# Verifying VSpec-Resolution Consistency

## Basics

The VSpec model is converted to a contextually equivalent ecore model. Each resolution model is converted to an instance of the ecore model. Then, the OCL constraint can be verified to check the consistency.

## Feature

Features are classes in ecore.

**class** F{

...

}

The configurable unit is always the root class. It contains the global constraints.

## VClassifier

VClassifiers are also modelled as classes.

## Mandatory

A mandatory relationship is modelled as a composed property.

**class** A{

**property** b : *B* { **composes** };  
 ...  
 }  
 **class** B;

## Optional

An optional relationship gets a question mark.

**class** A{

**property** b : *B*[?] { **composes** };  
 ...  
 }  
 **class** B;

## Instance-limitations

Here, B is a VClassifier with a "[1..\*]" instance multiplicity. It could be written out with numbers explicitly, as one would have to do with, e.g. "[2..4]".

**class** A{

**property** b : *B*[+] { **composes** };  
 ...  
 }  
 **class** B;

## Included and Excluded

That a feature is included or excluded are modelled by checking whether it is not null, "*f* <> **null**" or null, "*f* = **null**", respectively.

## Require and Exclude

A required-constraint between A and B is modelled as an implication:

**invariant** AReqB: *A* <> **null** **implies** *B* <> **null**;

And if A excludes B it is modelled by negating the right of the implication.

**invariant** AExB: *A* <> **null** **implies** *B* = **null**;

## Other Constraints

Specify a constraint either in a class or, for global constraint, in the root class.

**class** PrinterPool{

**property** printer : *Printer*[+] { **composes** };

**property** scanner : *Scanner*[\*] { **composes** };

**invariant** X:

**if** *scanner*->*size*() <> 0 **then**

*printer*->*collect*(*resolution*)->*max*() >= *scanner*->*collect*(*resolution*)->*max*()

**else**

**true**

**endif**;

}