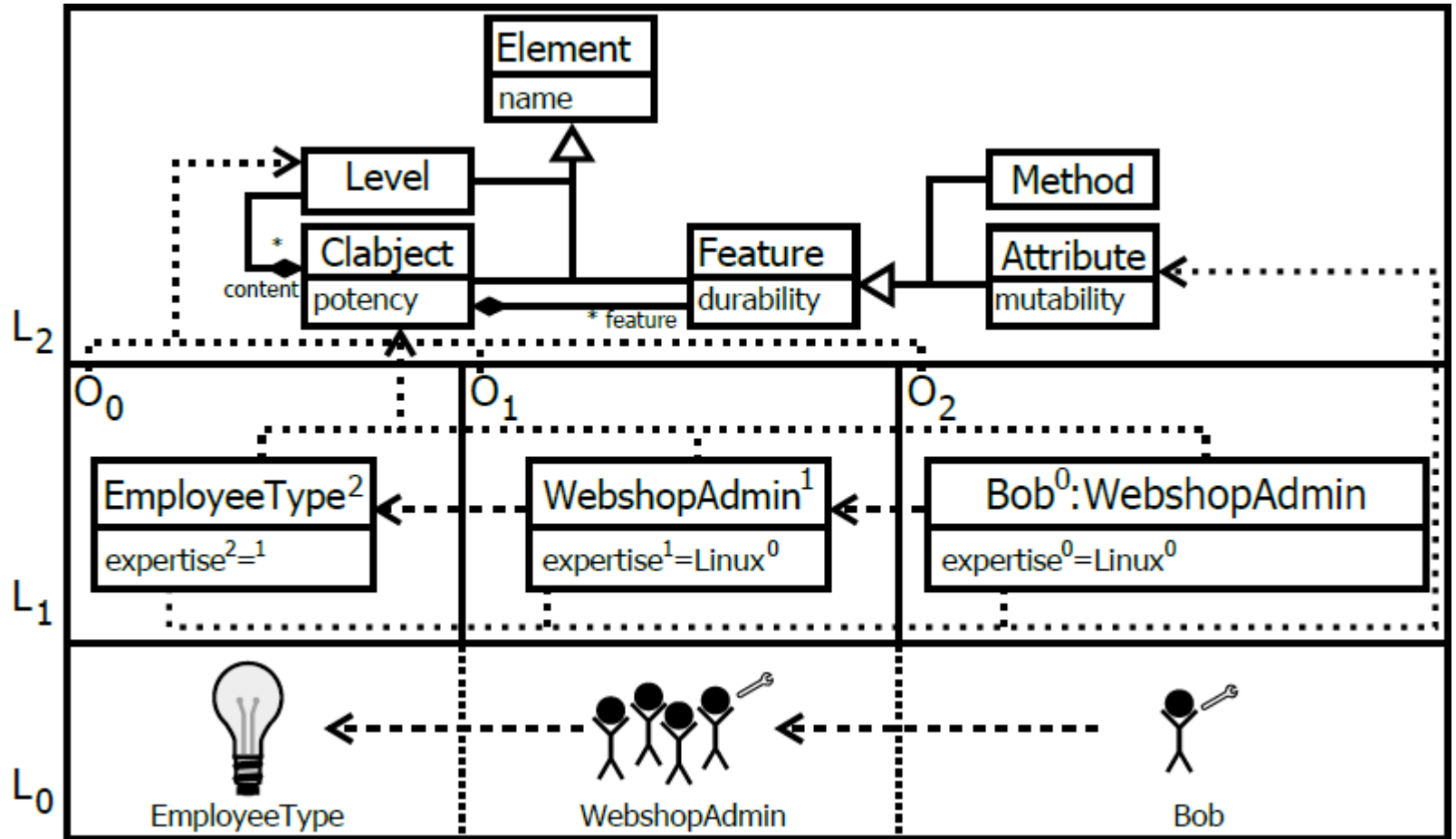




Aspect-oriented Concrete Syntax Definition for Deep Modelling Languages

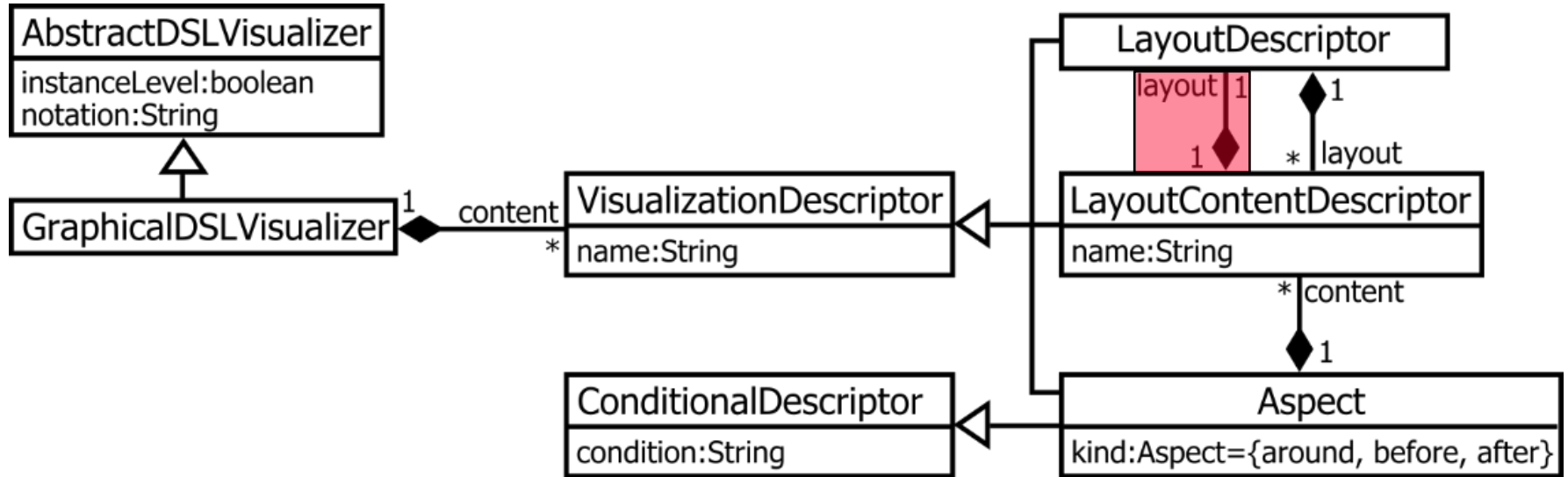
Colin Atkinson, Ralph Gerbig



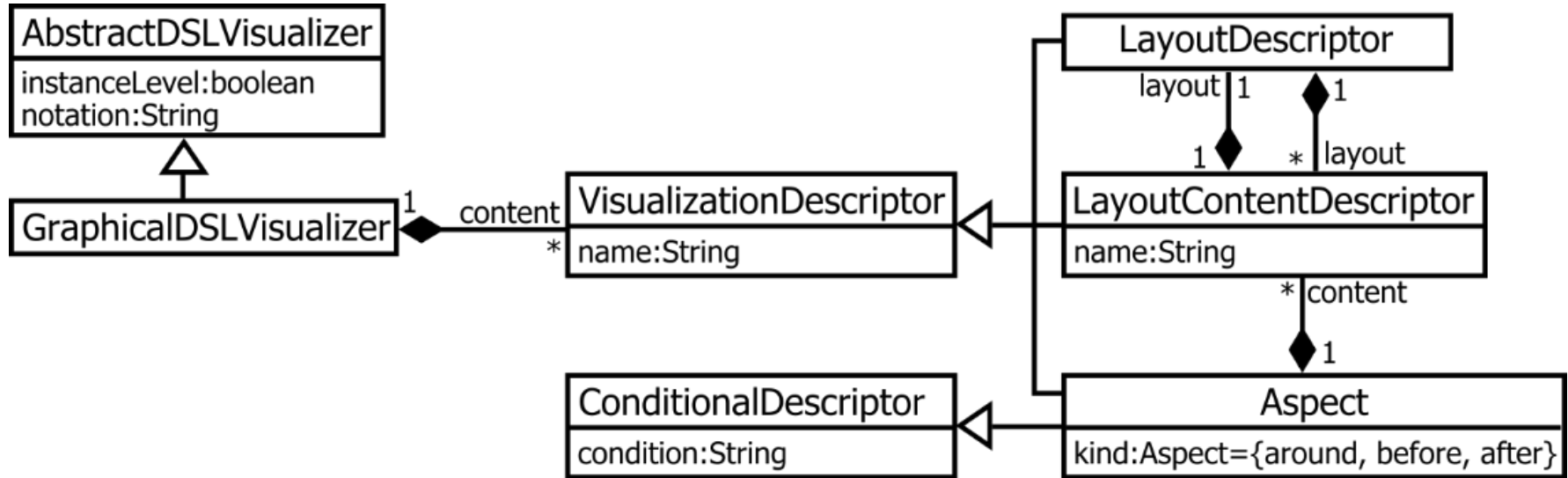


- Two Ingredients for concrete syntax definition
 - Concrete Syntax Definition Meta-model
 - Specific for each format
 - Does not ship with Melanee Core
 - Can support aspect oriented concepts in a way that best suits the modeled format
 - Visualization Search Algorithm
 - Ships with Melanee
 - Configurable via implementation of an Interface
 - E.g can support aspect-oriented concepts

Aspect Supporting Meta-model



Aspect Supporting Meta-model



GraphicalDSLVisualizer(notation=„Employee“...)

| Rectangle(outline=false)

| TableLayout(equalWidth=false, Columns=1)

| SVGFigure(URL=“stickman.svg”)

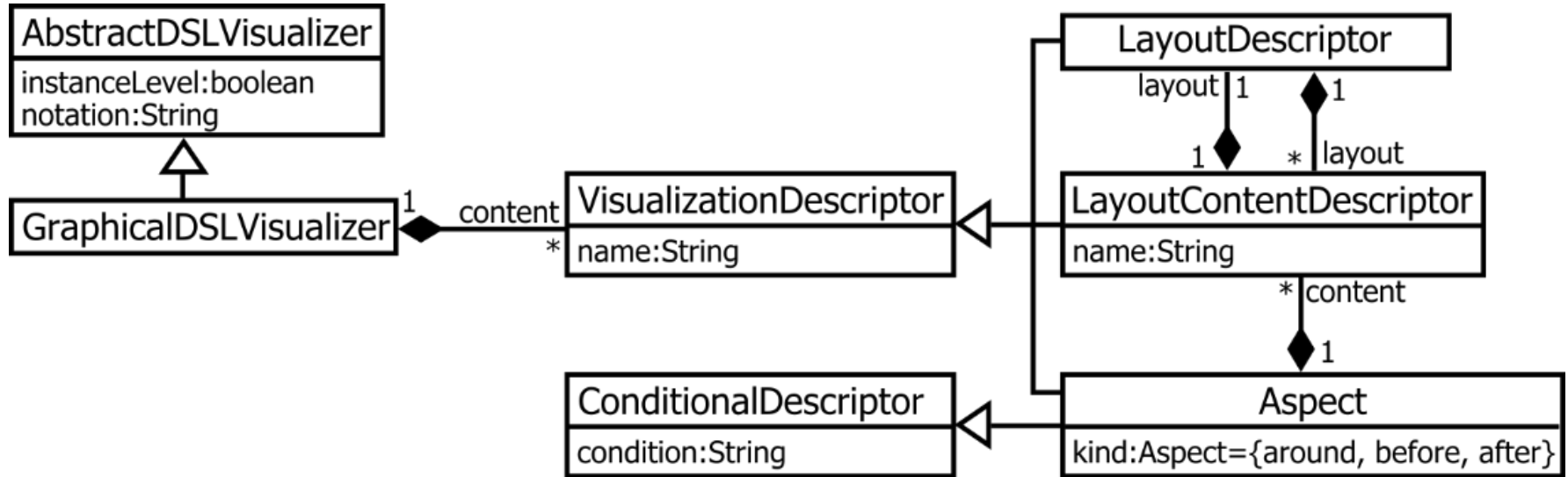
| TableLayoutInformation(grabHorizontalExcess=true...)

| MappingLabel(attribute=employeeName)

| TableLayoutInformation(grabHorizontalExcess=true...)

| ...

Aspect Supporting Meta-model



GraphicalDSLVisualizer(notation=„Employee“...)

| Rectangle(outline=false)

| TableLayout(equalWidth=false, Columns=1)

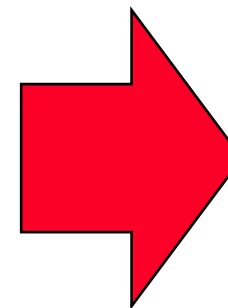
| SVGFigure(URL=“stickman.svg”)

| TableLayoutInformation(grabHorizontalExcess=true...)

| MappingLabel(attribute=employeeName)

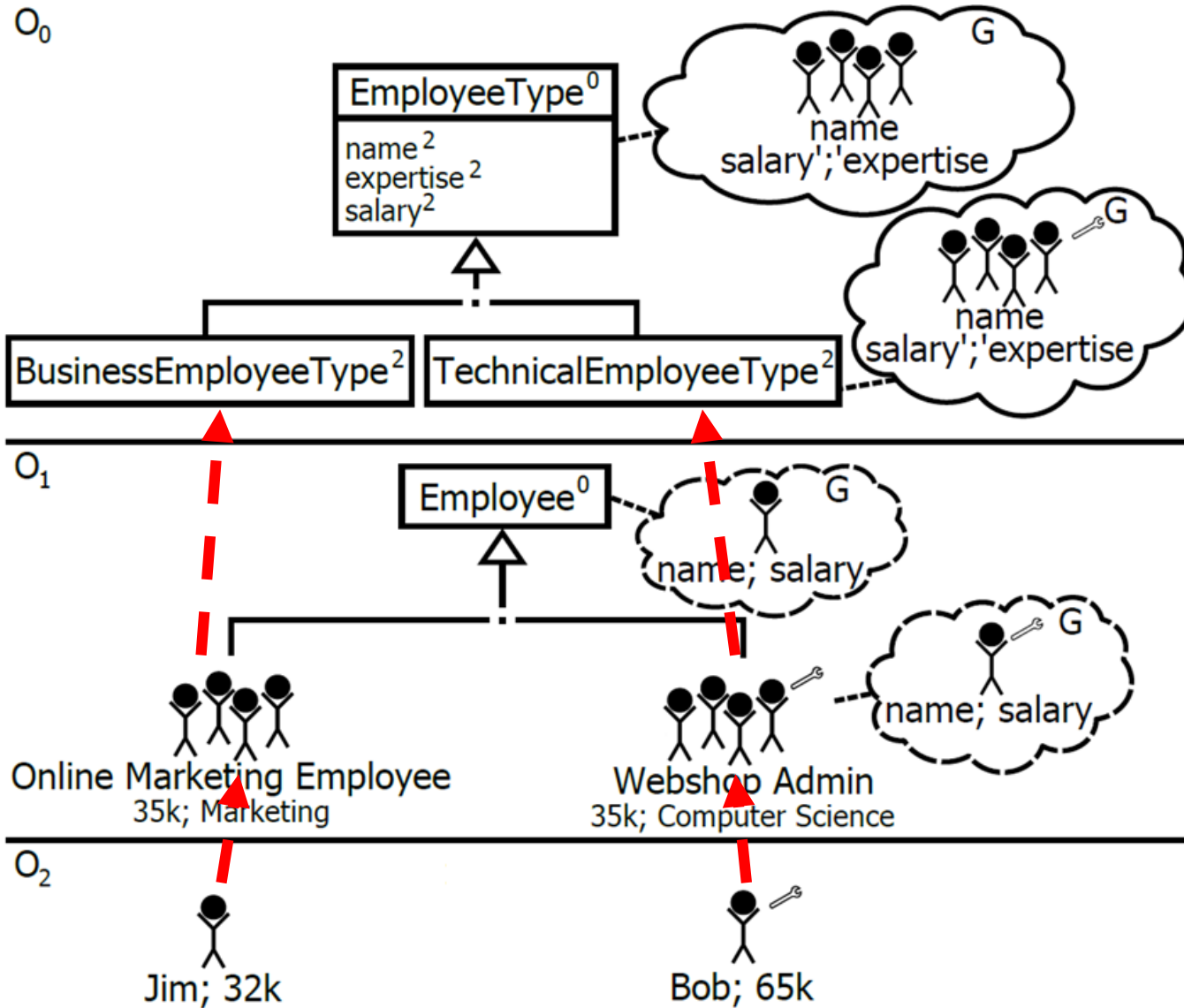
| TableLayoutInformation(grabHorizontalExcess=true...)

| ...

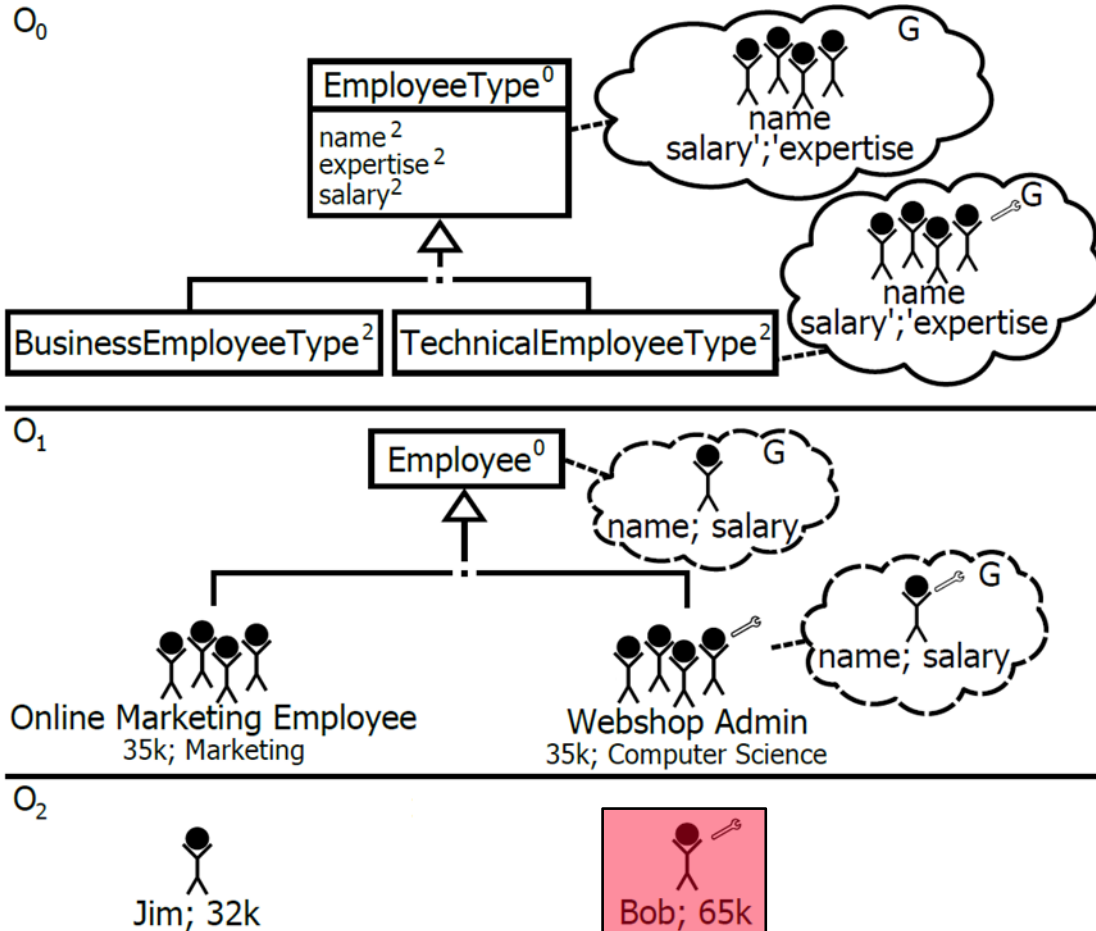


Bob; 65k

Search Algorithm



Search Algorithm



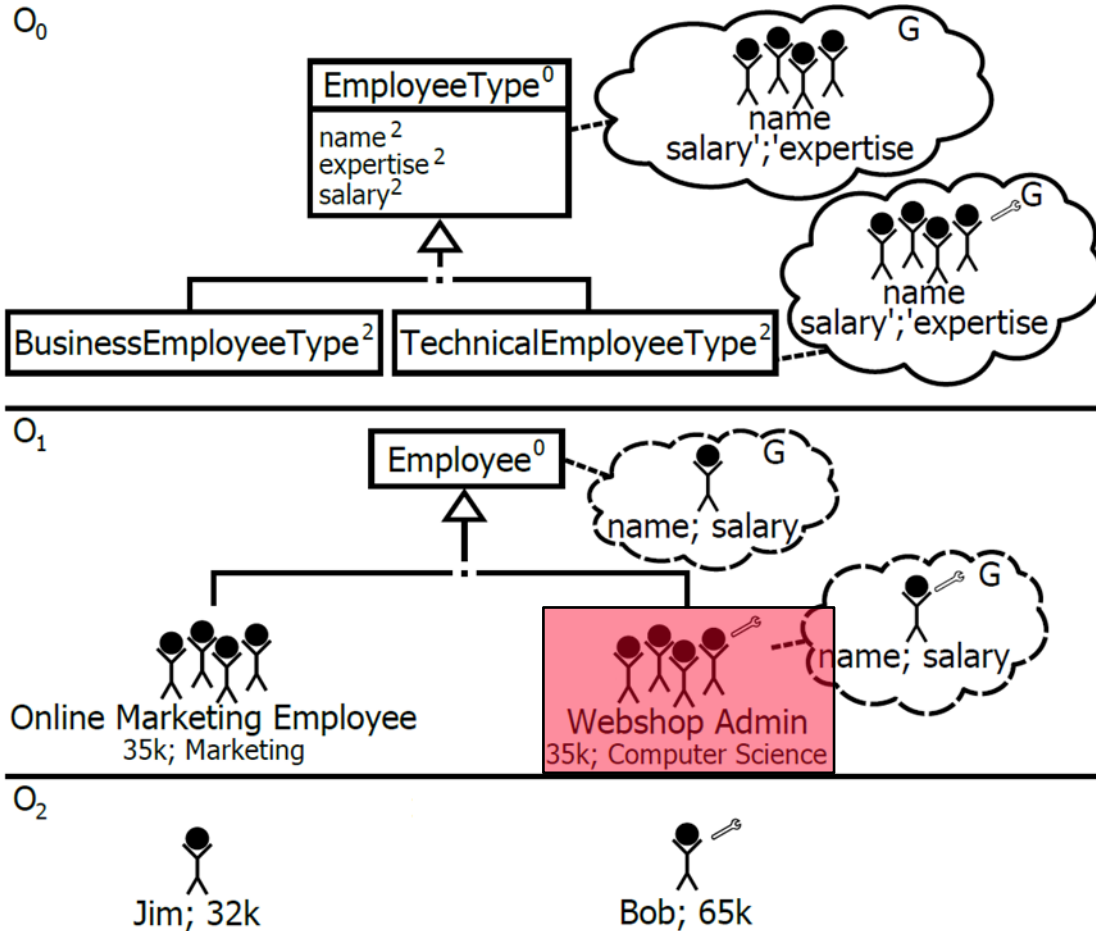
SuperTypes

{}

Types

{WebshopAdmin}

Search Algorithm



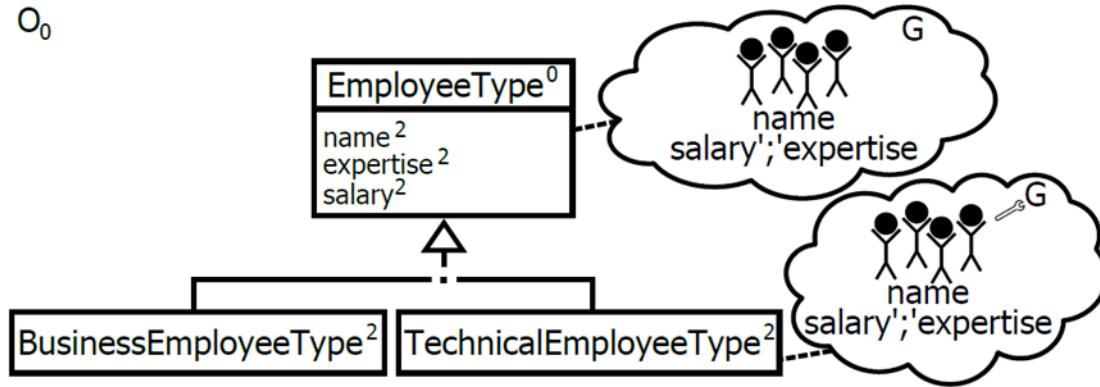
SuperTypes
{Employee}

Types
{TechnicalEmployee}

Search Algorithm



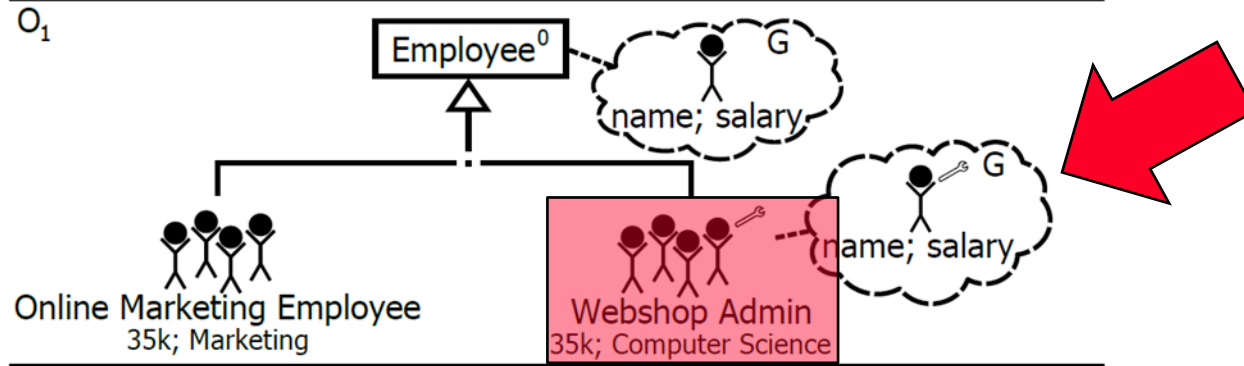
O₀



SuperTypes
{Employee}

Types
{TechnicalEmployee}

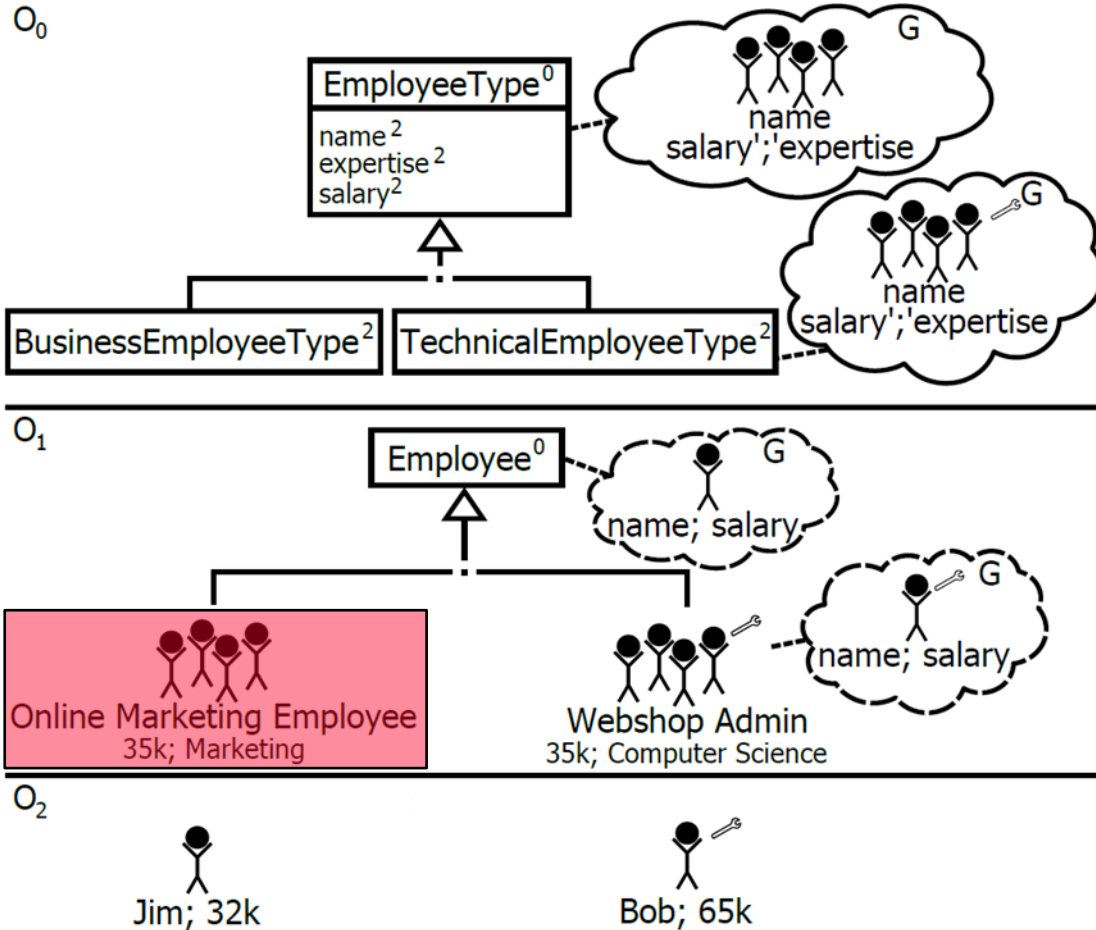
O₁



O₂

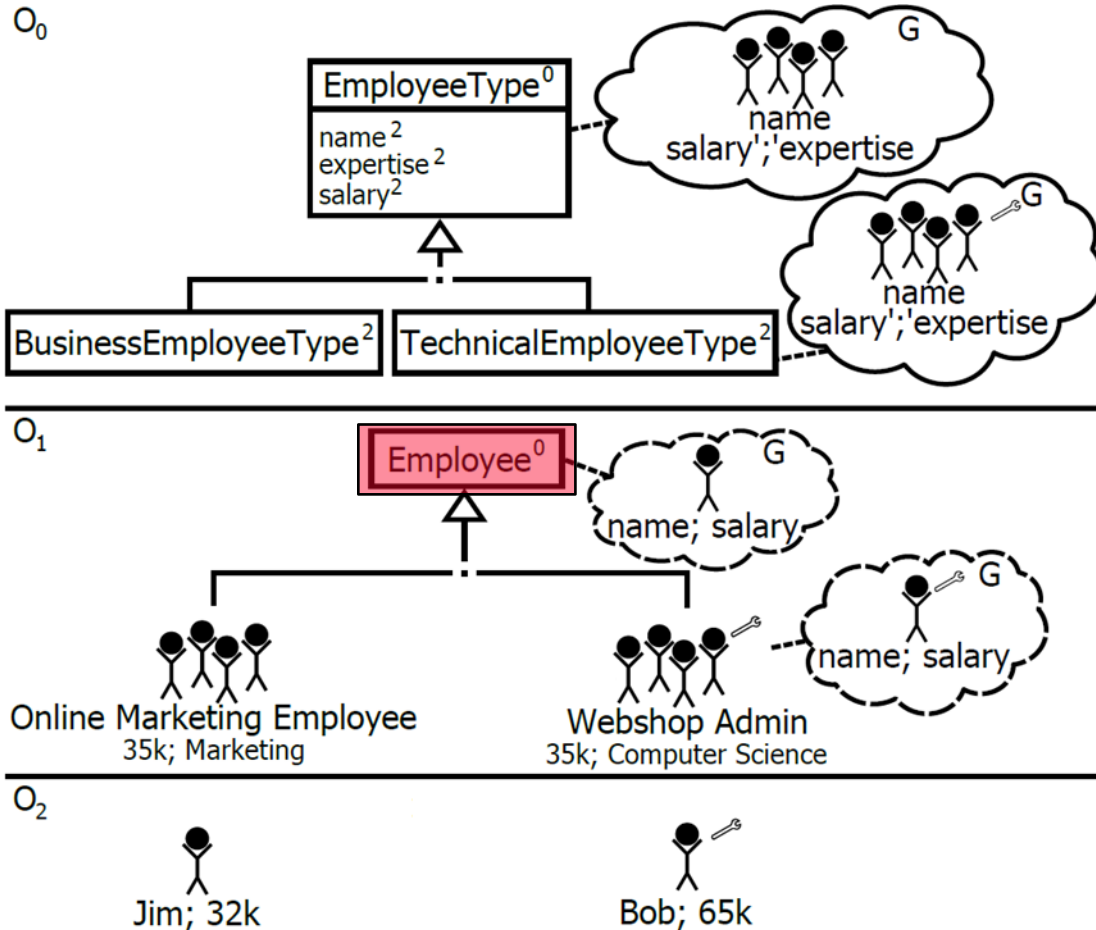


Search Algorithm



SuperTypes
{Employee}

Types
{BusinessEmployeeType}



SuperTypes

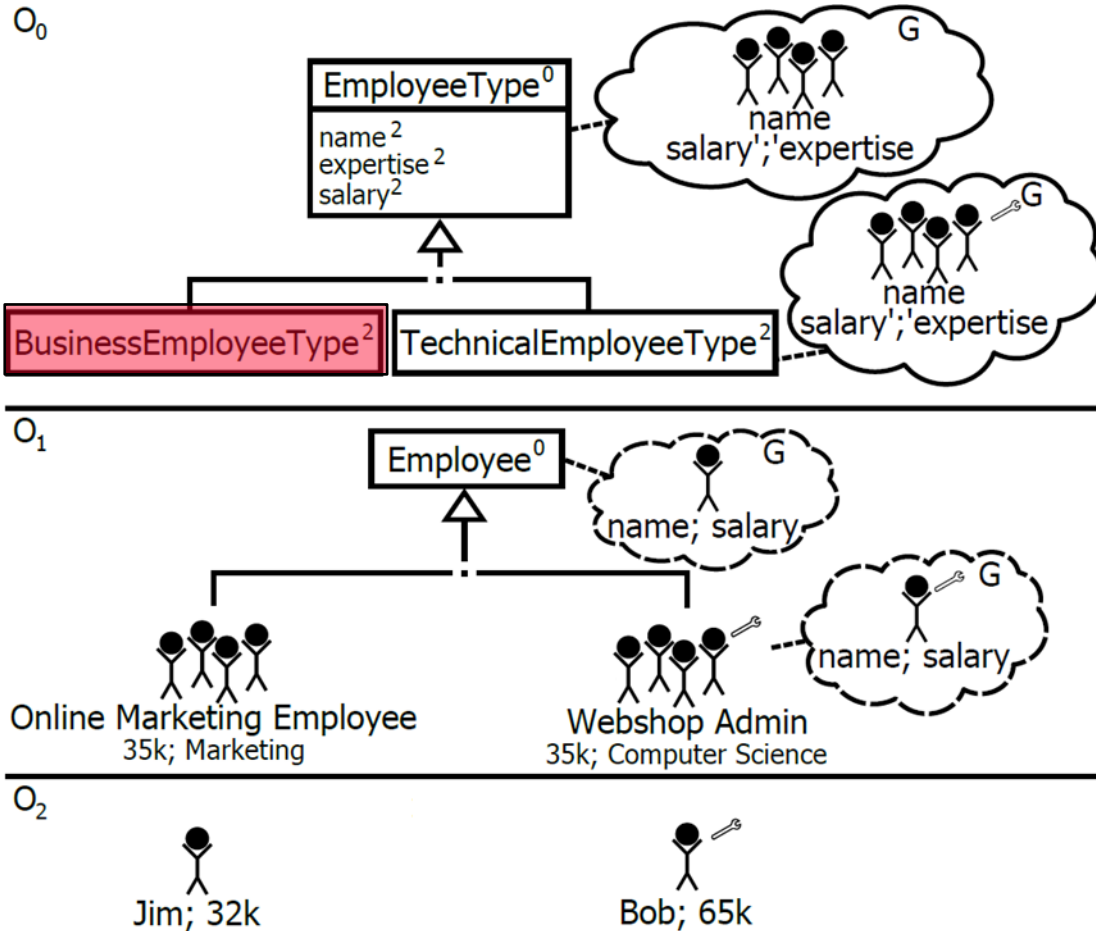
{}

Types

{BusinessEmployeeType}

The dashed cloud indicates that the visualizer is applied to instances only

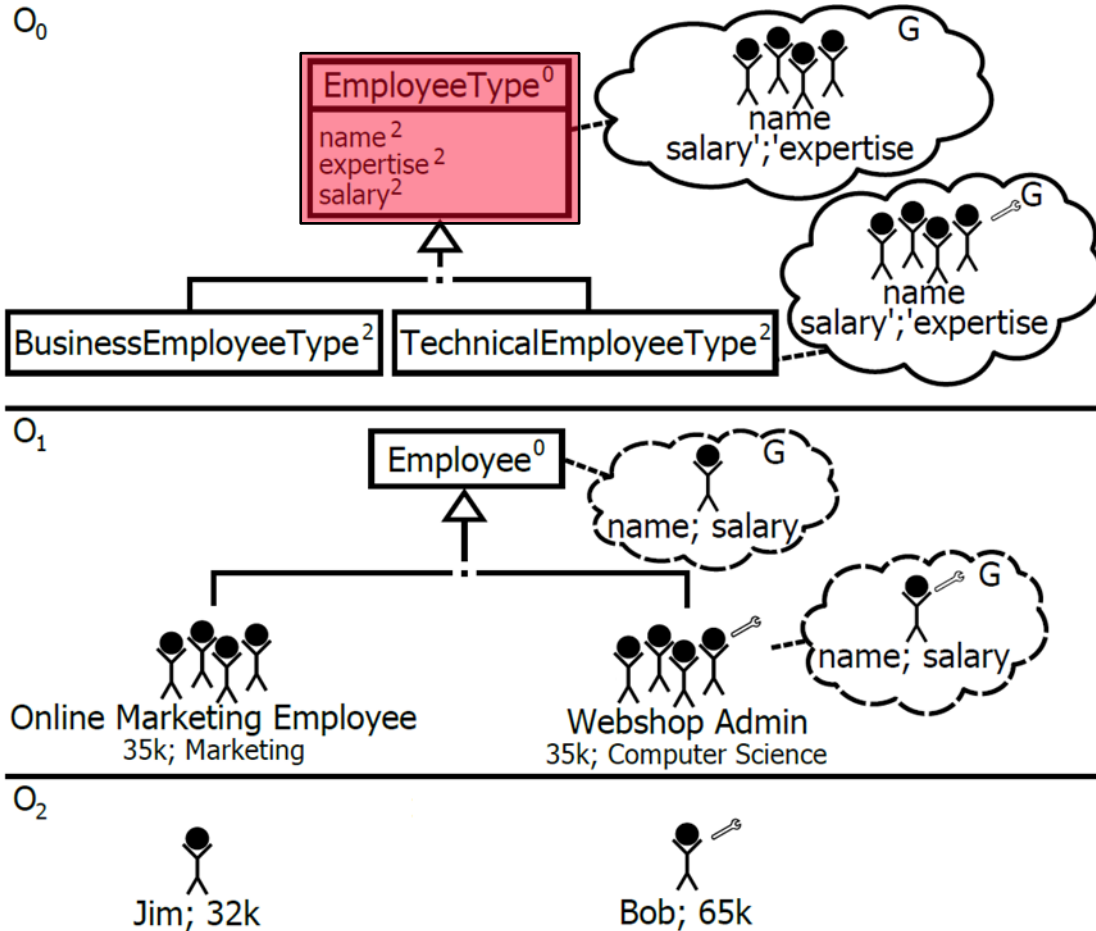
Search Algorithm



SuperTypes
{EmployeeType}

Types
{}

Search Algorithm



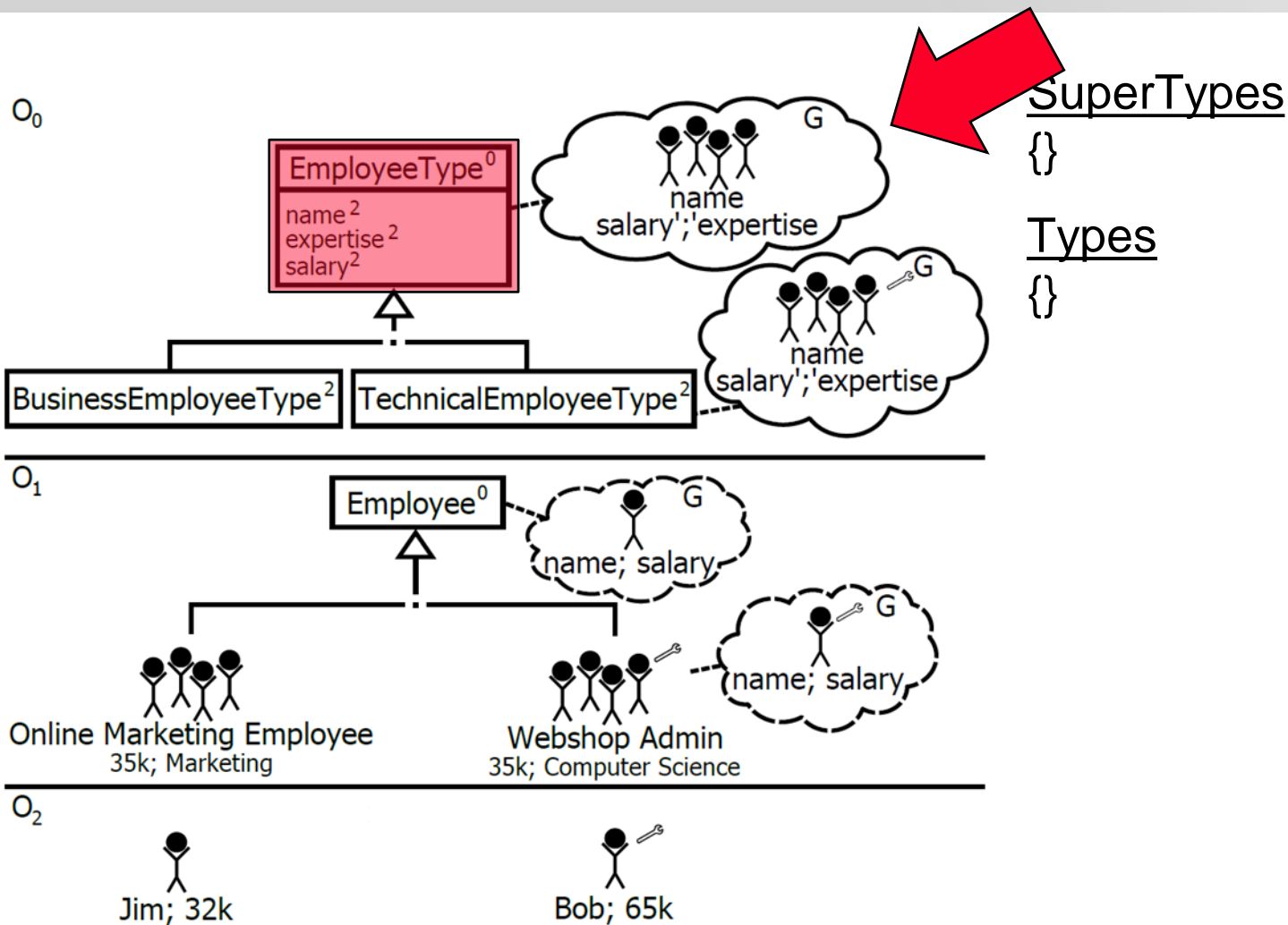
SuperTypes

{ }

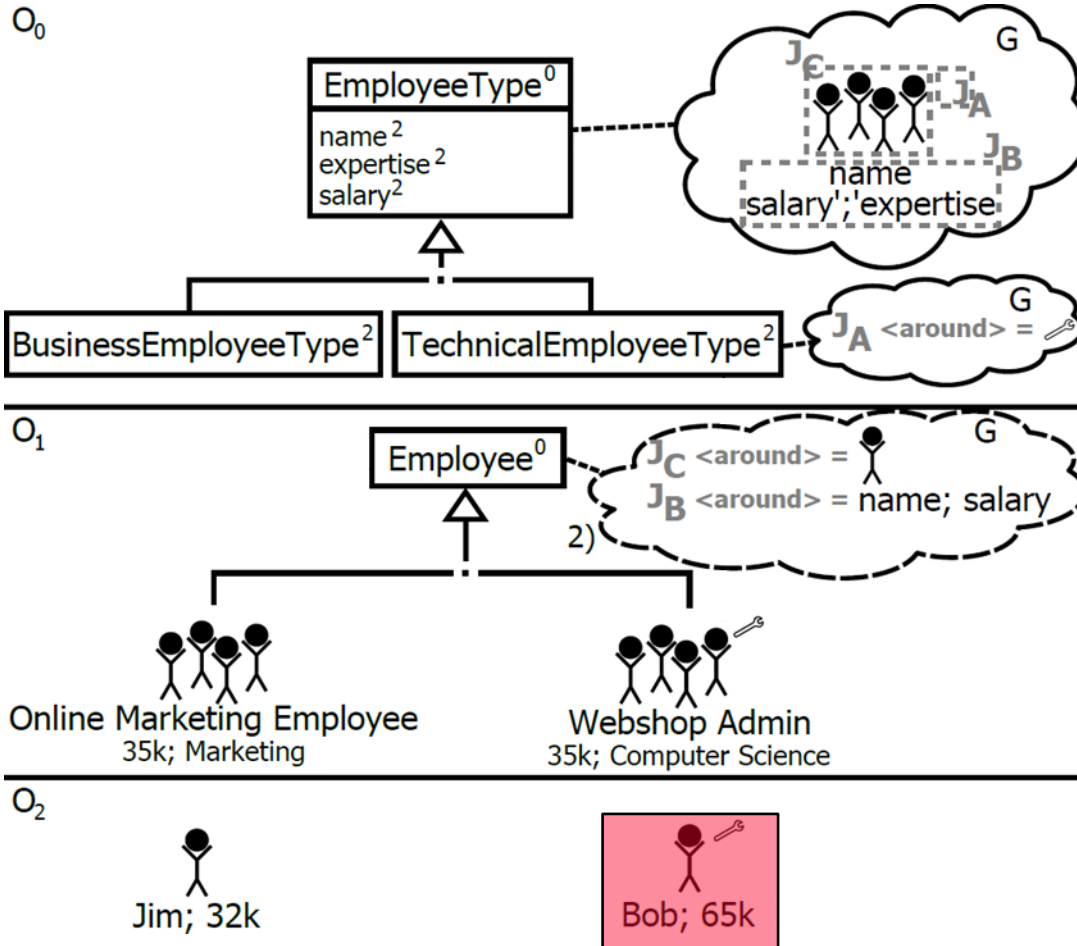
Types

{ }

Search Algorithm



Aspect Aware Search Algorithm



SuperTypes

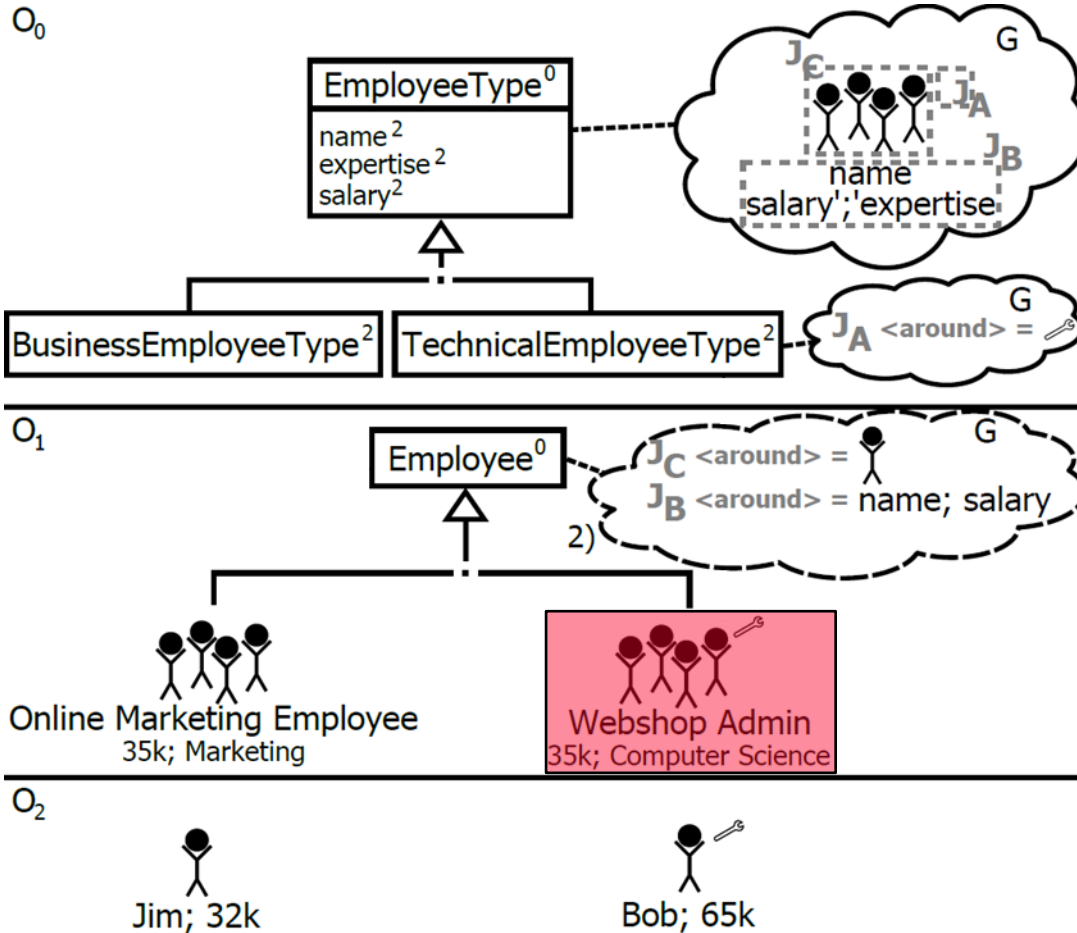
{}

Types

{WebshopAdmin}

Aspects

Aspect Aware Search Algorithm

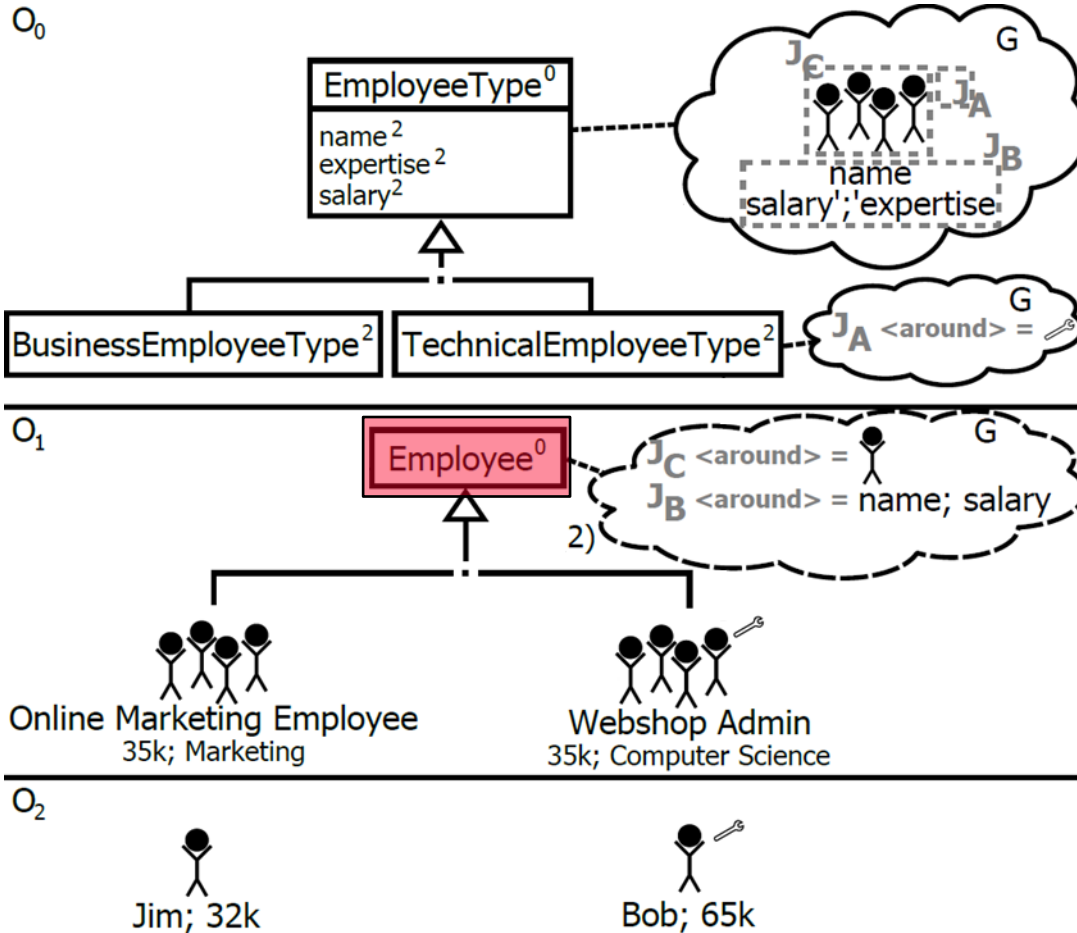


SuperTypes
{Employee}

Types
{TechnicalEmployeeType}

Aspects

Aspect Aware Search Algorithm



SuperTypes

{}

Types

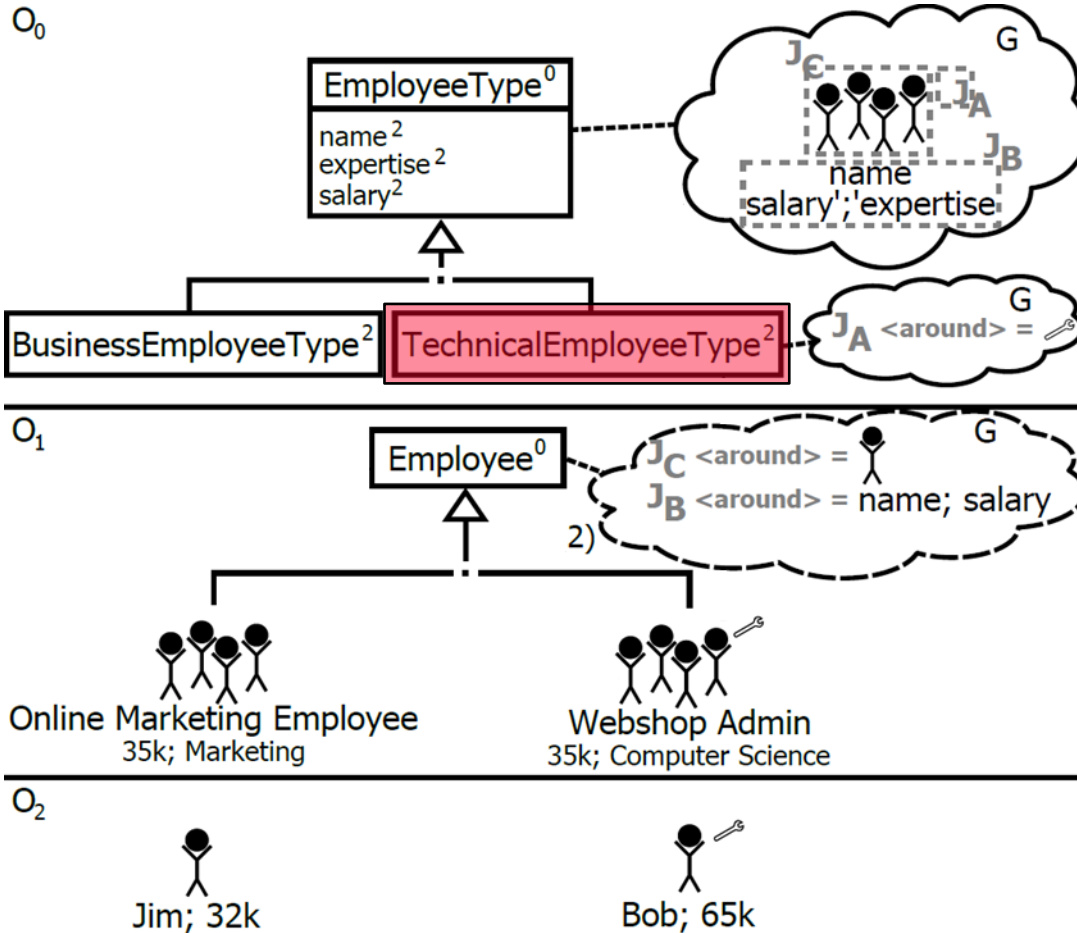
{TechnicalEmployeeType}

Aspects

J_C <around> =

J_B <around> = name; salary

Aspect Aware Search Algorithm



SuperTypes
 {EmployeeType}

Types
 {}

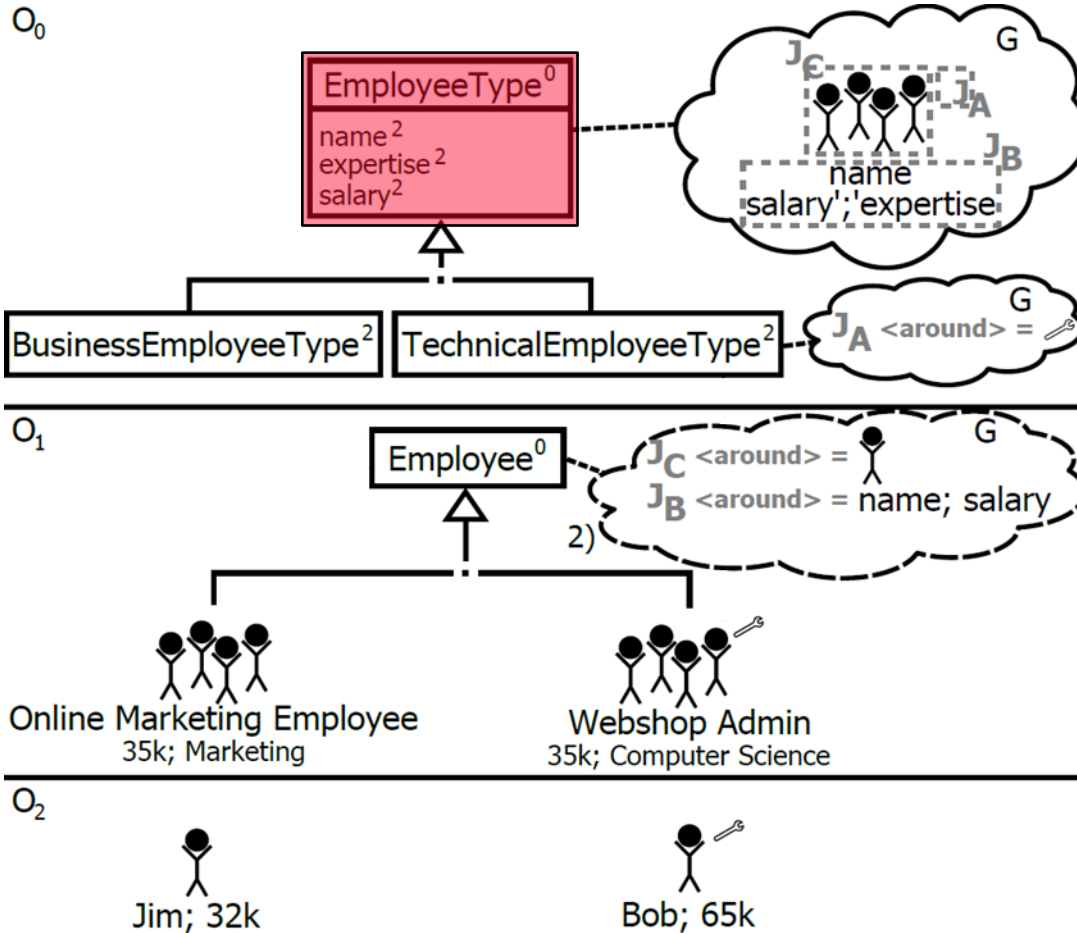
Aspects

J_C <around> =

J_B <around> = name; salary

J_A <around> =

Aspect Aware Search Algorithm



SuperTypes

{ }

Types

{ }

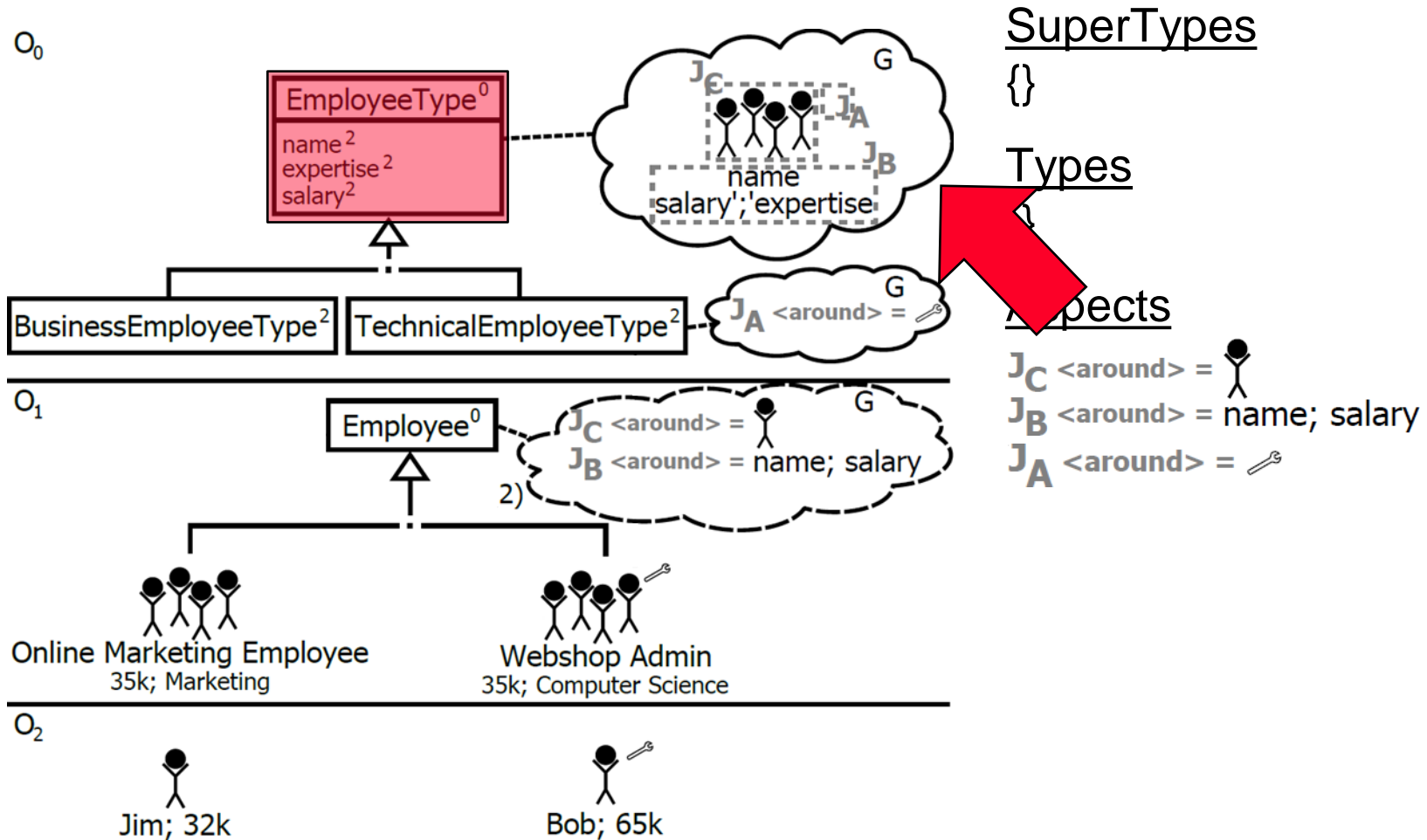
Aspects

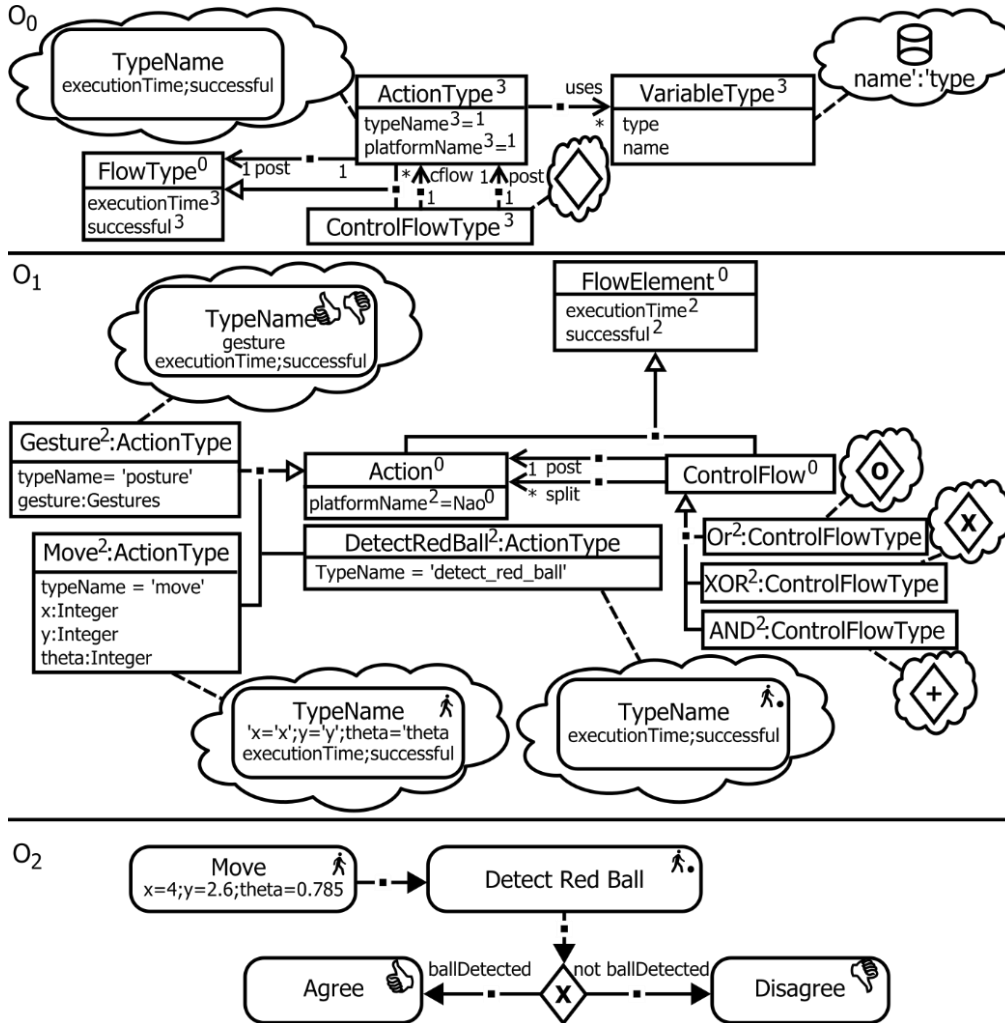
J_C <around> =

J_B <around> = name; salary

J_A <around> =

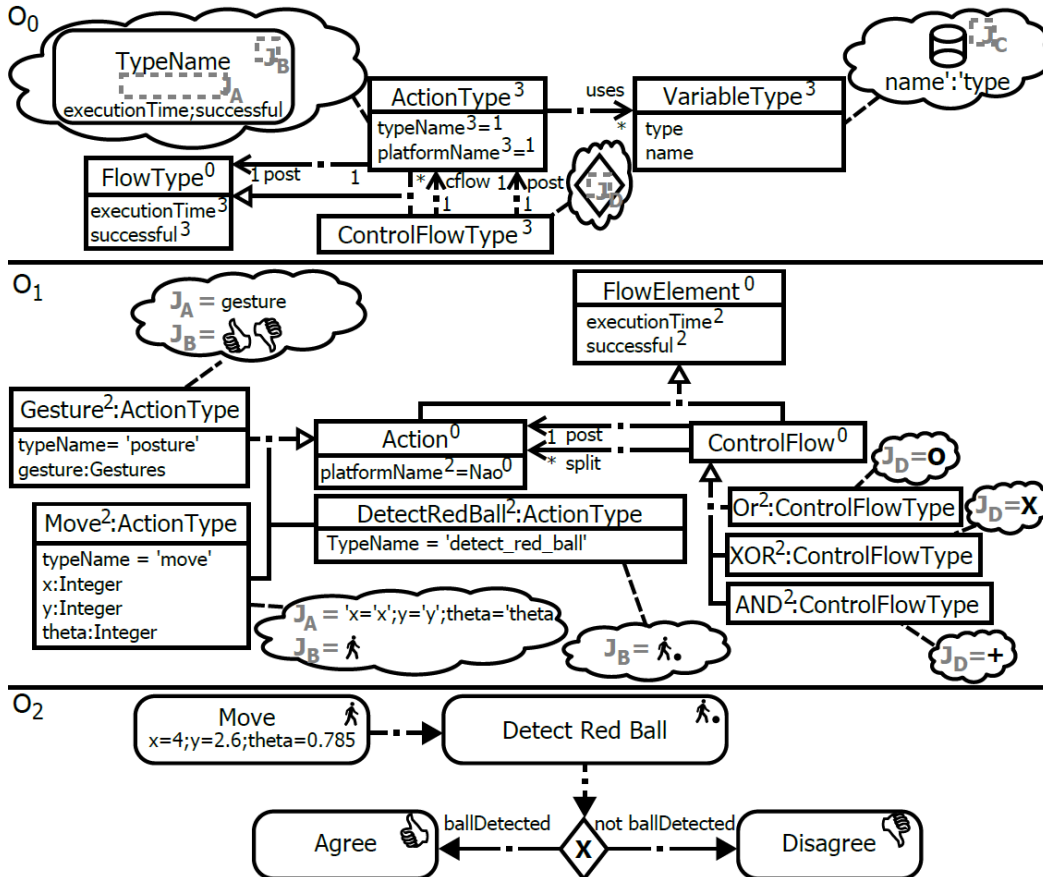
Aspect Aware Search Algorithm





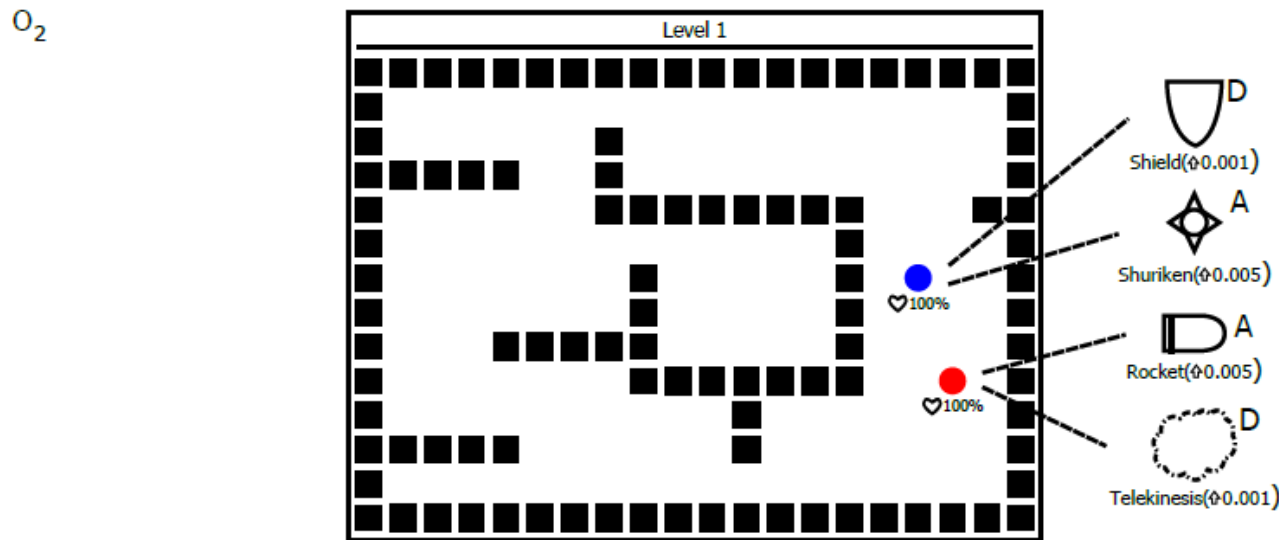
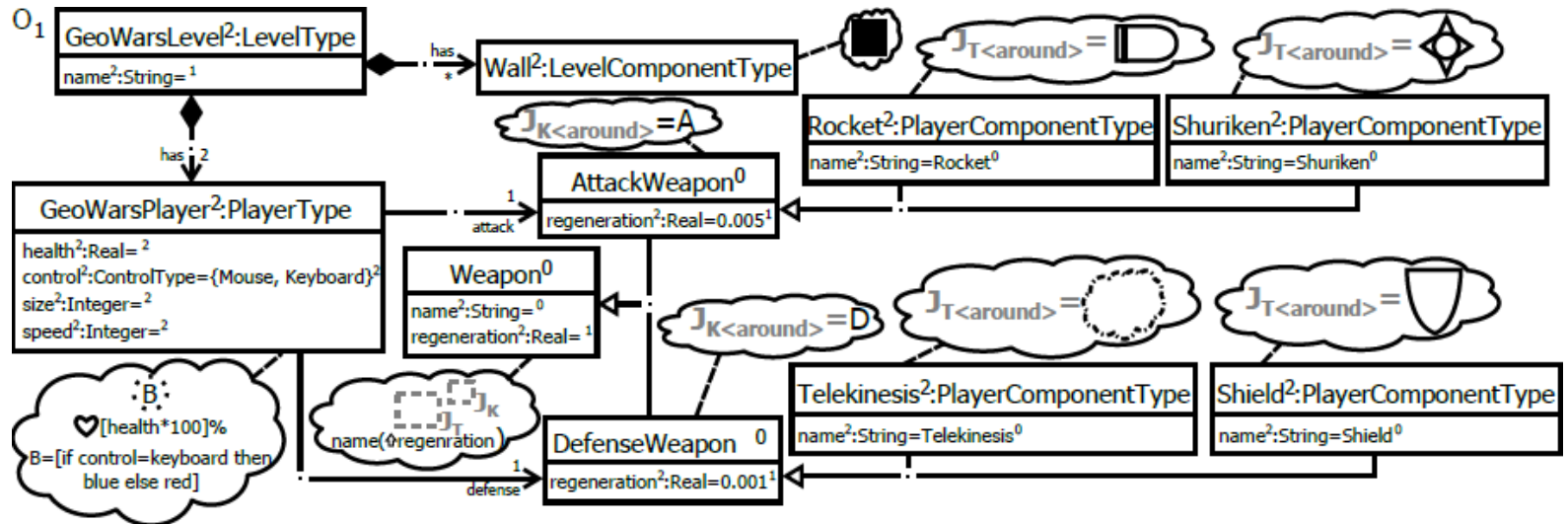
How is the following handled?

- Add a new action at O_1
 - Define the full visualization for the new action again and again
- Change general concrete syntax of actions
 - All actions at O_1 have to be modified



How is the following handled?

- Add a new action at O₁
 - Configure J_A and J_B
- Change general concrete syntax of actions
 - Nothing has to be done





- By applying aspect-oriented DSLs we could
 - Create **generic concrete** syntaxes which are refined **across classification and inheritance** hierarchies
 - **Reduce** the number of **fully defined visualizers**
 - **Reduce** the manual **effort when evolving** the generic syntax part
 - **Reduce** the amount of **work when introducing specializations** of a concrete syntax
 - Pave the way towards **central concrete syntax repositories** which can be **customized and reused**
- It is left open for future research
 - Whether comprehensibility of concrete syntax definitions is negatively impacted
 - Performance problems arise in huge models
 - If additional effort is introduced through the approach which outweighs the identified advantages.



- Visit us!
 - Melanee Webpage - <http://www.melanee.org>



Colin Atkinson

atkinson@informatik.uni-mannheim.de



Ralph Gerbig

gerbig@informatik.uni-mannheim.de