudentResearch@ MoDELS. 2013.

[2] Vogel, Martin, et al. "Scribbler - Drawing models in a creative and collaborative environment: From hand-drawn sketches to domain specific models and vice versa." Proceedings of the Fifteenth Australasian User Interface Conference-Volume 150. Australian Computer Society, Inc., 2014.

MLCBD:

[1] Cho, Hyun, Jeff Gray, and Eugene Syriani. "Creating visual domain-specific modeling languages from end-user demonstration." Modeling in Software Engineering (MISE), 2012 ICSE Workshop on. IEEE, 2012.

metaBup:

[1] López-Fernández, Jesús J. "An agile process for the example-driven development of modelling languages and environments." (PhD Thesis, 2017).

[2] López-Fernández, Jesús J., et al. "Example-driven meta-model development." Software & Systems Modeling Vol. 14 Issue 4 (2015): 1323-1347.

Resources (2)

FlexiSketch:

[1] Wüest, Dustin, Norbert Seyff, and Martin Glinz. "Flexisketch: A mobile sketching tool for software modeling." International Conference on Mobile Computing, Applications, and Services. Springer, Berlin, Heidelberg, 2012.

[2] Wüest, Dustin, Norbert Seyff, and Martin Glinz. "Flexisketch team: Collaborative sketching and notation creation on the fly." Software Engineering (ICSE), 2015 IEEE/ACM 37th IEEE International Conference on. Vol. 2. IEEE, 2015.

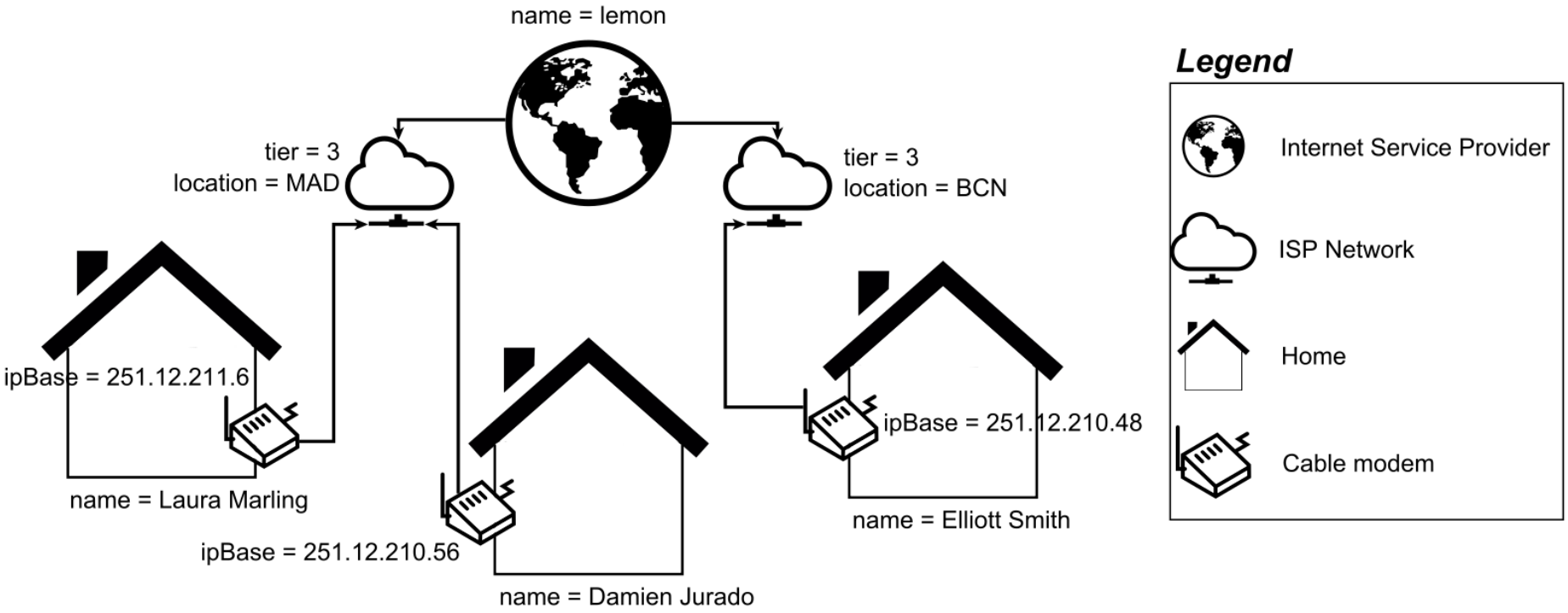
Model Workbench:

[1] Roth, Bastian, Matthias Jahn, and Stefan Jablonski. "On the way of bottom-up designing textual domain-specific modelling languages." Proceedings of the 2013 ACM workshop on Domain-specific modeling. ACM, 2013.

[2] Roth, B., M. Jahn, and S. Jablonski. "A Method for Directly Deriving a Concise Meta Model from Example Models." Proceedings of PATTERNS 2013 (2013): 52-58.

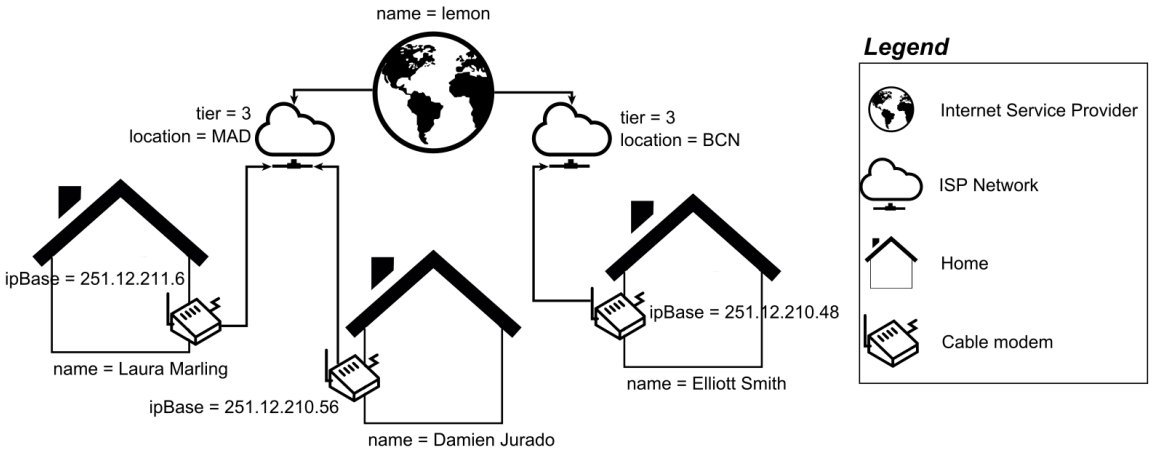
Example: metaBup

Step 1: Draw example models



Example: metaBup

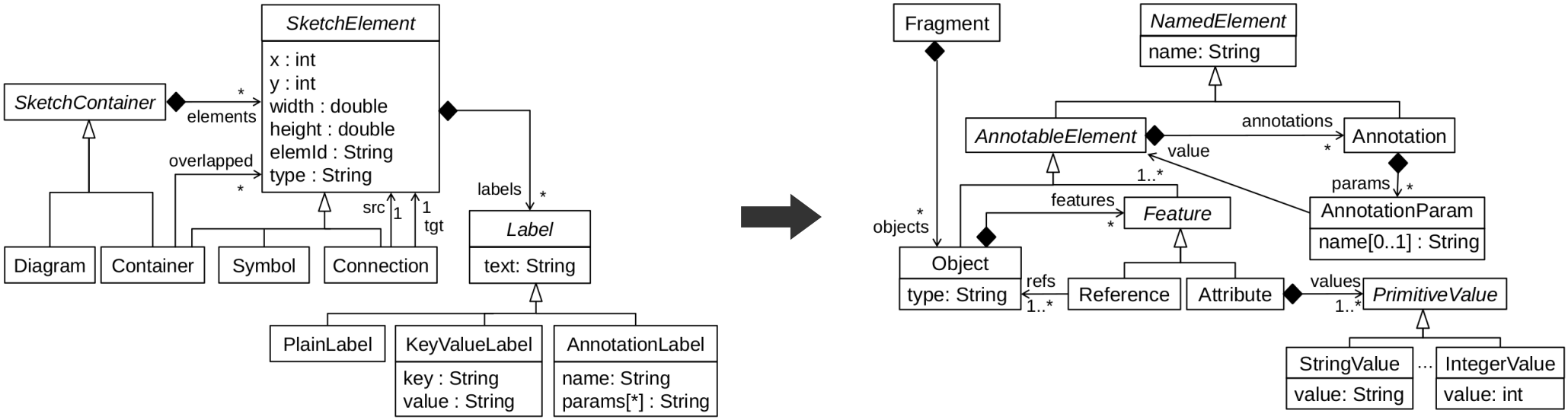
Step 2: Transform to text model




```
shell importedFragment fragment fragment1 {
    Home_1 : Home {
        attr name = "Elliott Smith"
        @ overlapping ref modem = CableModem_3
        }
    InternetServiceProvider_1 : InternetServiceProvider {
        attr name = "lemon"
        ref infrastructure = ISPNetwork_1, ISPNetwork_2
    }
    Home_2 : Home {
        attr name = "Damien Jurado"
        @ overlapping ref modem = CableModem_2
    }
    Home_3 : Home {
        attr name = "Laura Marlin"
        @ overlapping ref modem = CableModem_1
    }
    CableModem_1 : CableModem {
        attr ipBase =  "251.12.211.6"
        ref isp = ISPNetwork_1
    }
    CableModem_2 : CableModem {
        attr ipBase =  "251.12.210.56"
        ref isp = ISPNetwork_1
    }
    CableModem_3 : CableModem {
        attr ipBase =  "251.12.210.48"
        ref isp = ISPNetwork_2
    }
    ISPNetwork_1 : ISPNetwork {
        attr tier = 3
        attr location = "MAD"
    }
    ISPNetwork_2 : ISPNetwork {
        attr tier = 4
        attr location = "BCN"
    }
}
```

Example: metaBup

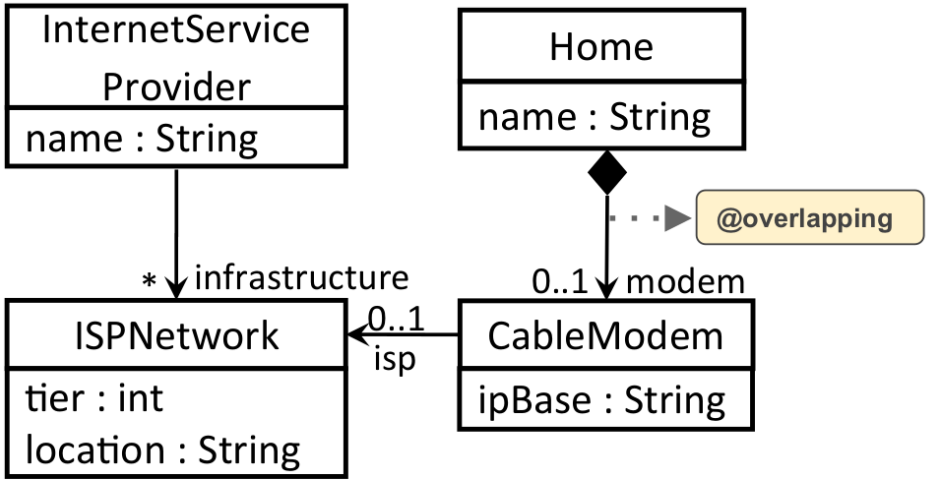
Step 2: Transform to text model (MM)



Example: metaBup

Step 3: Derive metamodel

```
shell importedFragment fragment fragment1 {
    Home_1 : Home {
        attr name = "Elliott Smith"
        @ overlapping ref modem = CableModem_3
        }
    InternetServiceProvider_1 : InternetServiceProvider {
        attr name = "lemon"
        ref infrastructure = ISPNetwork_1, ISPNetwork_2
    }
    Home_2 : Home {
        attr name = "Damien Jurado"
        @ overlapping ref modem = CableModem_2
    }
    Home_3 : Home {
        attr name = "Laura Marlin"
        @ overlapping ref modem = CableModem_1
    }
    CableModem_1 : CableModem {
        attr ipBase =  "251.12.211.6"
        ref isp = ISPNetwork_1
    }
    CableModem_2 : CableModem {
        attr ipBase =  "251.12.210.56"
        ref isp = ISPNetwork_1
    }
    CableModem_3 : CableModem {
        attr ipBase =  "251.12.210.48"
        ref isp = ISPNetwork_2
    }
    ISPNetwork_1 : ISPNetwork {
        attr tier = 3
        attr location = "MAD"
    }
    ISPNetwork_2 : ISPNetwork {
        attr tier = 4
        attr location = "BCN"
    }
}
```



Challenges

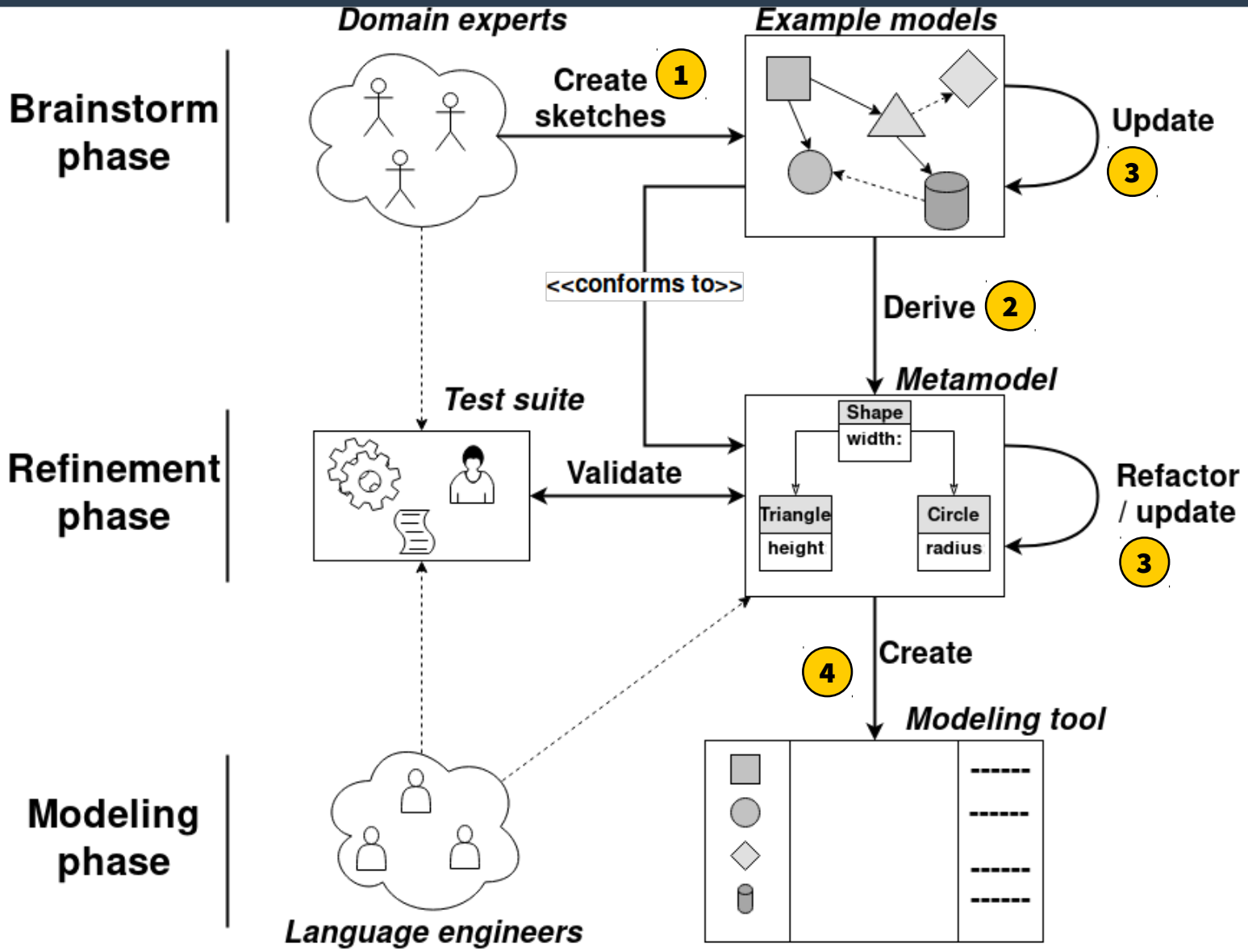
1) Sketch recognition

2) Metamodel inference

3) Evolution

4) Integration

Challenges


Domain experts
Example models
Brainstorm phase
Create sketches
1
Update
3
<<conforms to>>
Derive
2
Metamodel
Test suite
Refinement phase
Validate
Shape
width:
Triangle
height
Circle
radius
Refactor / update
3
4
Create
Modeling tool
Modeling phase
Language engineers

Evaluation

- Evaluate tools based on challenges in multiple dimensions
- Define test cases and criteria for each dimension

Evaluation

Unconstrained input

Tool	Language type	Input	Entites and relationships	Textual annotations	Spatial relationships
Scribbler	Visual	Freehand	Yes	No	No
MLCBD	Visual	Editor	Yes	Yes	Yes (draw)
FlexiSketch	Visual	Freehand	Yes	No	No
metaBup	Visual	Editor	Yes	Yes	Yes
Model Workbench	Textual Visual	Editor (text)	Yes	Yes	Yes (textual)

Evaluation

Metamodel generation

Tool	Metamodel generation	Advanced constructs	Automation
Scribbler	Manual	-	Manual
MLCBD	Implicit	-	Semi
FlexiSketch	Implicit	-	Full
metaBup	Explicit	Inheritance Abstract classes Compositions	Full
Model Workbench	Explicit	Inheritance Abstract classes	Semi

Evaluation

Co-evolution

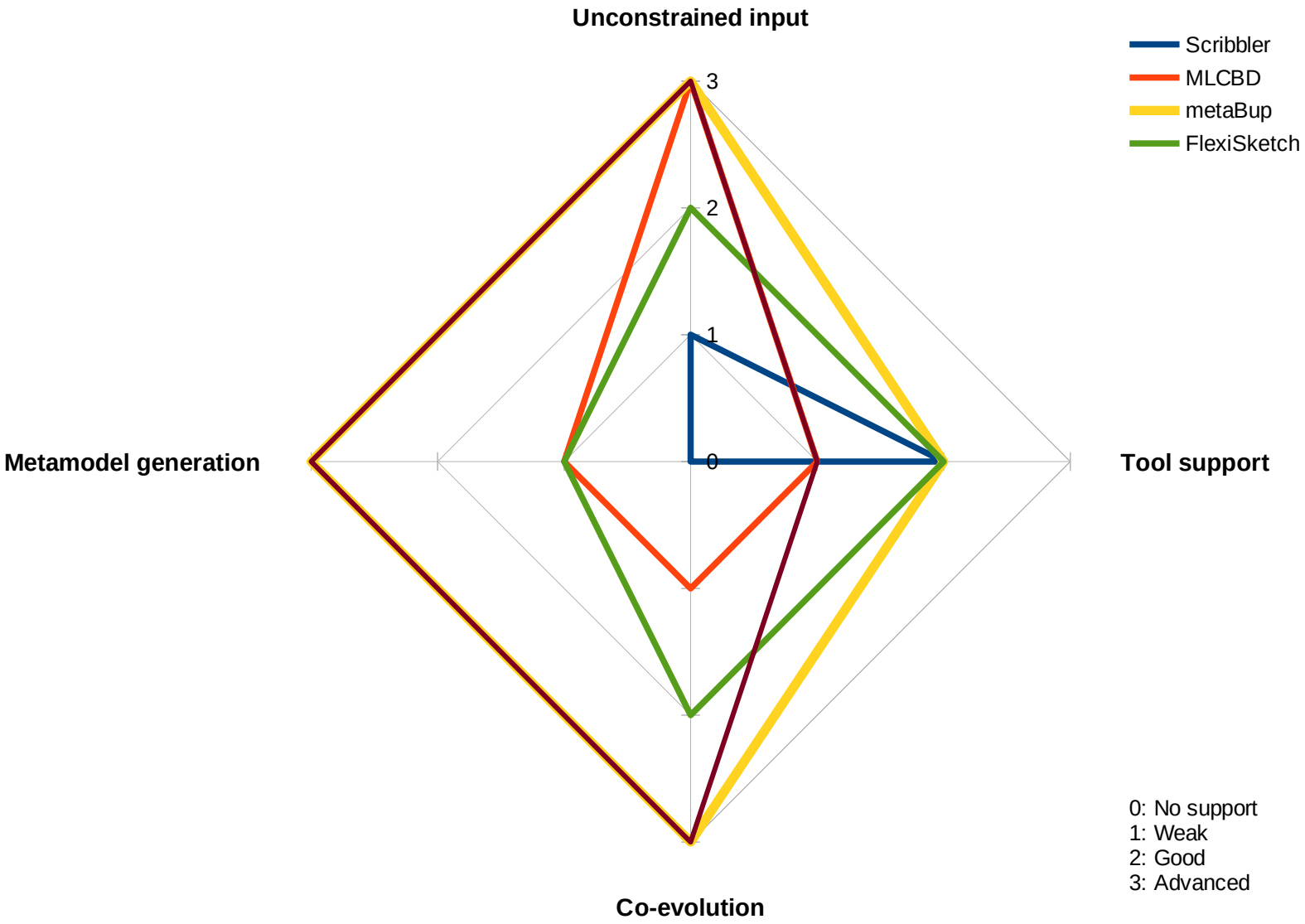
Tool	Forward evolution	Backward evolution	Classification
Scribbler	No	Manual	None
MLCBD	Manual	No	None
FlexiSketch	Yes	Yes	None
metaBup	Yes	Yes	Non-breaking Resolvable Unresolvable
Model Workbench	Yes	Yes	Non-breaking Breaking

Evaluation

Tool support

Tool	Scribbler	MLCBD	FlexiSketch	metaBup	Model Workbench
Implementation	Java Application	MS Visio Plugin	Android App Java Application	EMF Plugin	Java EE Application
Integration	EMF	-	-	EMF metaDepth	Self-contained
Usability	Industrial user study	Case study	User study	User study	-
Scalability	?	?	Limited	Good	?
Collaboration	Client-Server	-	Client-Server	-	Client-Server

Analysis

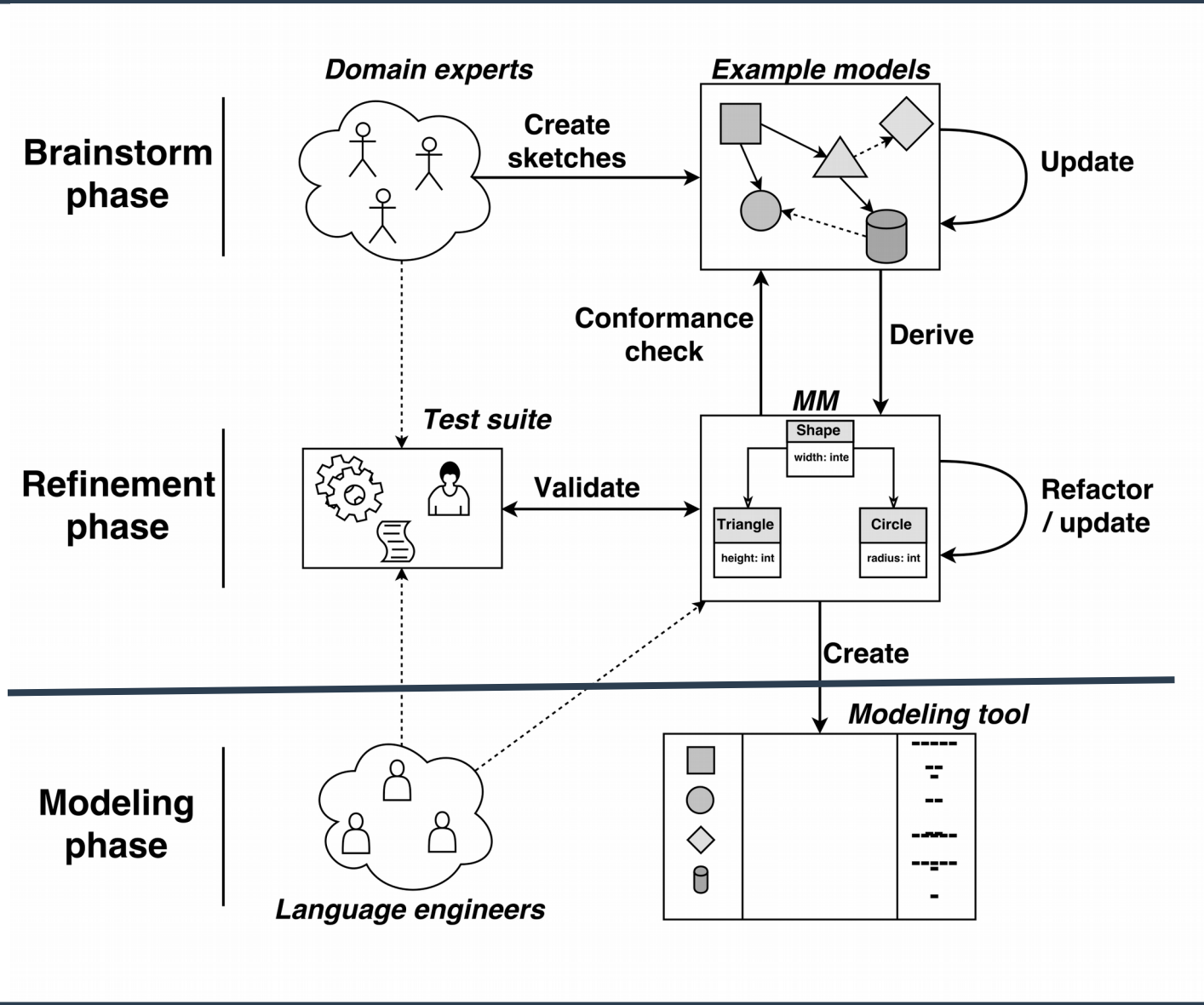


0: No support
1: Weak
2: Good
3: Advanced

Open research questions

- Concrete ↔ abstract syntax rigid
- Co-evolution mostly rudimentary
- Gap between metamodel design and usage phases
- Immature tools with little experience
- Semantics

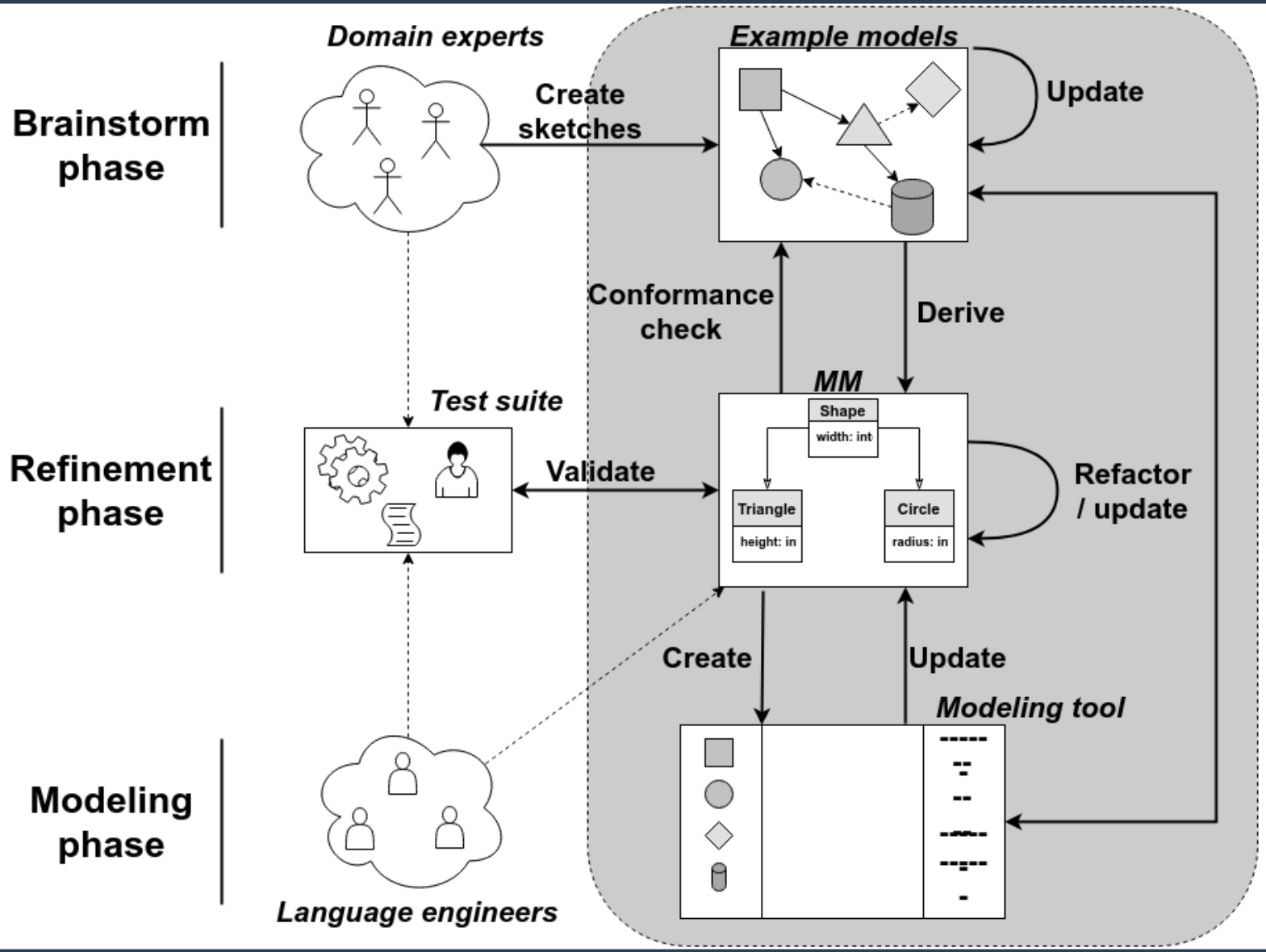
How agile is it really?



Open research questions

- Concrete ↔ abstract syntax rigid
- **Co-evolution mostly rudimentary**
- **Gap between metamodel design and usage phases**
- Immature tools with little experience
- Semantics

Future work


Brainstorm phase
Refinement phase
Modeling phase
Domain experts
Create sketches
Example models
Update
Conformance check
Derive
MM
Shape
width: int
Triangle
height: in
Circle
radius: in
Test suite
Validate
Refactor / update
Create
Update
Modeling tool
Language engineers