Feature Modelling:

A Survey, a Formalism and a Transformation for Analysis

Thomas De Vylder

Content

- Feature Modelling Origins
- Feature Modelling Notations & Definitions
- AToM^3 Meta-Model
- Formal Analysis of Feature Models
  + Operations
  + Model Transformation To Alloy
- Beware
- Demo

Feature Modelling Origins

Feature Modelling Origins

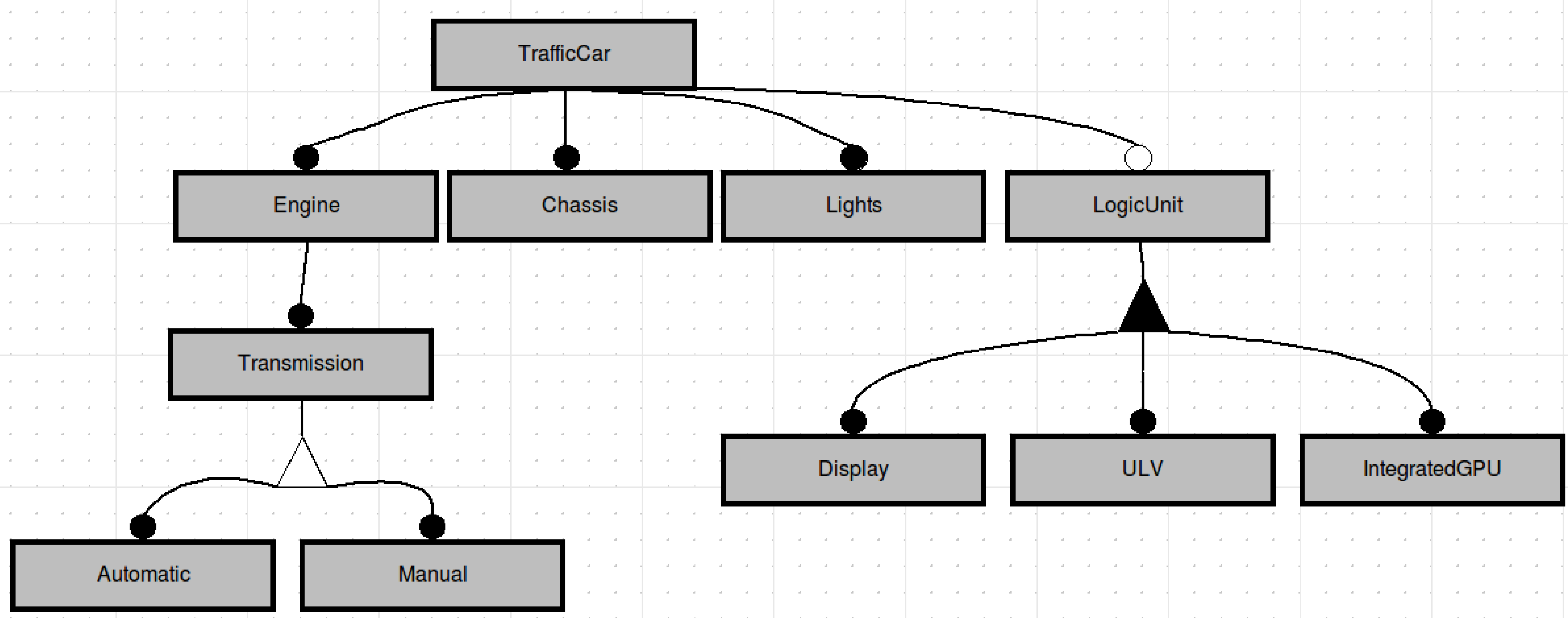
- First introduced in Feature-Oriented Domain Analysis (FODA) method*
  - FODA is the Domain Analysis component of Model Based Software Engineering (MBSE)
  - Feature modelling captures the commonalities and variabilities in terms of features of systems in a domain
- Vigoriously used in the Software Product Line (SPL) community
  - SPL is identified by a unique and legal combination of features

***** Kang, K. C., Cohen, S. G., Hess, J. A., Novak, W. E., Peterson, A. S., November 1990. Feature-oriented domain analysis (foda) feasibility study. Tech. rep., Carnegie-Mellon University Software Engineering Institute.

Feature Modelling Notations & Definitions

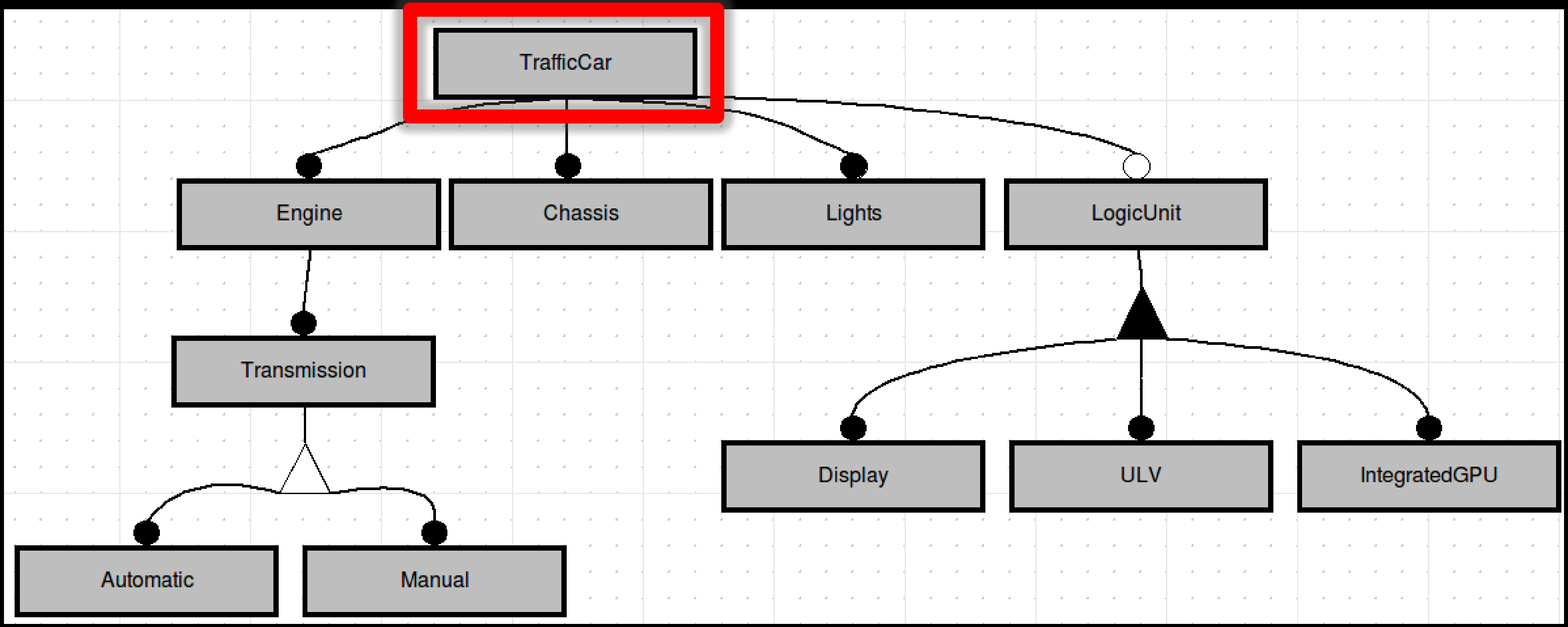
Feature Modelling Notations & Definitions

- Based on this diagram of a model:



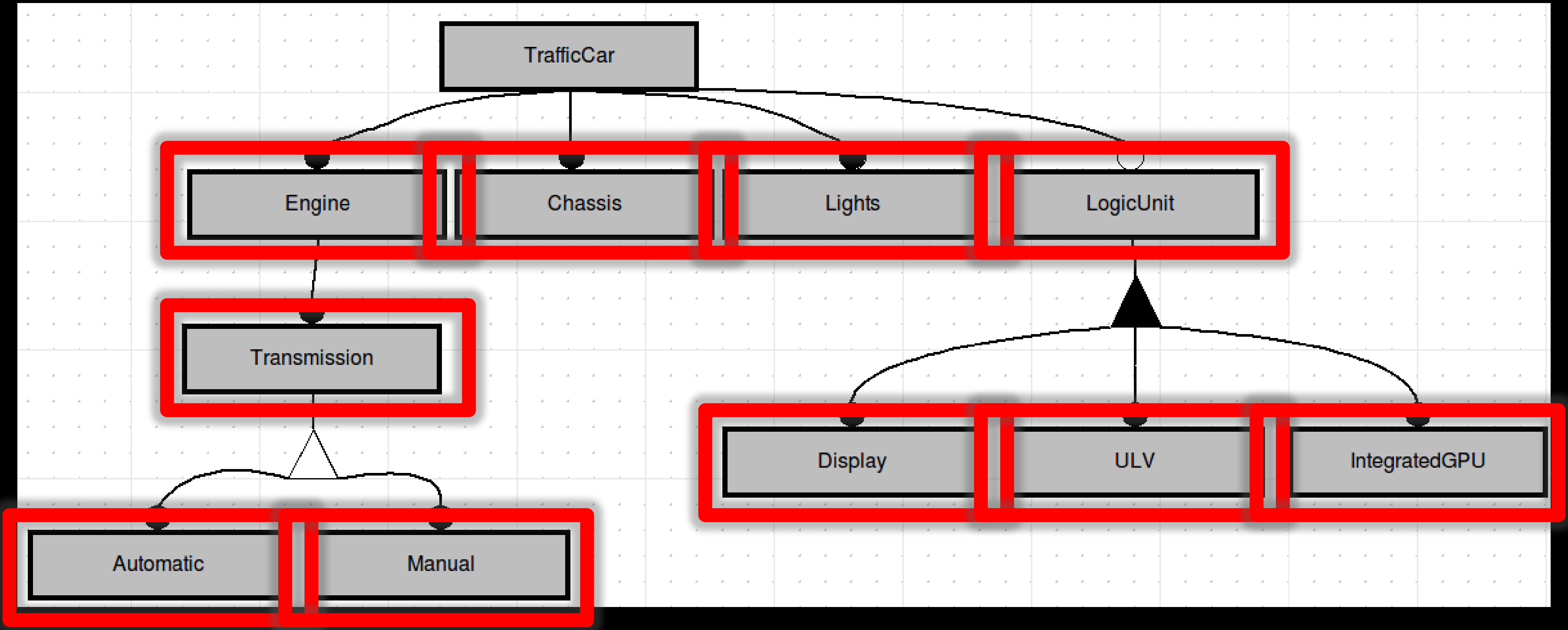
Feature Modelling Notations & Definitions

- Concept



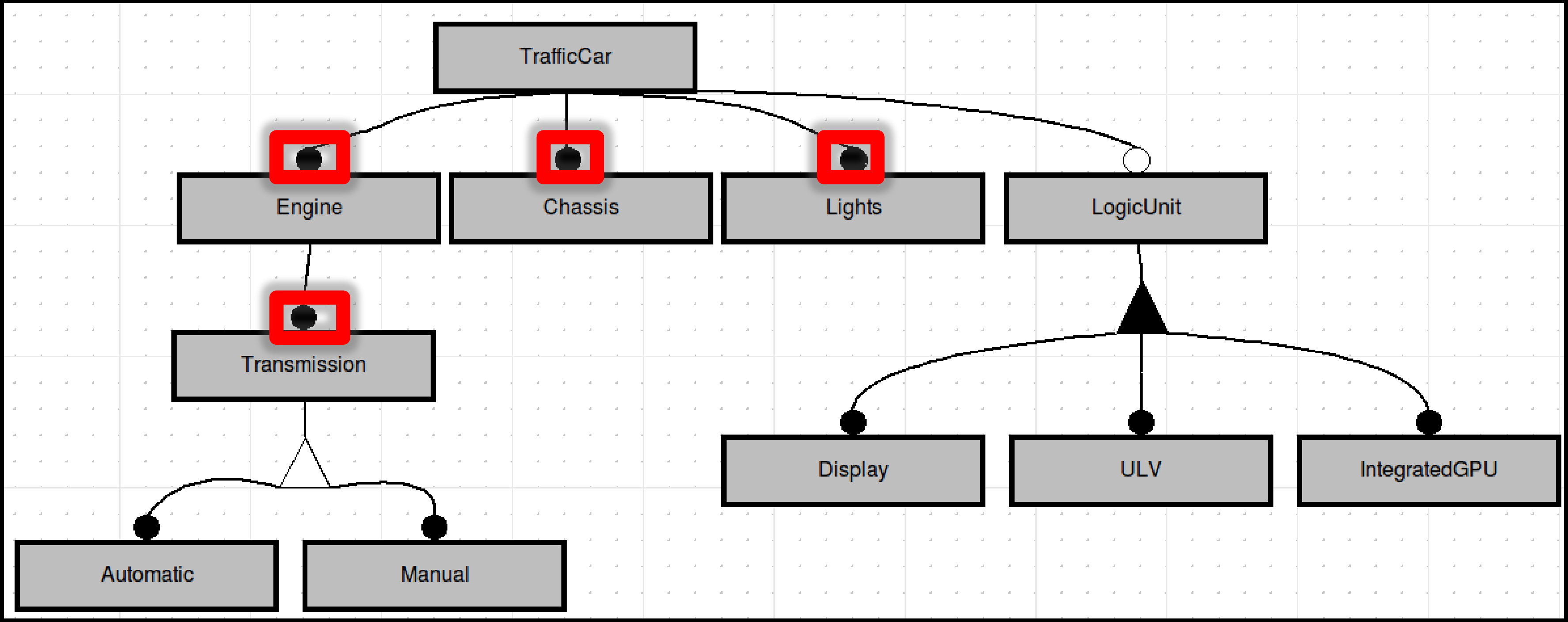
Feature Modelling Notations & Definitions

- Features



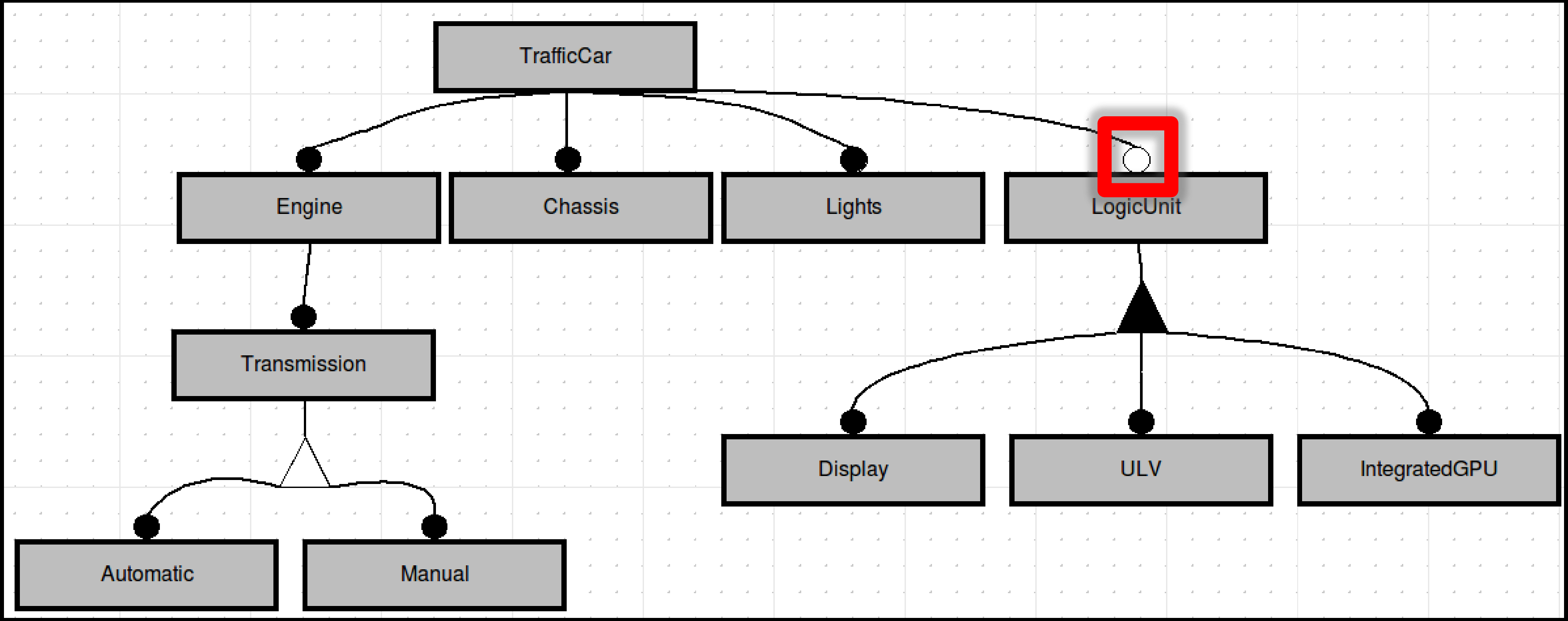
Feature Modelling Notations & Definitions

- Mandatory Features



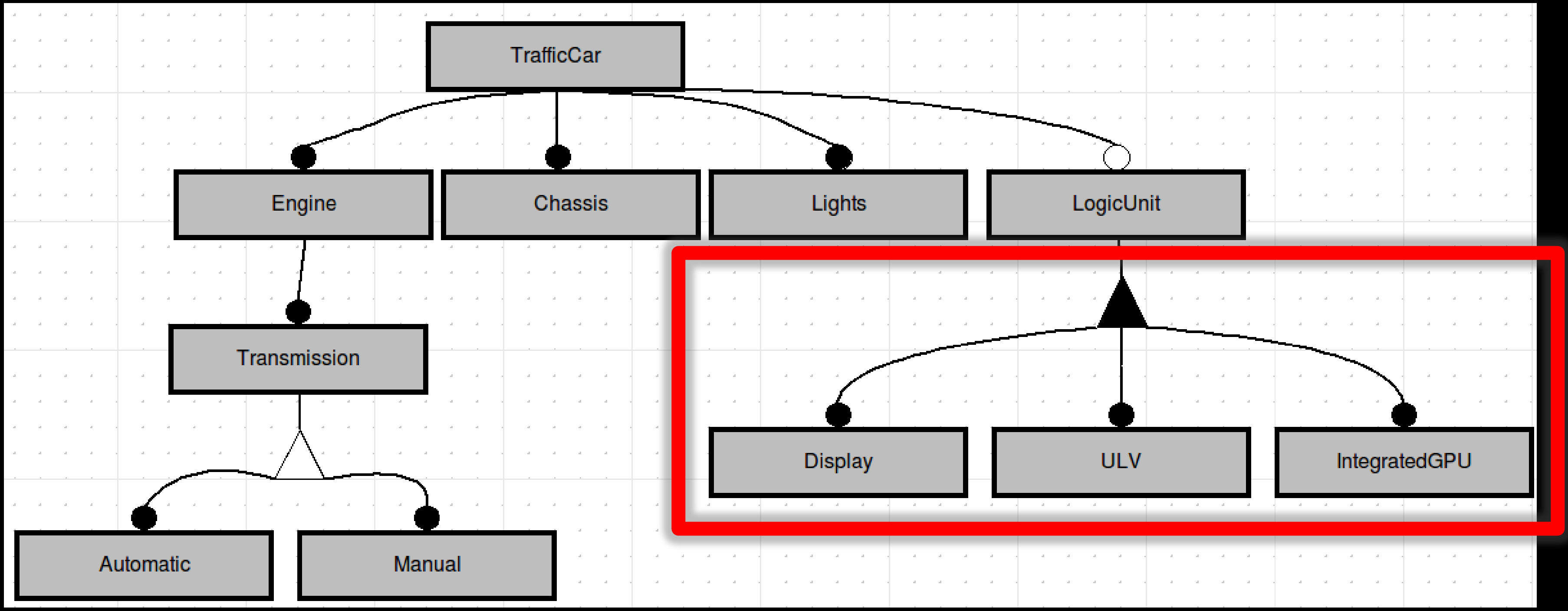
Feature Modelling Notations & Definitions

- Optional Features



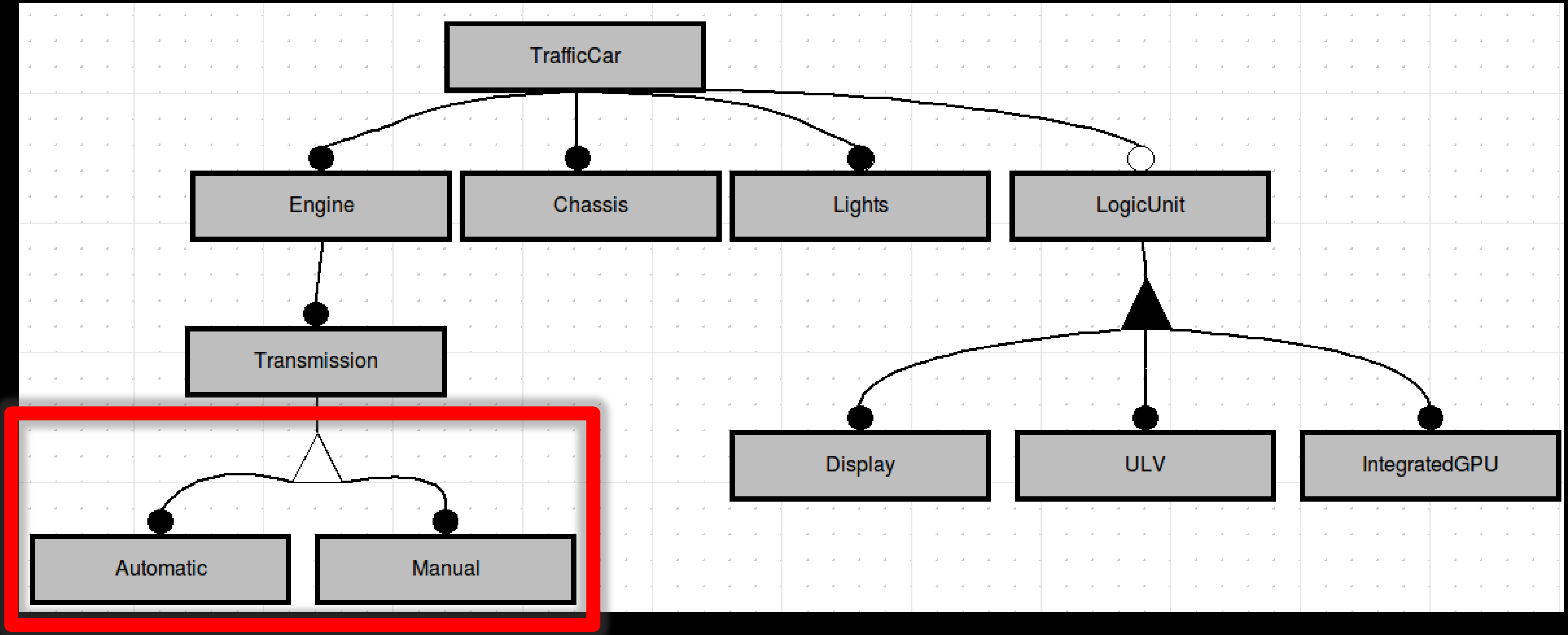
Feature Modelling Notations & Definitions

- OR Features (at least one)



Feature Modelling Notations & Definitions

- Alternative (XOR) Features (one and only one)



Feature Modelling Notations & Definitions

- Sets (XOR, OR) with optional features
  - Can be normalized
- Cross graph constraints can exist
  - “All TrafficCars with an automatic transmission must have a LogicUnit”
  - “All TrafficCars with a manual transmission may not have a LogicUnit”

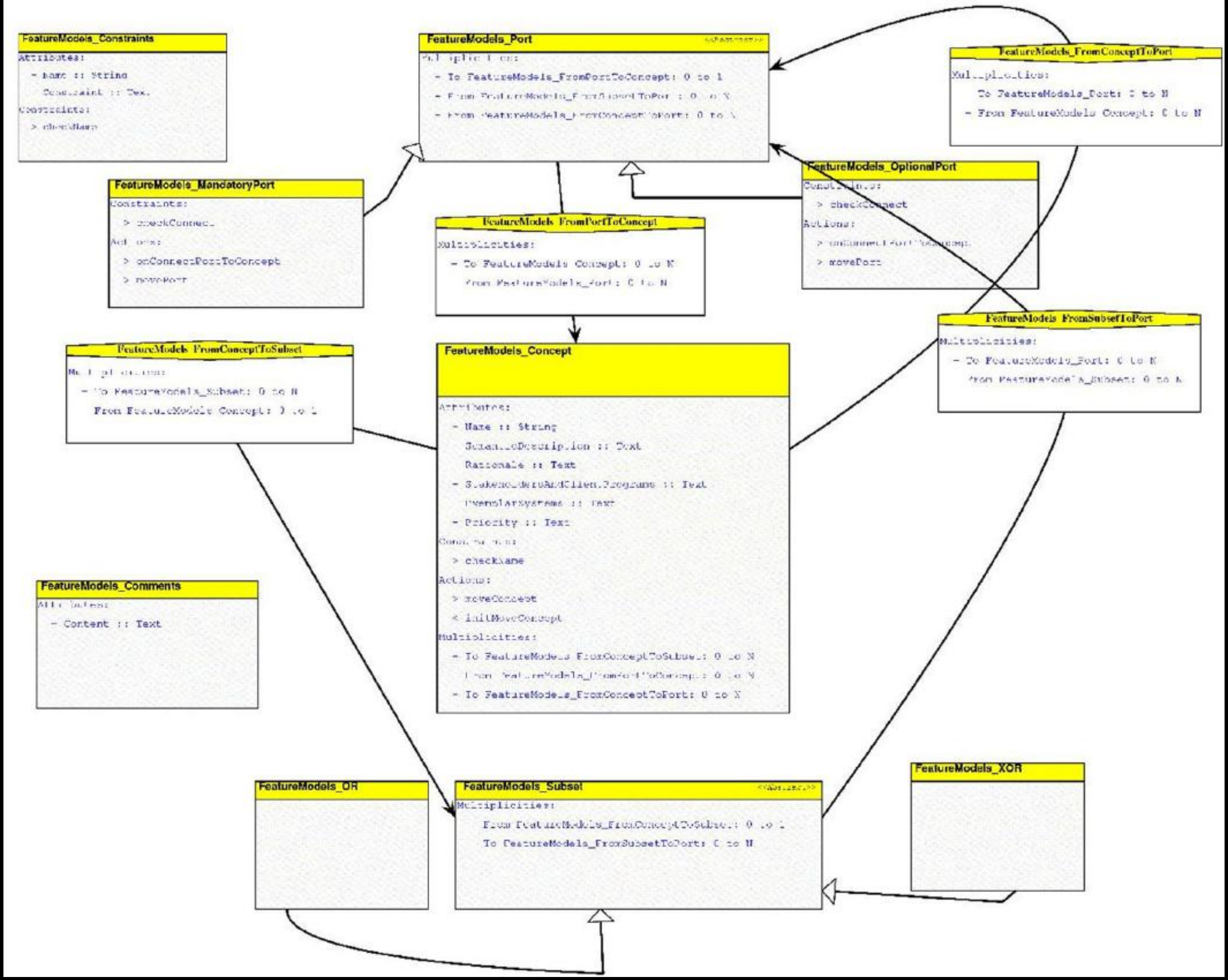
AToM^3 Meta-Model

AToM³* Meta-Model

- Concept/Feature (no distinction!)
  - Name (must start with capital letter)
  - SemanticDescription
  - Rationale
  - StakeholdersAndClientPrograms
  - ExemplarSystems
  - Priority
- Port (Inheritance: Optional/Mandatory)
- Subset (Inheritance: XOR/OR)
- Constraint
  - Name (must start with capital letter)
  - Constraint (textual relational language Alloy)
- Comments
  - Content

* De Lara, J., Vangheluwe, H., Posse, E., A. Vasudeva Murthy, I., Provost, M., Liang, W., 2002. AToM3 A Tool for Multi-formalism and Meta-Modelling. URL http://atom3.cs.mcgill.ca/index html

AToM³ Meta-Model



FeatureModels_Constraints
Attributes:
- Name :: String
Constraint :: Text
Constraints:
> checkName

FeatureModels_Port <<Abstract>>
Multiplicities:
- To FeatureModels_FromPortToConcept: 0 to 1
- From FeatureModels_FromSubsetToPort: 0 to N
- From FeatureModels_FromConceptToPort: 0 to N

FeatureModels_FromConceptToPort
Multiplicities:
To FeatureModels_Port: 0 to N
- From FeatureModels Concept: 0 to N

FeatureModels_MandatoryPort
Constraints:
> checkConnect
Actions:
> onConnectPortToConcept
> movePort

FeatureModels_FromPortToConcept
Multiplicities:
- To FeatureModels Concept: 0 to N
From FeatureModels_Port: 0 to N

FeatureModels_OptionalPort
Constraints:
> checkConnect
Actions:
> onConnectPortToConcept
> movePort

FeatureModels_FromSubsetToPort
Multiplicities:
- To FeatureModels_Port: 0 to N
From FeatureModels_Subset: 0 to N

FeatureModels_FromConceptToSubset
Multiplicities:
- To FeatureModels_Subset: 0 to N
From FeatureModels Concept: 0 to 1

FeatureModels_Concept
Attributes:
- Name :: String
SemanticDescription :: Text
Rationale :: Text
- StakeholdersAndClientPrograms :: Text
ExemplarSystems :: Text
- Priority :: Text
Constraints:
> checkName
Actions:
> moveConcept
< initMoveConcept
Multiplicities:
- To FeatureModels FromConceptToSubset: 0 to N
From FeatureModels_FromPortToConcept: 0 to N
- To FeatureModels_FromConceptToPort: 0 to N

FeatureModels_Comments
Attributes:
- Content :: Text

FeatureModels_OR

FeatureModels_Subset <<[illegible]>>
Multiplicities:
From FeatureModels_FromConceptToSubset: 0 to 1
To FeatureModels_FromSubsetToPort: 0 to N

FeatureModels_XOR

Formal Analysis of Feature Models

Formal Analysis of Feature Models

- Which operations must be included?*
  - no consensus
- Possible tasks:
  - Determine the satisfiability
    Is a product represented by the feature model?
  - Dead feature
    Can a product represented by the feature model have this feature?
  - Find a product
  - Obtain all products
  - …

* Benavides, D., Ruiz-Corts, A., Trinidad, P., Segura., S., 2006. A survey on the automated analyses of feature models. In: Jornadas de Ingeniera del Software y Bases de Datos (JISBD).

Formal Analysis of Feature Models

- Problem: no analysis tools that run on my feature models
- Solution: model transformation!

- Transform to?
  - Alloy*: a textual modelling language based on first order relational logic.

***** Jackson, D., 2002. Alloy Analyzer website. URL http://alloy.mit.edu/

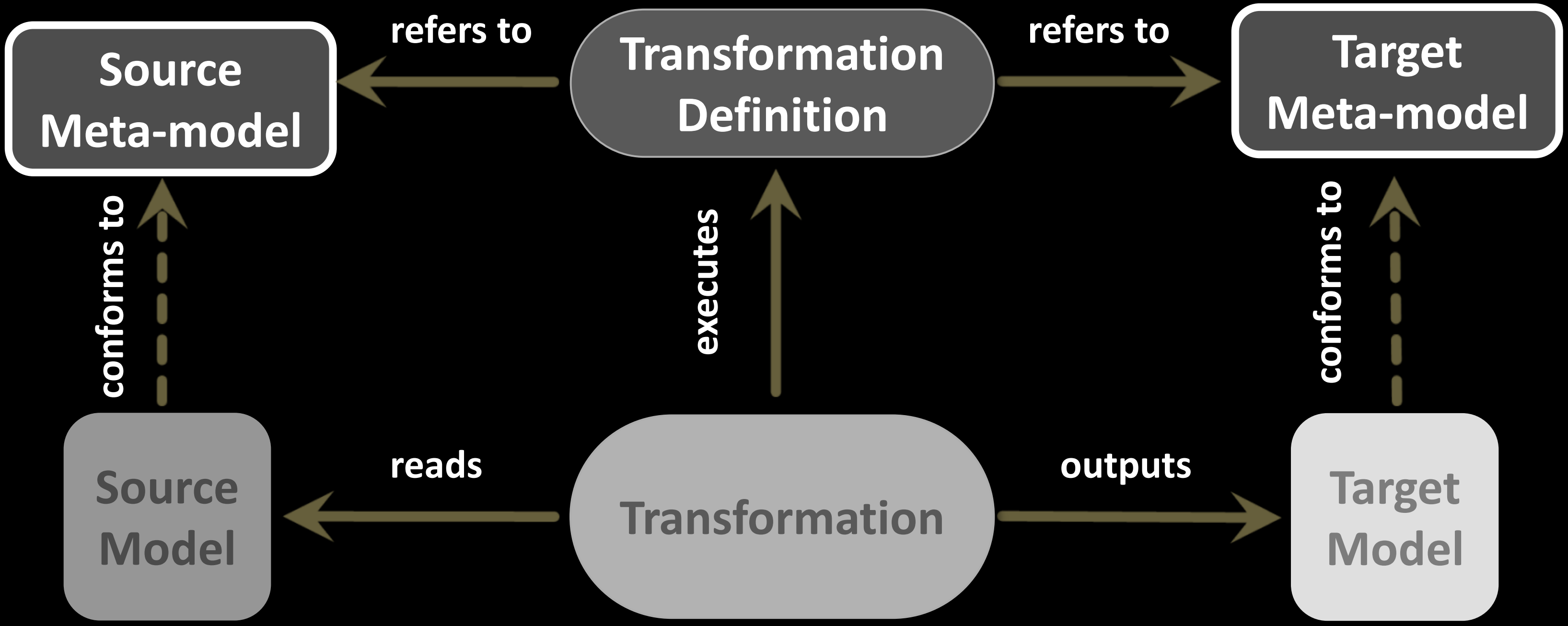
Formal Analysis of Feature Models

Skip?

- Alloy
  - Signatures
  - Fields
  - Facts
  - Asserts
  - Predicates
  - some X
  - no X
  - one X
  - lone X
  - all x: X | formula
  - some x: X | formula
  - no x: X | formula
  - one x: X | formula
  - lone x: X | formula

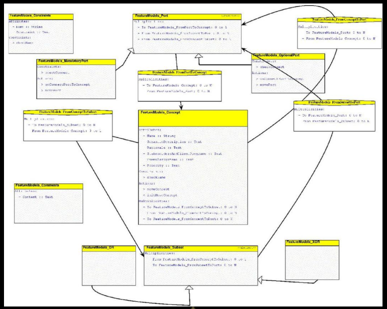
Formal Analysis of Feature Models

- Model Transformation

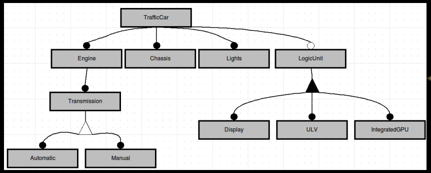


Formal Analysis of Feature Models

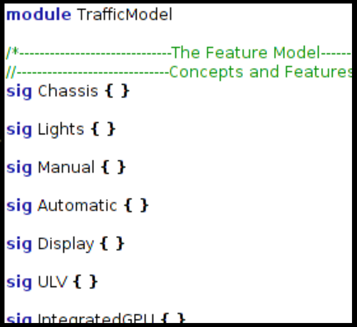
- Model Transformation



TransformerFM
2Alloy.py



Transformation



```
module TrafficModel

/*----------------------------The Feature Model------
//----------------------------Concepts and Features
sig Chassis { }

sig Lights { }

sig Manual { }

sig Automatic { }

sig Display { }

sig ULV { }

sig IntegratedGPU { }
```

Formal Analysis of Feature Models

- Model Transformation
  - metadata → comments + module name
  - features/concepts → signatures with same name
  - relations → fields with constraints/facts
  - cross graph constraints → facts
- Extra:
  - comments to create structure
  - assertions to check for dead features
  - predicates/facts and runs for product finding and determining the satisfiability

Beware

Beware

- Presentation $\subset$ Paper
  - More background/detail -> read paper!
  - More mathematics -> read paper!
  - Extentions to feature models -> read paper!
  - Exact rules -> read paper!
  - Characteristics of transformation -> read paper!
  - …. -> read paper!

Beware

- Wikipedia has other notations!
  - Everybody has their own notations (Czarnernecki, Kang, me, …)
  - I tried to be the same as Czarnernecki
- Feature models are abstract
  - Do you want your entity of a car to have “the same” entity of an engine as the entity of the car of someone else?
  - Simulation/Real world requires more information

Demo