CVL to Clafer transformation

Tom Wijsman

University
of Antwerp

Overview

1. Clafer

2. CVL (Common Variability Language)

3. CVL to Clafer transformation

Chapter 1

Clafer

Introduction to Clafer

<u>Cla</u>ss, <u>fe</u>ature, <u>r</u>eference

A general-purpose lightweight modeling language with first-class support for feature modelling, which prefers unification over hybridity; developed at the GSD Lab, University of Waterloo and MODELS group at IT University of Copenhagen.

Design goals:

- Concise notation for feature modeling and meta-modeling;

- mixes feature models and meta-models;

- minimal number of concepts;

- uniform semantics.

Introduction to Clafer

A set of concepts:

- type definitions: a class or a feature (no distinction!);

- features: attributes or role names of association

  and composition relationships;

- constraints: Alloy-based constraints limit the variability;

- …

RPGGame in Clafer example

abstract RPGGame

    xor Players

        Singleplayer

        Multiplayer

    or Enemies?

        Villain

        Dragon

    or Environment?

        Door

        Goal

        Key

        Trap

    [Key => Door]

or WinCondition

    TakeAllGoals

    KillAllEnemies

[WinCondition.TakeAllGoals => Environment.Goal]

[WinCondition.KillAllEnemies => Enemies]

[Environment.Goal => no WinCondition.KillAllEnemies]

Generating instances for RPGGame with Alloy

Game : RPGGame
[Singleplayer]
[Key]
[Trap]

Game	Game	Game	Game	Game	Game	Game
Players	Players	Players	Players	Players	Players	Players
Singleplayer	Singleplayer	Singleplayer	Singleplayer	Singleplayer	Singleplayer	Singleplayer
Environment	Enemies	Enemies	Enemies	Enemies	Enemies	Enemies
Door	Dragon	Villain	Villain	Dragon	Villain	Villain
Goal	Environment	Environment	Environment	Environment	Dragon	Dragon
Key	Door	Door	Door	Door	Environment	Environment
Trap	Key	Key	Goal	Goal	Door	Door
WinCondition	Trap	Trap	Key	Key	Key	Goal
TakeAllGoals	WinCondition	WinCondition	Trap	Trap	Trap	Key
	KillAllEnemies	KillAllEnemies	WinCondition	WinCondition	WinCondition	Trap
			TakeAllGoals	TakeAllGoals	KillAllEnemies	WinCondition
						TakeAllGoals

Multiple-Objective Optimization with Clafer Moo Visualizer
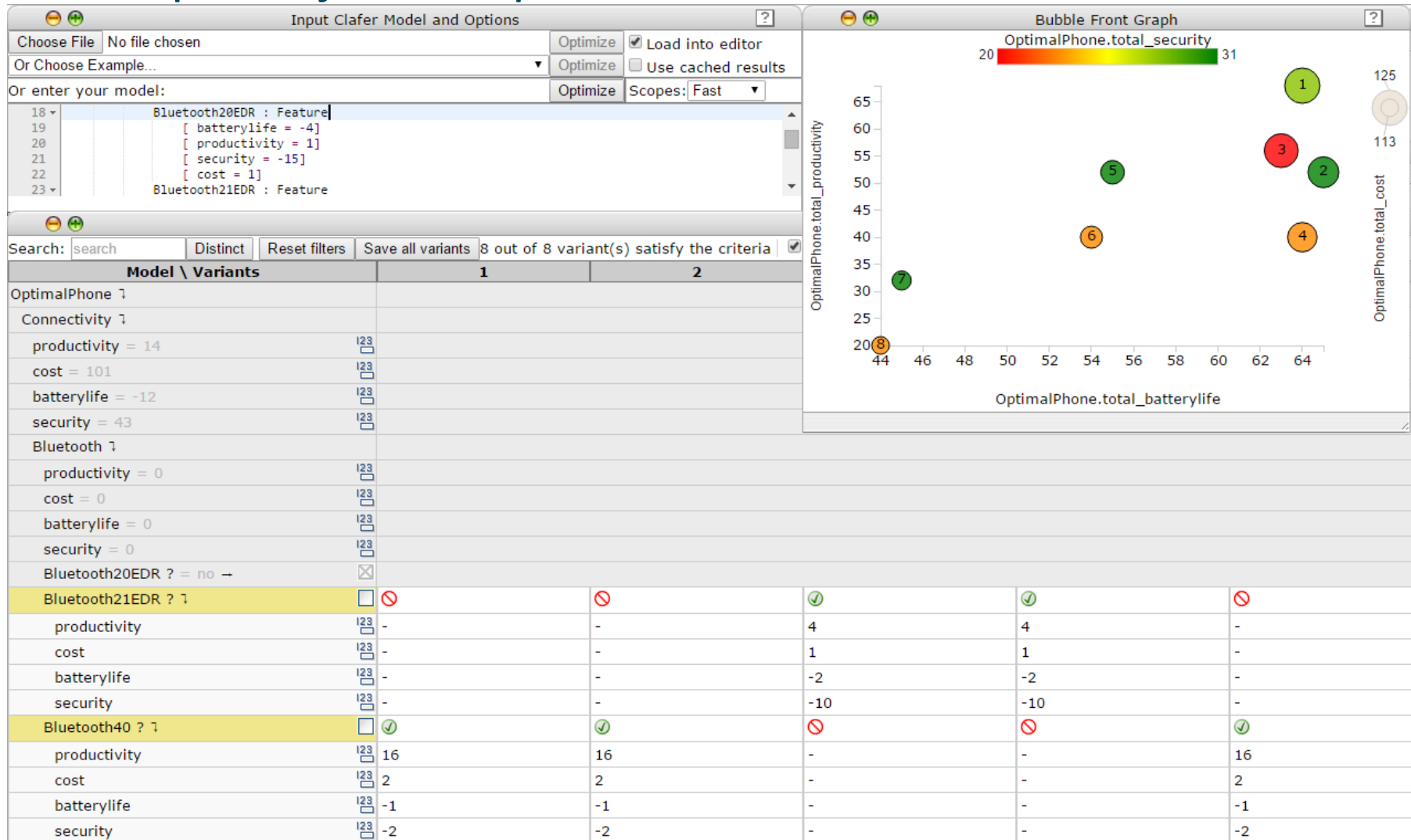
```
total_productivity : integer = sum Feature.productivity      << max OptimalPhone.total_batterylife >>
total_batterylife : integer = sum Feature.batterylife       << max OptimalPhone.total_productivity >>
total_security : integer = sum Feature.security             << max OptimalPhone.total_security >>
total_cost : integer = sum Feature.cost                     << min OptimalPhone.total_cost >>
```
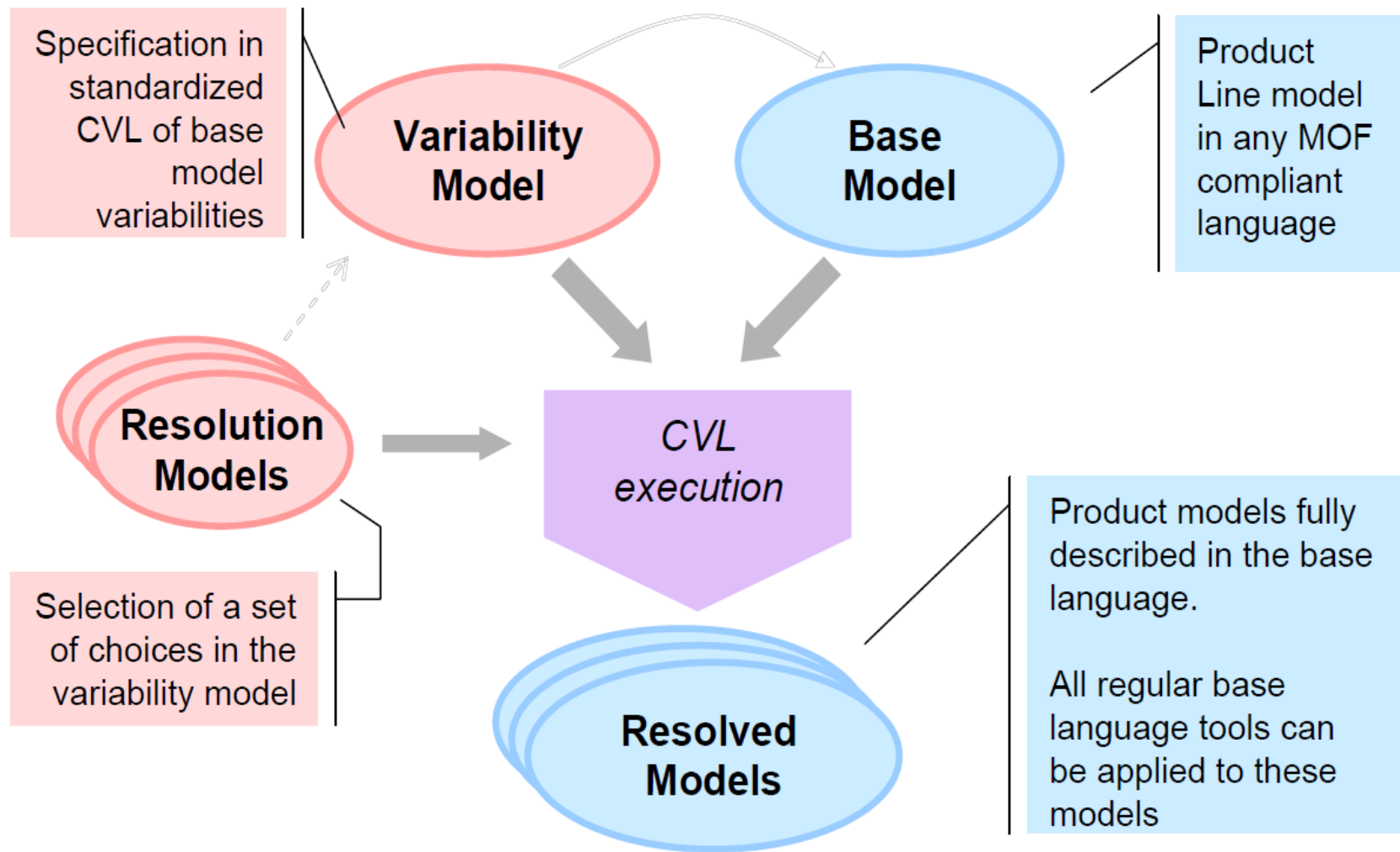
Chapter 2
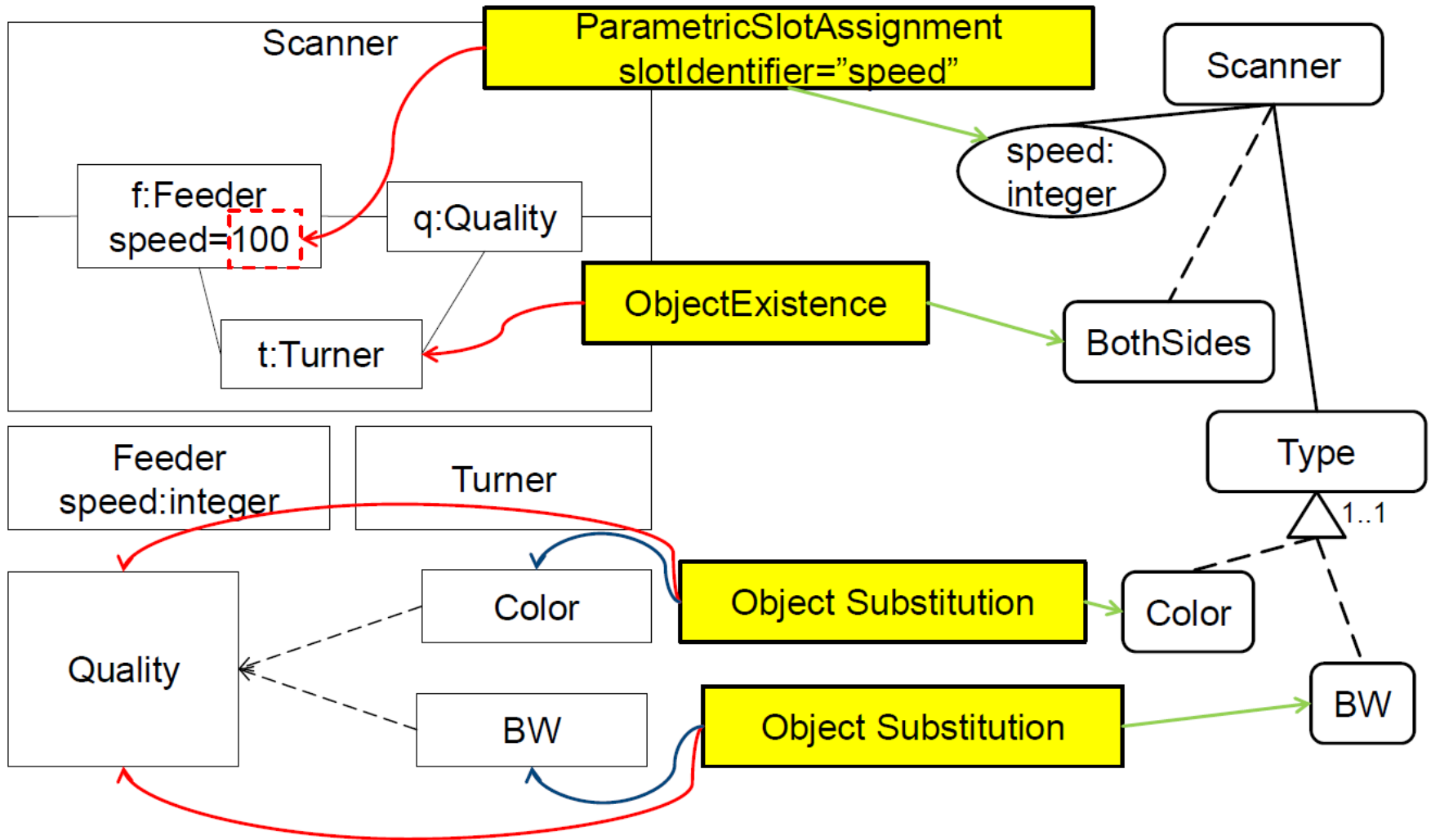
CVL (Common Variability Language)

Introduction to CVL

Specification in standardized CVL of base model variabilities — **Variability Model**

Base Model — Product Line model in any MOF compliant language

Resolution Models

Selection of a set of choices in the variability model

CVL execution

Resolved Models

Product models fully described in the base language.

All regular base language tools can be applied to these models

IBM, FOKUS, F., Thales, Services, T. C., August 2012. Common Variability Language (CVL). See the CVL Revised Submission section online at http://www.omgwiki.org/variability/doku.php.

Introduction to CVL

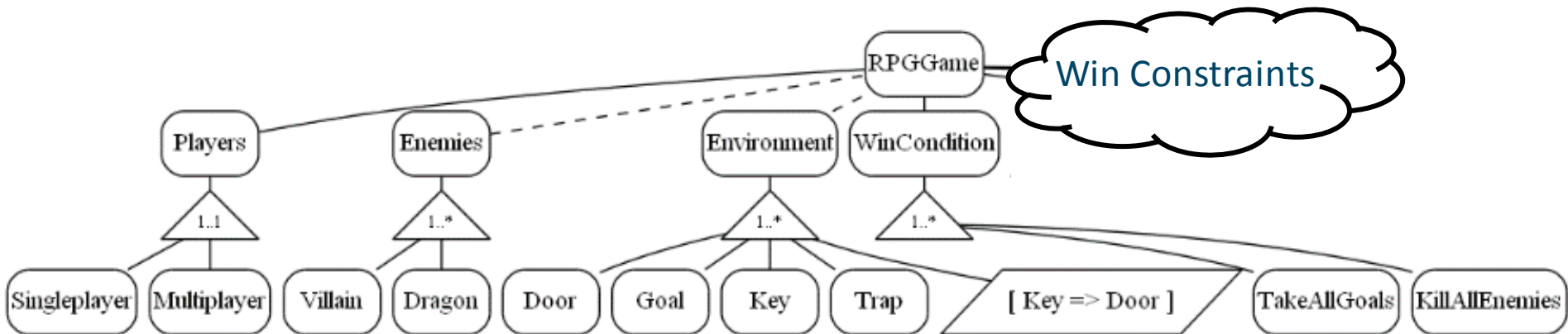

IBM, FOKUS, F., Thales, Services, T. C., August 2012. Common Variability
    Language (CVL). See the CVL Revised Submission section online at
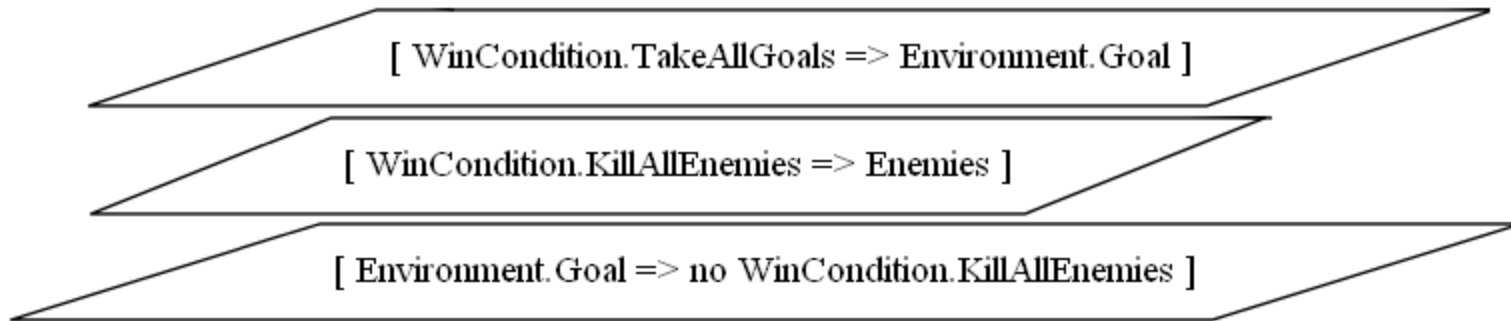   http://www.omgwiki.org/variability/doku.php.
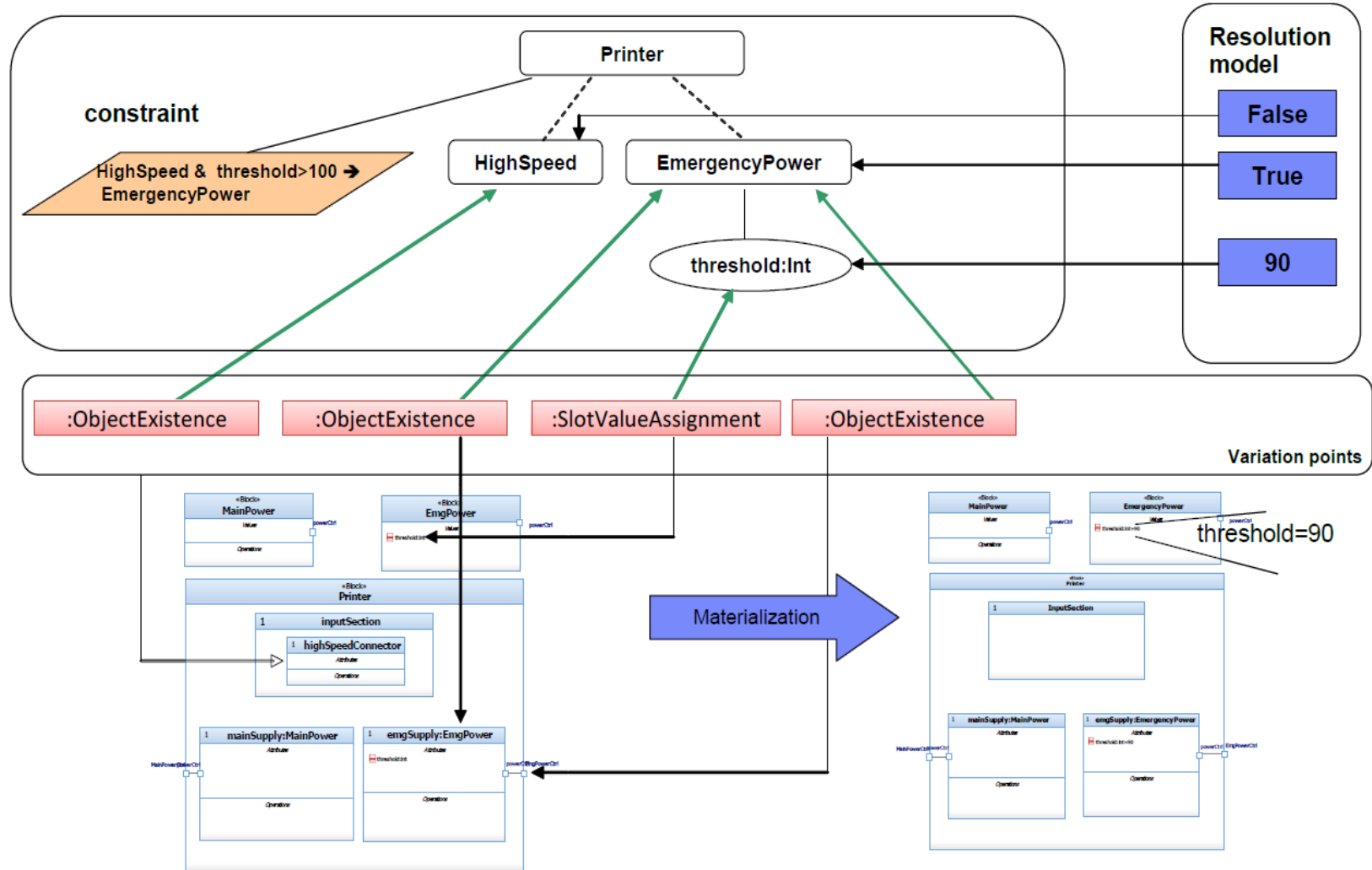
Introduction to CVL

IBM, FOKUS, F., Thales, Services, T. C., August 2012. Common Variability
Language (CVL). See the CVL Revised Submission section online at
http://www.omgwiki.org/variability/doku.php.

RPGGame in CVL example

Win Constraints:

[WinCondition.TakeAllGoals => Environment.Goal]

[WinCondition.KillAllEnemies => Enemies]

[Environment.Goal => no WinCondition.KillAllEnemies]

Generated with Clafer Compiler and GraphViz.

CVL has more features (composition, interface, …)

IBM, FOKUS, F., Thales, Services, T. C., August 2012. Common Variability Language (CVL). See the CVL Revised Submission section online at http://www.omgwiki.org/variability/doku.php.
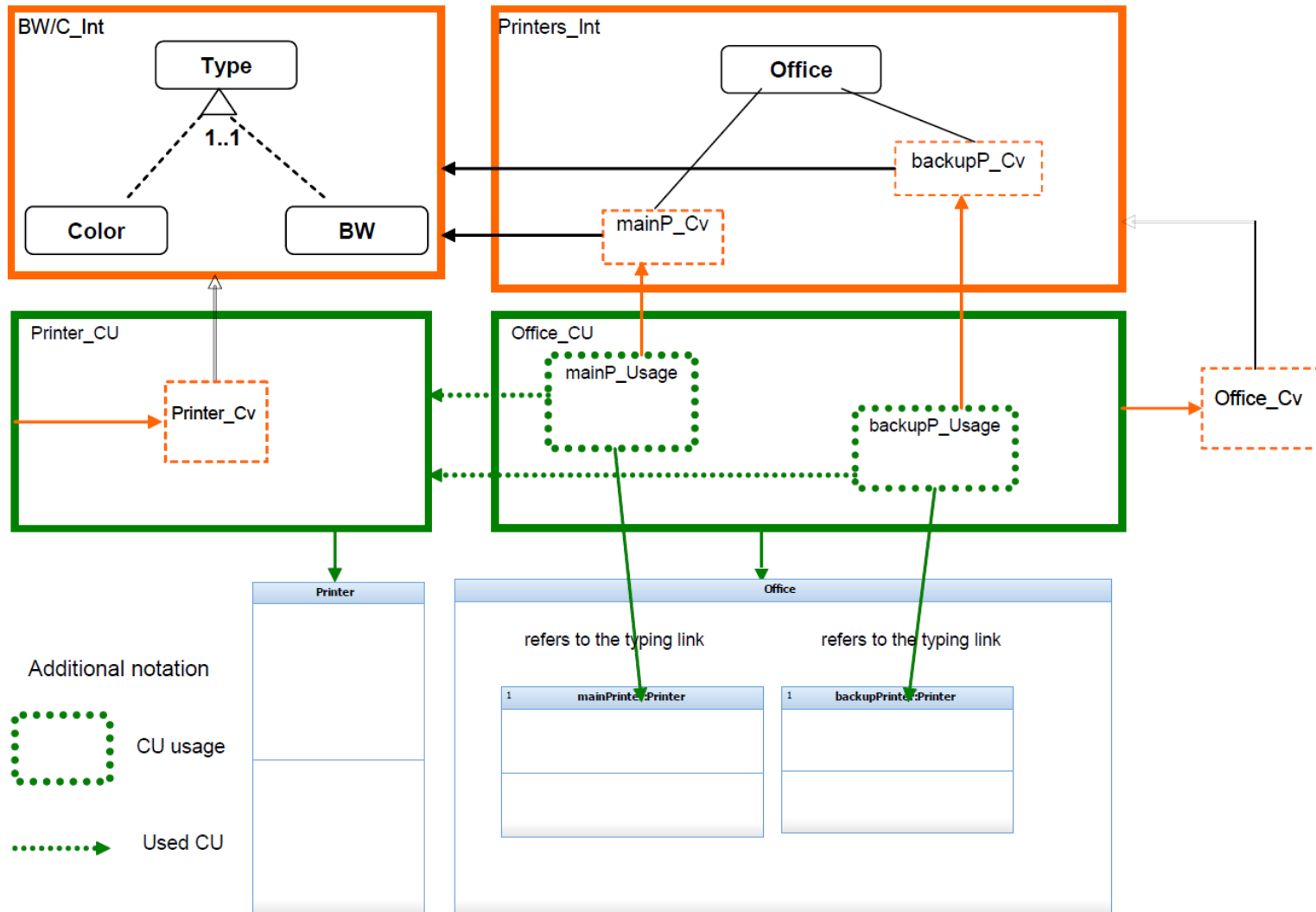
Chapter 3

CVL to Clafer transformation

Steps

Preparation

- identify a reasonable set of common features for both CVL, Clafer and their constraint languages

- create an abstract and concrete syntax of CVL in AToMPM

Transformation

- AToMPM CVL *=EGL=>* metaDepth CVL *=ETL=>* metaDepth Clafer *=EGL=>* Clafer

Verification

- verify by a reverse transformation using Clafer Compiler

Bibliography

Kacper, B., Krzysztof, C., Andrzej, W., 2011. Feature and meta-models in
Clafer: Mixed, specialized, and coupled. Lecture Notes in Computer Science 6563, 102-122.

Kacper, B., Zinovy, D., Micha, A., Krzysztof, C., Andrzej, W., December
2014. Clafer: unifying class and feature modeling. Software & Systems
Modeling 14.

Micha, A., Kacper, B., Alexandr, M., Jimmy, L., Rafael, O., Krzysztof,
C., 2013. Clafer Tools for product line engineering. In: Proceedings of
the 17th International Software Product Line Conference co-located workshops. Software Product Line Conference, ACM, Tokyo, Japan, pp. 130-135.

IBM, FOKUS, F., Thales, Services, T. C., August 2012. Common Variability
Language (CVL). See the CVL Revised Submission section online at
http://www.omgwiki.org/variability/doku.php.

Thank you for your attention!

Comments and questions are welcome.

University
of Antwerp