The 7th OOPSLA Workshop on
Domain-Specific Modeling

Model Driven Ecological Interface Creation: The Constraints Model

Alexandre Moïse

Jean-Marc Robert

Department of Mathematics and Industrial Engineering

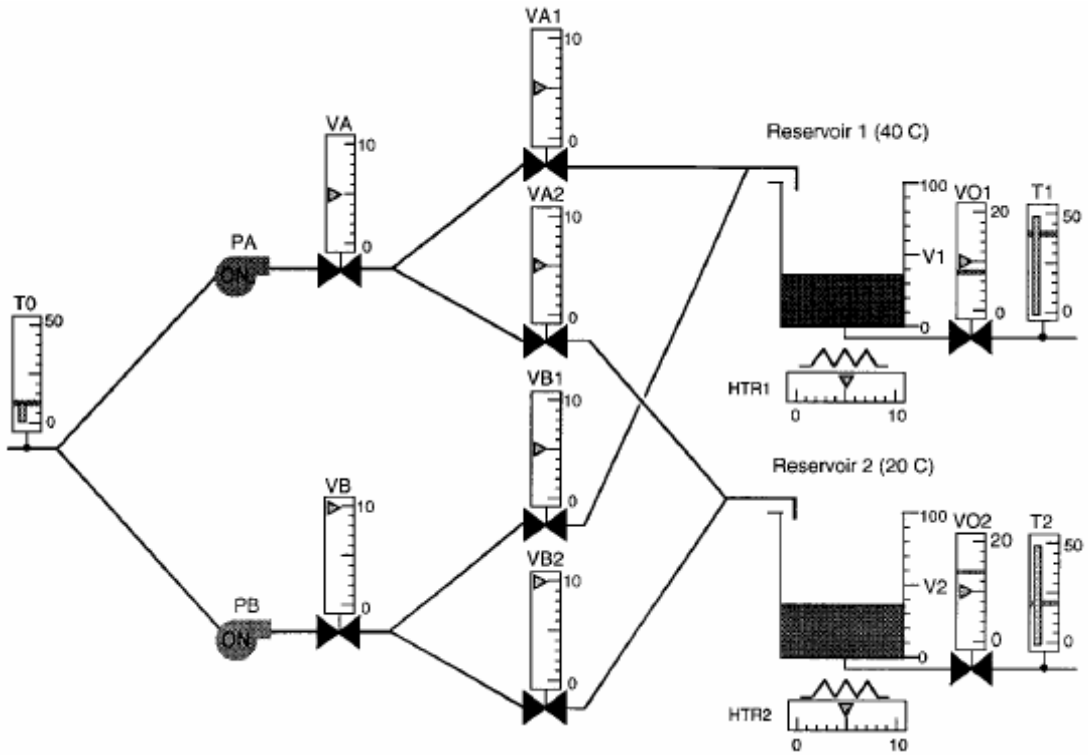
École Polytechnique de Montréal

2007.10.22

What is an ecological interface?

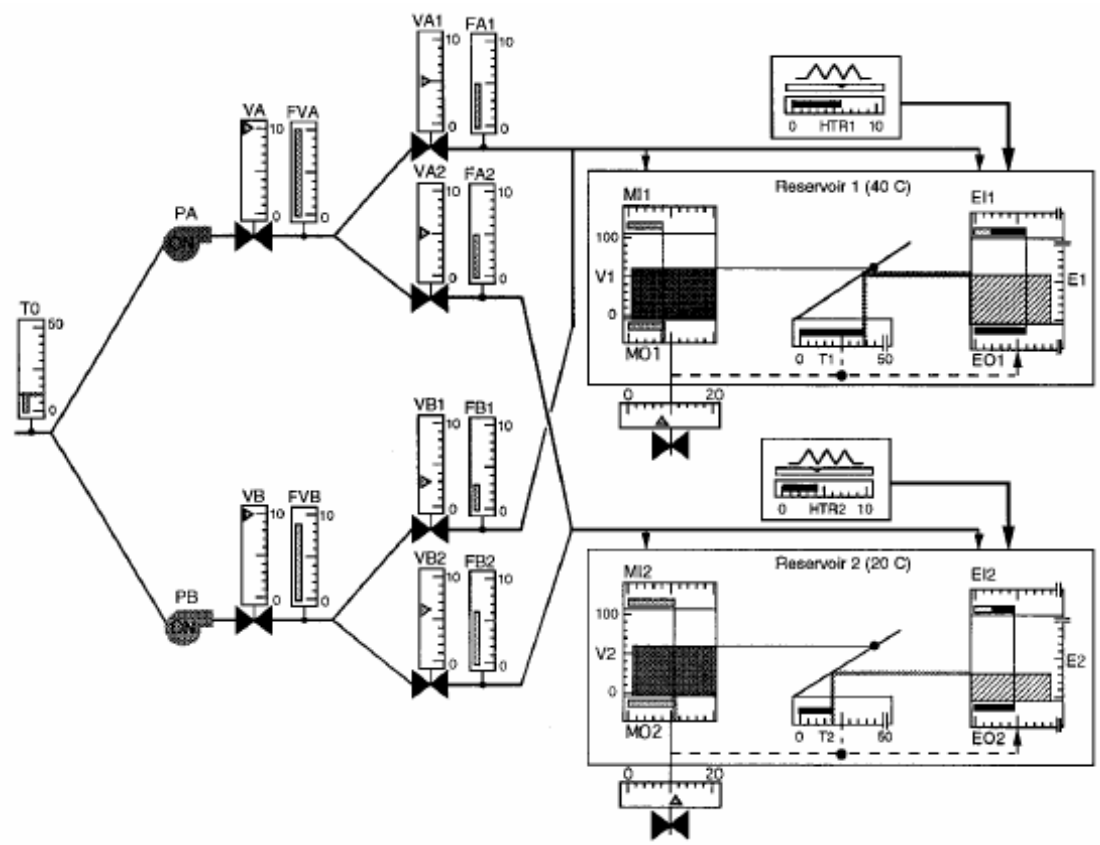
- User interface
  - Ecological psychology
  - Fault detection and diagnosis
  - Structure of an environment
    - Functional hierarchy
    - Constraints
  - Emphasis on relations
  - Direct manipulation

Traditional interface



Pawlak, W. S., Vicente, K. J. "Inducing effective operator control through ecological interface design", *Int. J. Human-Computer Studies*, vol. 44, pp. 653-688, 1996.

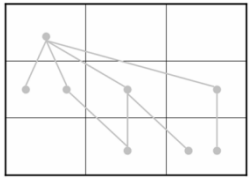
Ecological interface



Pawlak, W. S., Vicente, K. J. "Inducing effective operator control through ecological interface design", *Int. J. Human-Computer Studies*, vol. 44, pp. 653-688, 1996.

The design (and implementation) gap

Analysis model



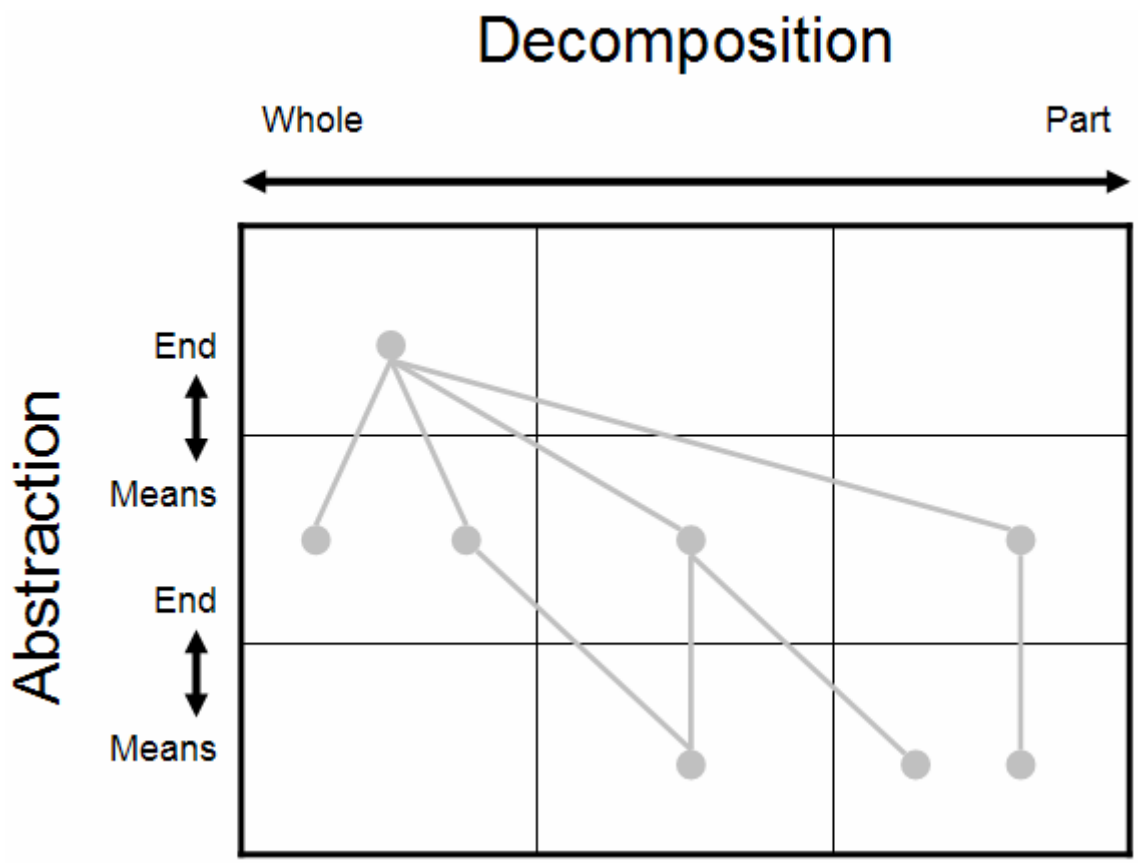
Source code

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class uneServlet extends HttpServlet {
   public void doGet(HttpServletRequest request,
HttpServletResponse response) throws
ServletException, IOException {
      // Ajouter code
   }

   public void doPost(HttpServletRequest
request, HttpServletResponse response) throws
ServletException, IOException {
      // Ajouter code
   }
}
```

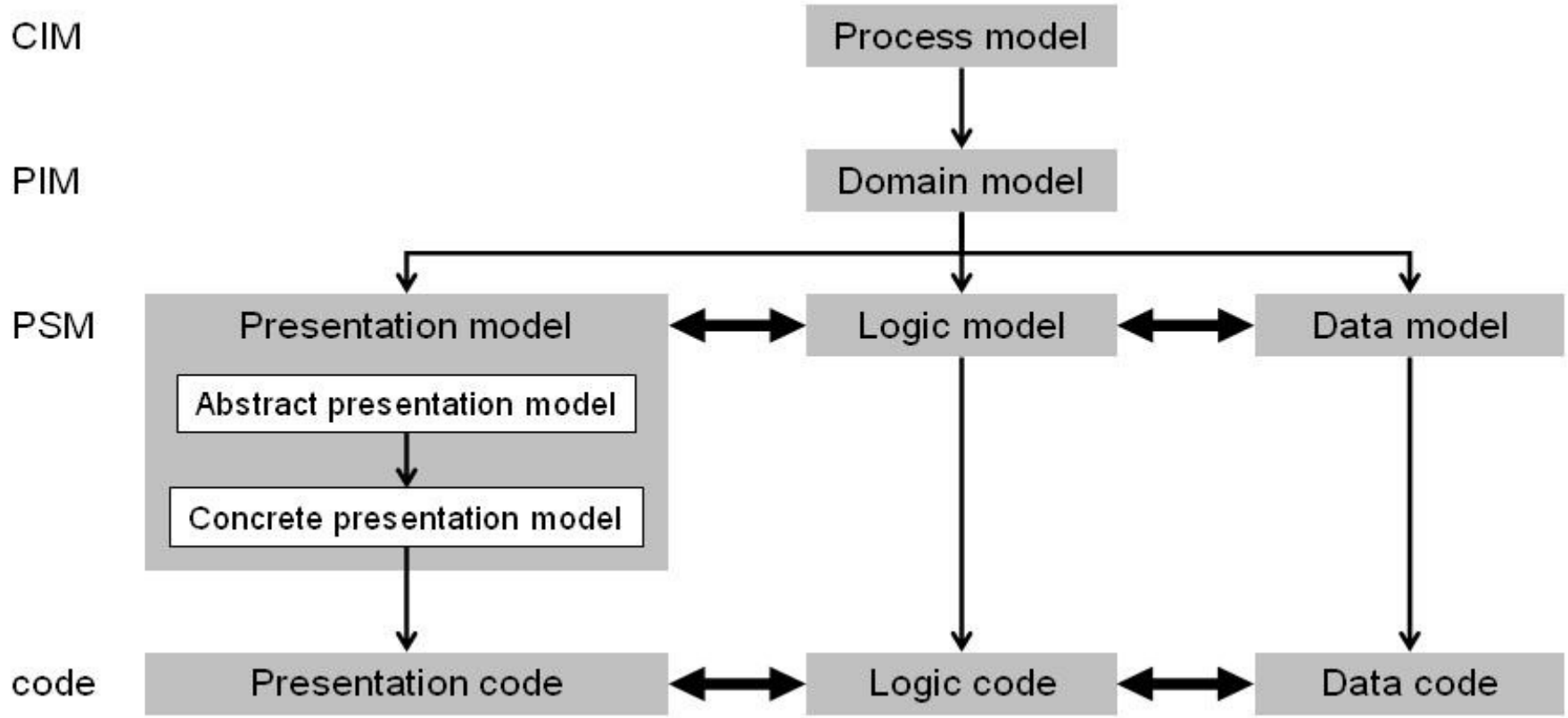
Abstraction-decomposition hierarchy



User interface modeling

- Abstract presentation
- Concrete presentation
- Final presentation

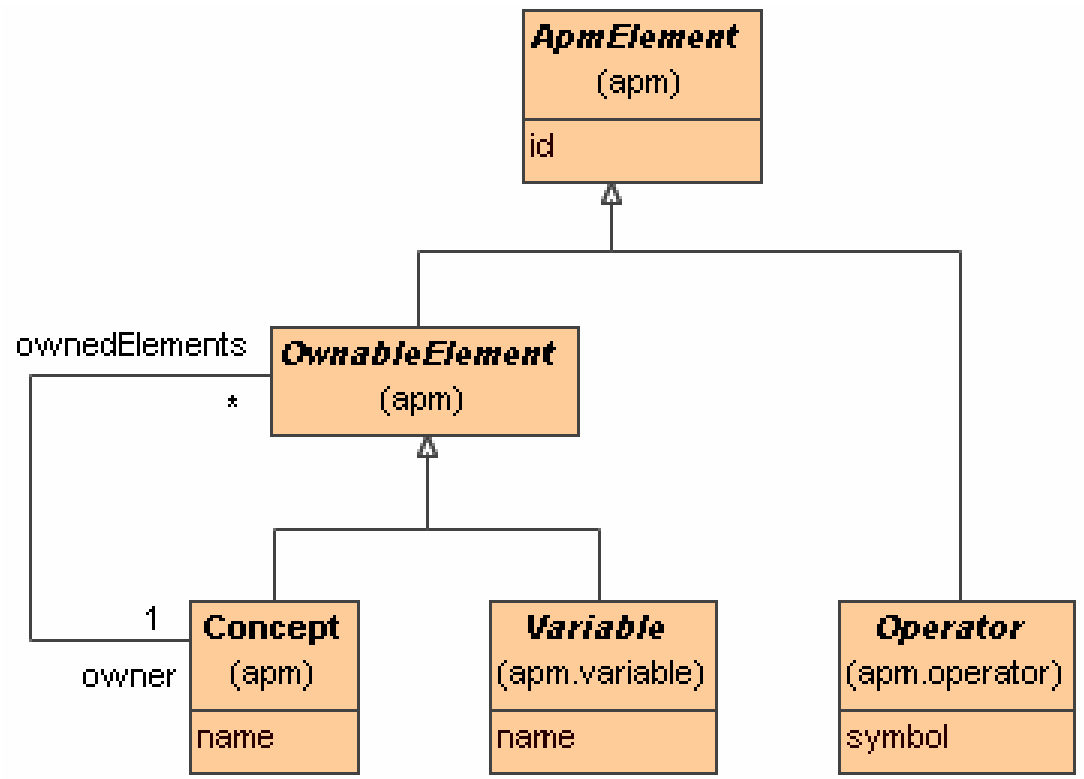
User interface modeling



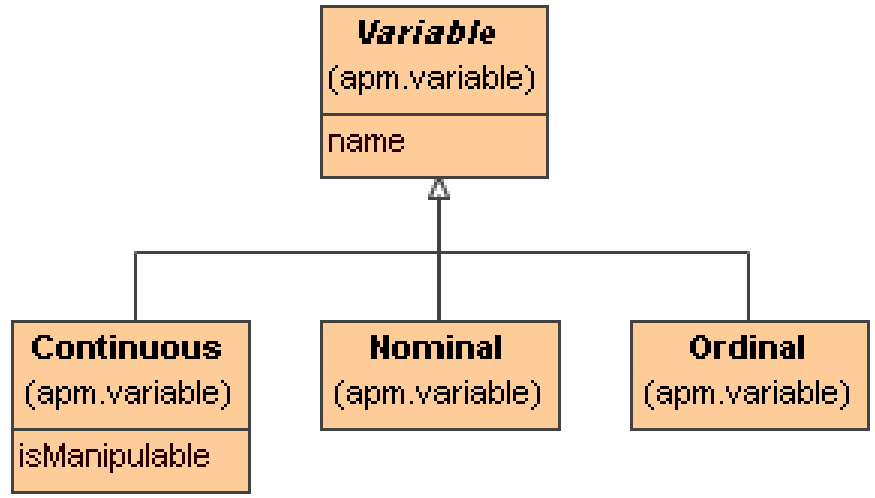
Modeling the abstract presentation

- Abstract individual component
  - Concrete form: text area, drop-down list, check box
- Abstract containter
  - Concrete form: section, grouping, frame

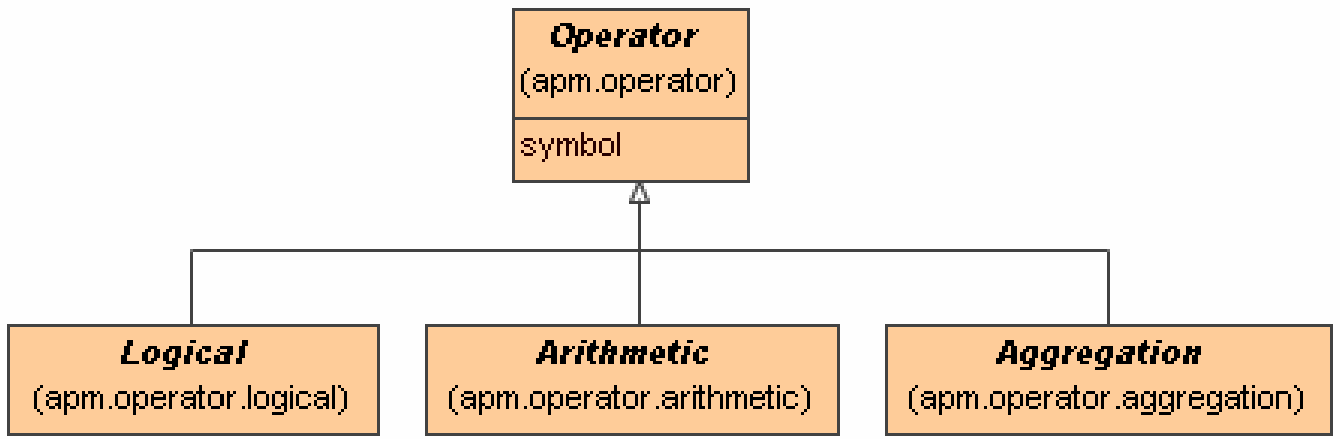
Abstract syntax



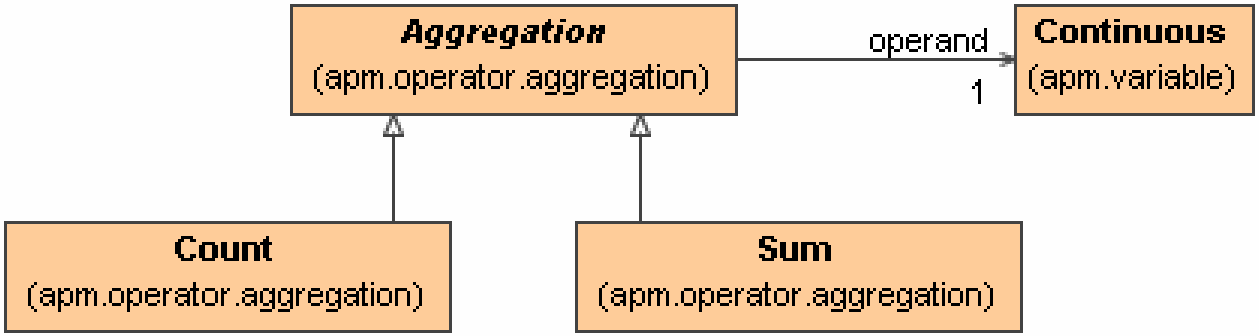
Abstract syntax



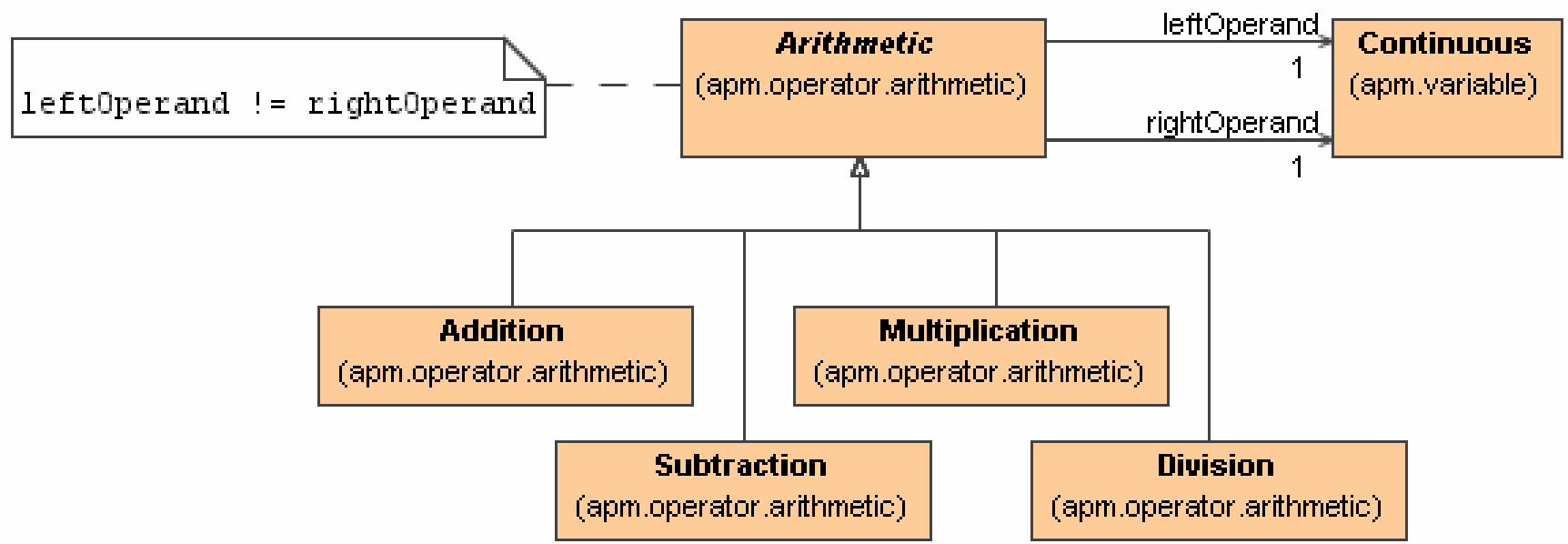
Abstract syntax



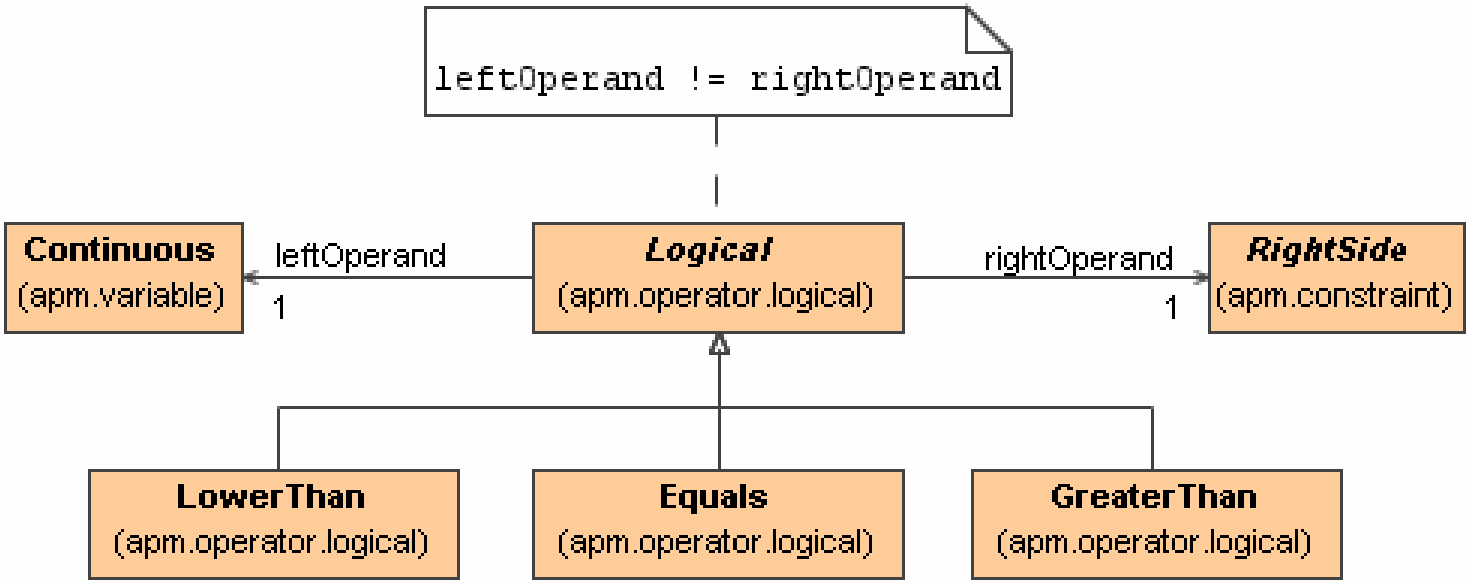
Abstract syntax



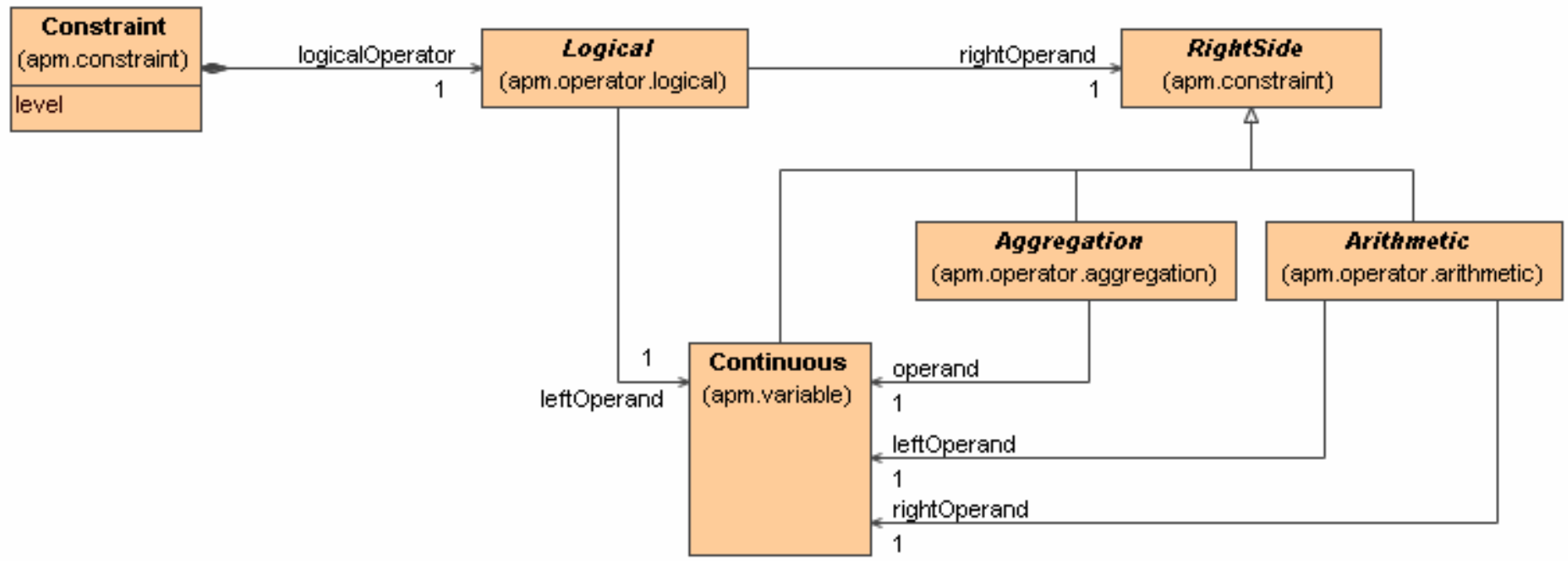
Abstract syntax



Abstract syntax



Abstract syntax

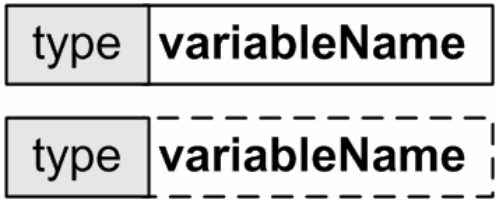


Concrete syntax

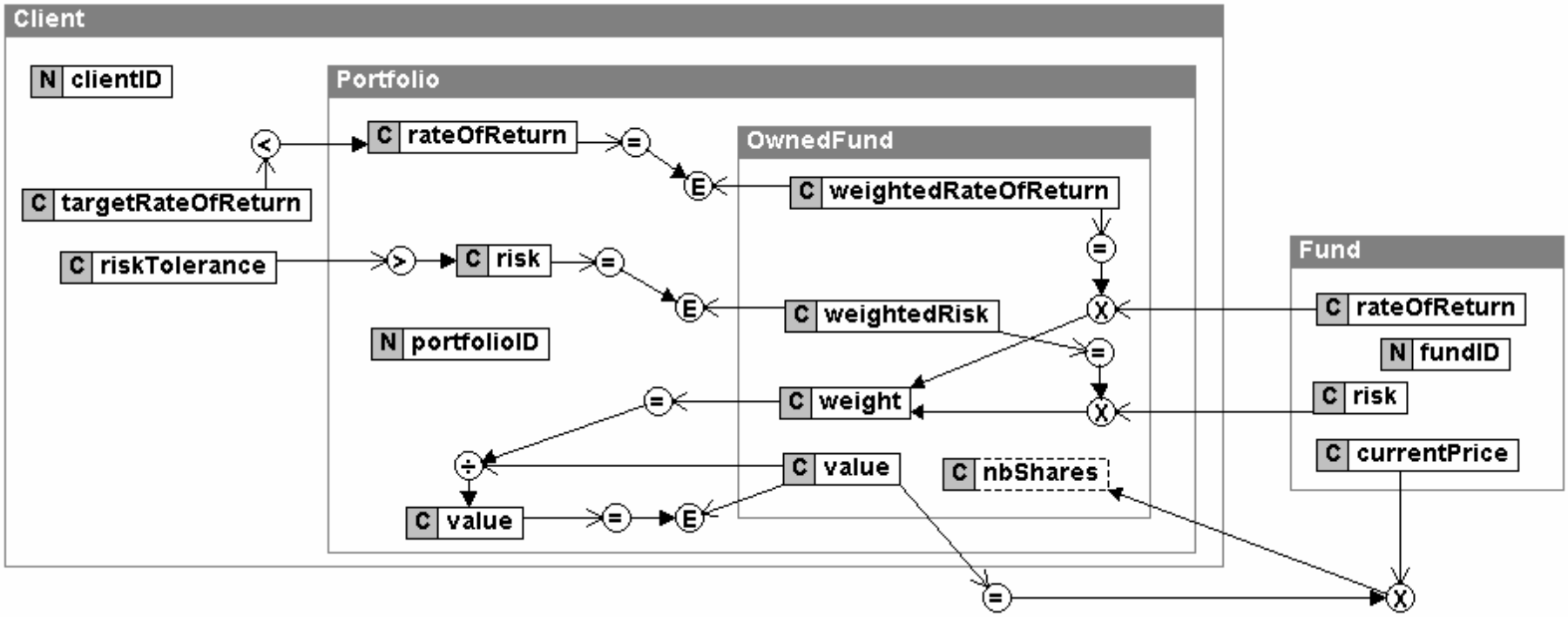
- Concept

- Operation

- Variable



Application (structure and constraints)



Application (functional hierarchy)

Client.targetRateOfReturn **<** Portfolio.rateOfReturn
Client.riskTolerance **>** Portfolio.risk

Portfolio.risk **=** **SUM** OwnedFund.weightedRisk
Portfolio.rateOfReturn **=** **SUM** OwnedFund.weightedRateOfReturn

OwnedFund.weightedRisk **=** Fund.risk **X** OwnedFund.weight
OwnedFund.weightedRateOfReturn **=** Fund.rateOfReturn **X** OwnedFund.weight

OwnedFund.weight **=** OwnedFund.value **÷** Portfolio.value

Portfolio.value **=** **SUM** OwnedFund.value
OwnedFund.value **=** Fund.currentPrice **X** OwnedFund.nbShares

Conclusion

- Another one of the few applications of DSML to UI modeling
- Better understanding of ill-defined concepts
  - Constraint, Functional hierarchy
- Future work
  - Concrete presentation, Final presentation
  - Transformations