



SPLAB
SOFTWARE PRACTICES LABORATORY

Identifying Guidelines for Constructing Metamodels

Autores:

Andreza Vieira e Franklin Ramalho

andreza@copin.ufcg.edu.br

franklin@dsc.ufcg.edu.br

Universidade Federal de Campina Grande





ROTEIRO

- Introdução e Motivação
- Objetivo
- Identificação das Diretrizes
- Avaliação
- Trabalhos Relacionados
- Considerações Finais



Introdução e Motivação

- A abordagem MDD vem sendo adotada cada vez mais
 - Casos de sucesso [OMG,2012]
- Artefatos essenciais
 - Metamodelos



Introdução e Motivação

- Metamodelagem
 - Tarefa não trivial
 - Carência de orientações
 - Dificuldade para iniciantes
 - Consequências na produtividade
 - Necessário conhecimento prévio do domínio
 - Confusão com modelagem
- Há grande necessidade de diretrizes para metamodelagem
 - Poucos trabalhos abordam diretrizes em MDD



Introdução e Motivação

- Diretrizes
 - Guias de orientação
 - Diretrizes para desenvolvimento de software
 - Boas práticas de programação em C/Java/PHP...
 - Modelagem de um domínio



Objetivo

- Identificar diretrizes para metamodelos
 - Auxiliar pessoas na metamodelagem
 - Benefícios aos metamodelos:
 - Construção
 - Compreensão
 - Manutenção
 - Reuso
 - Evolução



Identificação das Diretrizes

- Processo manual
 - Análise de metamodelos
 - Java [Barbero,2006], KobrA2 [Robin et al.,2009], UML [OMG,2012]
 - Características em comum
 - Comparação entre os elementos de cada metamodelo
 - Análise das melhores práticas para sistemas OO [Ambler, 2005]
- Definição de um template
 - Baseado no proposto por GoF [Gamma et al., 1994]



Conjunto de Diretrizes

Sigla	Nome
G1	<i>Abstracting Common Association</i>
G2	<i>Adding Abstractions Package</i>
G3	<i>Defining Boolean Attributes Default Value</i>
G4	<i>Redefining Boolean Attribute Names</i>
G5	<i>Adding Utility Operations</i>
G6	<i>Abstracting Common Attributes</i>
G7	<i>Generalizing Common Attributes</i>
G8	<i>Grouping Related Metaclasses</i>
G9	<i>Adding Primitive Types Package</i>
G10	<i>Adding Core Package</i>
G11	<i>Defining Enum Default Value</i>
G12	<i>Defining Association Member Ends Features</i>
G13	<i>Redefining Enum Names</i>



Conjunto de Diretrizes

Sigla	Nome
G1	<i>Abstracting Common Association</i>
G2	<i>Adding Abstractions Package</i>
G3	<i>Defining Boolean Attributes Default Value</i>
G4	<i>Redefining Boolean Attribute Names</i>
G5	<i>Adding Utility Operations</i>
G6	<i>Abstracting Common Attributes</i>
G7	<i>Generalizing Common Attributes</i>
G8	<i>Grouping Related Metaclasses</i>
G9	<i>Adding Primitive Types Package</i>
G10	<i>Adding Core Package</i>
G11	<i>Defining Enum Default Value</i>
G12	<i>Defining Association Member Ends Features</i>
G13	<i>Redefining Enum Names</i>



Conjunto de Diretrizes

Sigla	Nome
G1	<i>Abstracting Common Association</i>
G2	<i>Adding Abstractions Package</i>
G3	<i>Defining Boolean Attributes Default Value</i>
G4	<i>Redefining Boolean Attribute Names</i>
G5	<i>Adding Utility Operations</i>
G6	<i>Abstracting Common Attributes</i>
G7	<i>Generalizing Common Attributes</i>
G8	<i>Grouping Related Metaclasses</i>
G9	<i>Adding Primitive Types Package</i>
G10	<i>Adding Core Package</i>
G11	<i>Defining Enum Default Value</i>
G12	<i>Defining Association Member Ends Features</i>
G13	<i>Redefining Enum Names</i>



G1: Abstracting Common Associations

- Objetivo
 - Criar uma superclasse abstrata para generalizar todos os relacionamentos em comum de um metamodelo
- Motivação
 - Redundância de relacionamentos com as mesmas características
 - Dificuldade de compreensão
 - Dificuldade para alteração de um relacionamento

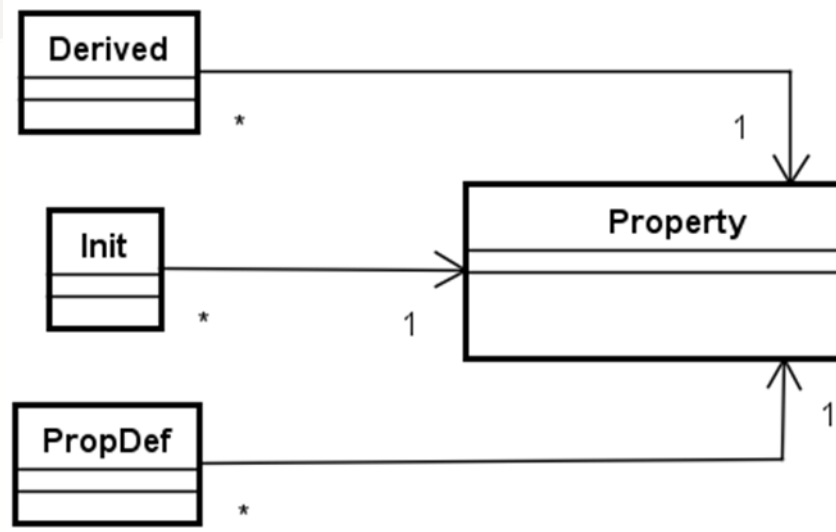


G1: Abstracting Common Associations

- Motivação

Metamodelo de Kobra2

[Robin et al.,2009]





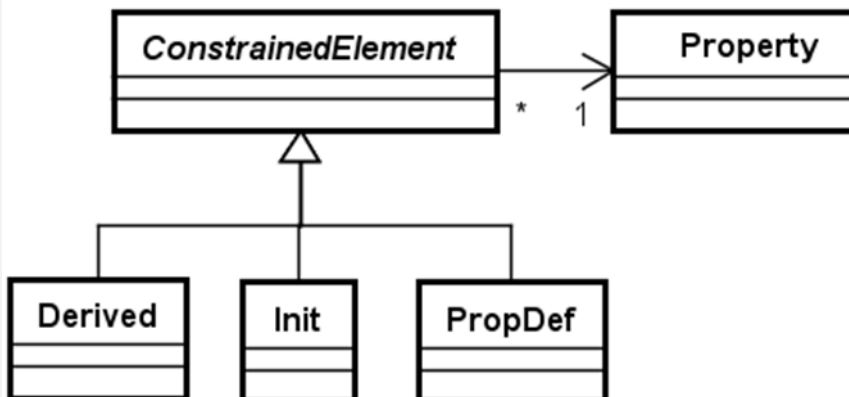
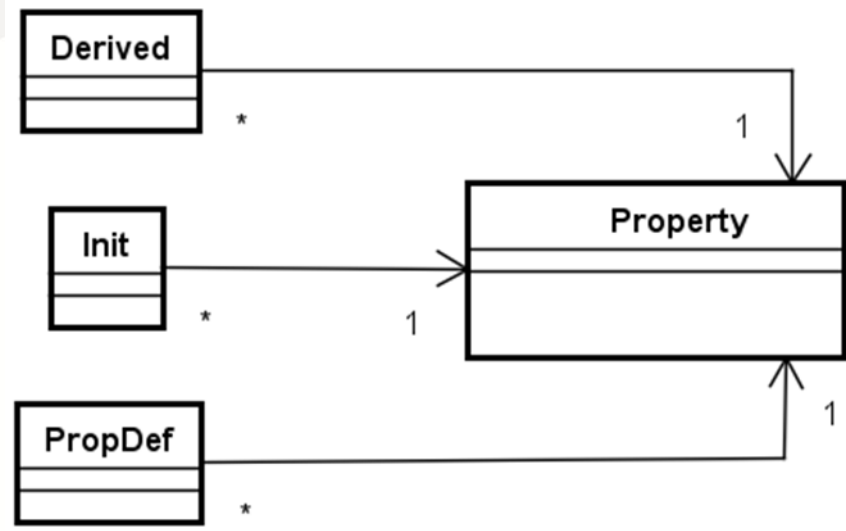
G1: Abstracting Common Associations

- Motivação

Metamodelo de Kobra2

[Robin et al.,2009]

- Solução





G1: Abstracting Common Associations

- Consequência
 - Diminuição da redundância
 - Facilidade para
 - Manutenção, compreensão, evolução e reuso
- Aplicabilidade
 - Metaclasses que possuem relacionamentos em comum
 - Mesmo nome dos relacionamentos ou papéis
 - Mesma multiplicidade
 - Mesmas características



G2: Adding Abstractions Package

- Objetivo
 - Definir um pacote *Abstractions* no metamodelo para conter todas as metaclasses abstratas reutilizáveis, assim, facilitando a localização de tais metaclasses
- Motivação
 - Má estruturação e organização das metaclasses
 - Dificuldade de compreensão
 - Dificuldade de manutenção
 - Dificuldade para reuso de partes do metamodelo



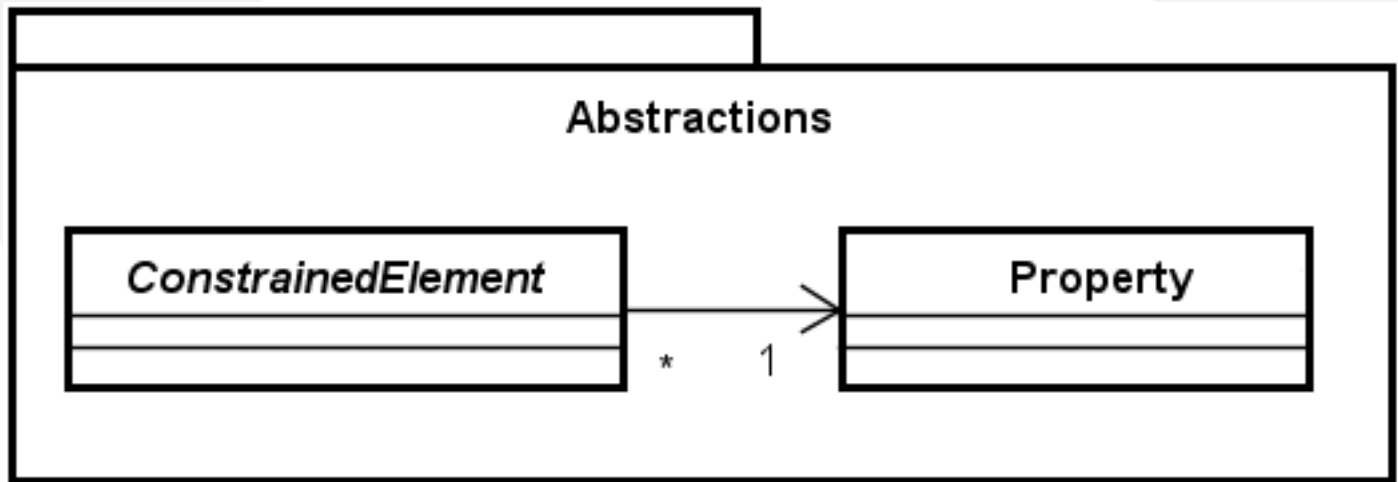
G2: Adding Abstractions Package

- Solução
 - Criar um pacote *Abstractions* no metamodelo contendo as metaclasses reutilizáveis
- Consequência
 - Melhor estruturação e organização
 - Facilidade para
 - Manutenção, compreensão, evolução e reuso
- Aplicabilidade
 - Qualquer metamodelo



G2: Adding Abstractions Package

- Exemplo
 - Metamodelo de KobrA2





G5: Adding Utility Operations

- **Objetivo**
 - Definir operações adicionais para evitar a repetição de expressões OCL que ocorrem mais de uma vez em restrições de metaclasses
- **Motivação**
 - Definição de uma mesma expressão OCL em diferentes partes do metamodelo para uma mesma finalidade
 - Dificuldade de entendimento e manutenção
 - Metamodelo propício a erros



G5: Adding Utility Operations

- Motivação
 - Metamodelo KobrA2

```
[1] context StructuralElement
inv: let kindAllowed = Set{TypeElement, InstanceElement}
in (kindAllowed->exists(e: Element | self.oclIsKindOf(e)))
```

```
[2] context InstanceElement
inv: let kindAllowed = Set{ComponentObject, Slot, Link}
in (kindAllowed->exists(e: Element | self.oclIsKindOf(e)))
```

```
[3] context StructuralConstraintElement
inv: let kindAllowed = Set{Inv, Derived, Init, PropDef}
in (kindAllowed->exists(e: Element | self.oclIsKindOf(e)))
```



G5: Adding Utility Operations

- Motivação
 - Metamodelo KobrA2

```
[1] context StructuralElement
inv: let kindAllowed = Set{TypeElement, InstanceElement}
in (kindAllowed->exists(e: Element | self.ocIsKindOf(e)))
```

```
[2] context InstanceElement
inv: let kindAllowed = Set{ComponentObject, Slot, Link}
in (kindAllowed->exists(e: Element | self.ocIsKindOf(e)))
```

```
[3] context StructuralConstraintElement
inv: let kindAllowed = Set{Inv, Derived, Init, PropDef}
in (kindAllowed->exists(e: Element | self.ocIsKindOf(e)))
```



G5: Adding Utility Operations

- Solução
 - Criar uma operação adicional para cada expressão utilizada com frequência no metamodelo

```
context Element
def: existsElement(set: Set(OclAny)) : Boolean =
set->exists(e: Element | self.oclIsKindOf(e))
```

Operação
criada



G5: Adding Utility Operations

- Solução
 - Criar uma operação adicional para cada expressão utilizada com frequência no metamodelo

```
context Element
def: existsElement(set: Set(OclAny)) : Boolean =
set->exists(e: Element | self.oclIsKindOf(e))
```

Operação
criada

```
[1] context StructuralElement
inv: let kindAllowed = Set{TypeElement, InstanceElement}
in self.existsElement(kindAllowed)
```



G5: Adding Utility Operations

- Consequência
 - Diminuição da repetição e de possíveis erros
 - Facilidade para
 - Manutenção, compreensão e reuso
- Aplicabilidade
 - Quando há a repetição de expressões nas restrições do metamodelo



Avaliação

- Baseada no método GQM (*Goal-Question-Metric*) [Basili et al., 2001]
- Conjunto de metamodelos avaliados
 - QVT, SPEM e OCL (OMG) [OMG,2012]
 - Java [Barbero, 2006], Ant [Ruchaud et al., 2005] e Kobra2 [Robin et al., 2009] (Outras organizações)
- Objetivo
 - Verificar a aplicabilidade de cada uma das diretrizes
- Questão
 - Quão aplicáveis são as diretrizes aos metamodelos?



i: diretriz avaliada

j: conjunto de metamodelos

AN: número de aplicações da diretriz

m: número de metamodelos avaliados

AGN: número de diretrizes aplicadas

Avaliação

- Métrica AAG_i (*Applications Average of a Guideline i*)

$$AAG_i = \sum_{j=1}^m AN_{i,j} / m$$

- Métrica AGA : (*Applied Guidelines Average*)

$$AGA = \sum_{j=1}^m AGN_j / m$$



Avaliação

Diretriz / Metamodelo	<i>KobrA2</i>	<i>Ant</i>	<i>Java</i>	<i>OCL</i>	<i>QVT</i>	<i>SPEM</i>	<i>Total</i>	<i>AAG_i</i>
<i>G1</i>	2	4	18	0	2	2	28	4.6
<i>G2</i>	1	1	1	0	1	1	5	0.8
<i>G3</i>	12	0	29	1	12	2	56	9.3
<i>G4</i>	7	0	29	0	2	1	39	6.5
<i>G5</i>	22	0	0	0	0	0	22	3.6

G1: Abstracting Common Associations

G2: Adding Abstractions Package

G3: Defining Boolean Attributes Default Value

G4: Redefining Boolean Attribute Names

G5: Adding Utility Operations



Avaliação

Diretriz / Metamodelo	<i>KobrA2</i>	<i>Ant</i>	<i>Java</i>	<i>OCL</i>	<i>QVT</i>	<i>SPEM</i>	<i>Total</i>	<i>AAG_i</i>
<i>G1</i>	2	4	18	0	2	2	28	4.6
<i>G2</i>	1	1	1	0	1	1	5	0.8
<i>G3</i>	12	0	29	1	12	2	56	9.3
<i>G4</i>	7	0	29	0	2	1	39	6.5
<i>G5</i>	22	0	0	0	0	0	22	3.6

G2: Adding Abstractions Package



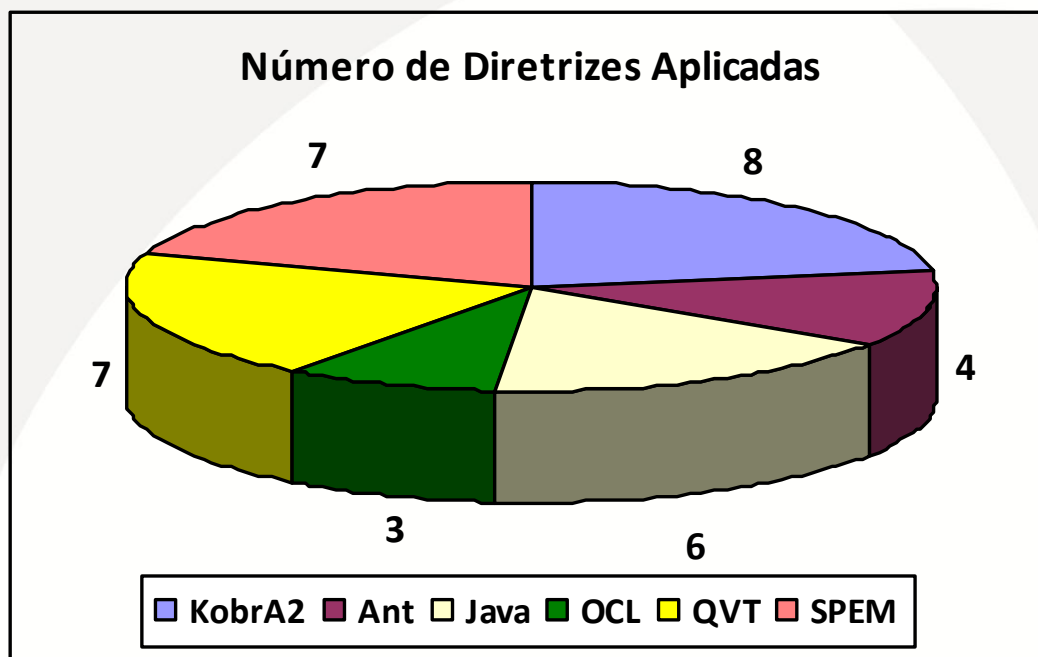
Avaliação

Diretriz / Metamodelo	<i>KobrA2</i>	<i>Ant</i>	<i>Java</i>	<i>OCL</i>	<i>QVT</i>	<i>SPEM</i>	<i>Total</i>	<i>AAG_i</i>
<i>G1</i>	2	4	18	0	2	2	28	4.6
<i>G2</i>	1	1	1	0	1	1	5	0.8
<i>G3</i>	12	0	29	1	12	2	56	9.3
<i>G4</i>	7	0	29	0	2	1	39	6.5
<i>G5</i>	22	0	0	0	0	0	22	3.6

G3: Defining Boolean Attributes Default Value



Avaliação



Média de diretrizes aplicadas (AGA) = **5.8**



Trabalhos Relacionados

- Padrões em OCL
 - [Ackermann, 2006] - Coleção de 18 padrões (OCL)
- Padrões em Transformações
 - [Bézivin et al., 2005] - Coleção de 23 padrões (ATL)
 - [Iacob et al., 2008] - Coleção de padrões (QVT)
- Padrões em Metamodelos
 - [Mili et al., 1998] - Coleção de 3 padrões (Smalltalk)
 - [García et al., 2009] - Diretriz para definição de metamodelos específicos
 - [Hu et al., 2001] - Padrões para metamodelos XML



Considerações Finais

- As diretrizes são simples, porém...
- Aplicabilidade
 - Versões finais dos metamodelos
 - Metamodelos construídos por especialistas
- Identificação semiautomática
 - Técnicas de mineração de dados
 - *Defining Association Member Ends Features*



Referências

- [OMG-2012] Object management group (OMG), <http://www.omg.org>.
- [Barbero-2006] M. Barbero, “Java 1.0 metamodel,” AtlanMod, 2006, http://www.emn.fr/z-info/atlanmod/index.php/Ecore#JavaAbstractSyntax_1.0
- [Robin et al.] J. Robin, D. Stoll, C. Atkison, “KobrA2 metamodel,” unpublished.
- [A. Vieira, 2010] A. Vieira, “Identificação de diretrizes para a construção de meta-modelos na infra-estrutura de MDA,” master dissertation, Federal University of Campina Grande - Paraíba, 2010,
http://docs.computacao.ufcg.edu.br/posgraduacao/dissertacoes/2010/Dissertacao_AndrezaDeSousaVieira.pdf
- [Ackermann, 2006] J. Ackermann, “Detailed description of OCL specification patterns,” Germany, University of Augsburg, 2006.
- [Bézivin et al., 2005] J. Bézivin, F. Jouault, J. Paliès, “Towards model transformation design patterns,” In: 1st European Workshop on Model Transformations, 2005.
- [Ambler,2005] S. Ambler, “The elements of UML 2.0 style,” Cambridge University Press, 2005, ISBN: 9780521616782.



Referências

- [Gamma et al., 1994] E. Gamma, R. Helm, R. Johnson, J. Vlissides, “Design Patterns: elements of reusable object-oriented software”, Addison-Wesley Professional, 1994.
- [Iacob et al., 2008] M.-E. Iacob, M. W. A. Steen, L. Heerink, “Reusable model transformation patterns,” In: Workshop on Models and Model-driven Methods for Enterprise Computing, 2008, pp. 1-10.
- [Mili et al., 1998] H. Mili, F. Pachet, “Patterns for metamodeling,” 1998, <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.56.2811>
- [García et al., 2009] I. García-Magariño, R. Fuentes-Fernández, J. Gómez-Sanz, “Guideline for the definition of EMF metamodels using an entity-relationship approach,” Information and Software Technology, Elsevier, 2009
- [Basili et al., 2001] V. R. Basili, G. Caldiera, H. D. Rombach, “The goal question metric approach,” Wiley-Interscience, Encyclopedia of Software Engineer, 2001, 2nd ed., vol. 1.
- [Hu et al., 2001] Z. Hu, G. Vollmar, “Towards XML metamodel patterns for XML data modeling,” In: 12th International Workshop on Database and Expert Systems Applications, 2001



SPLAB
SOFTWARE PRACTICES LABORATORY

Identifying Guidelines for Constructing Metamodels

Autores:

Andreza Vieira e Franklin Ramalho

andreza@copin.ufcg.edu.br

franklin@dsc.ufcg.edu.br

Universidade Federal de Campina Grande

