

Comparação de Modelos de Configuração

Elder Cirilo (PUC-Rio)

Leopoldo Teixeira (UFPE)

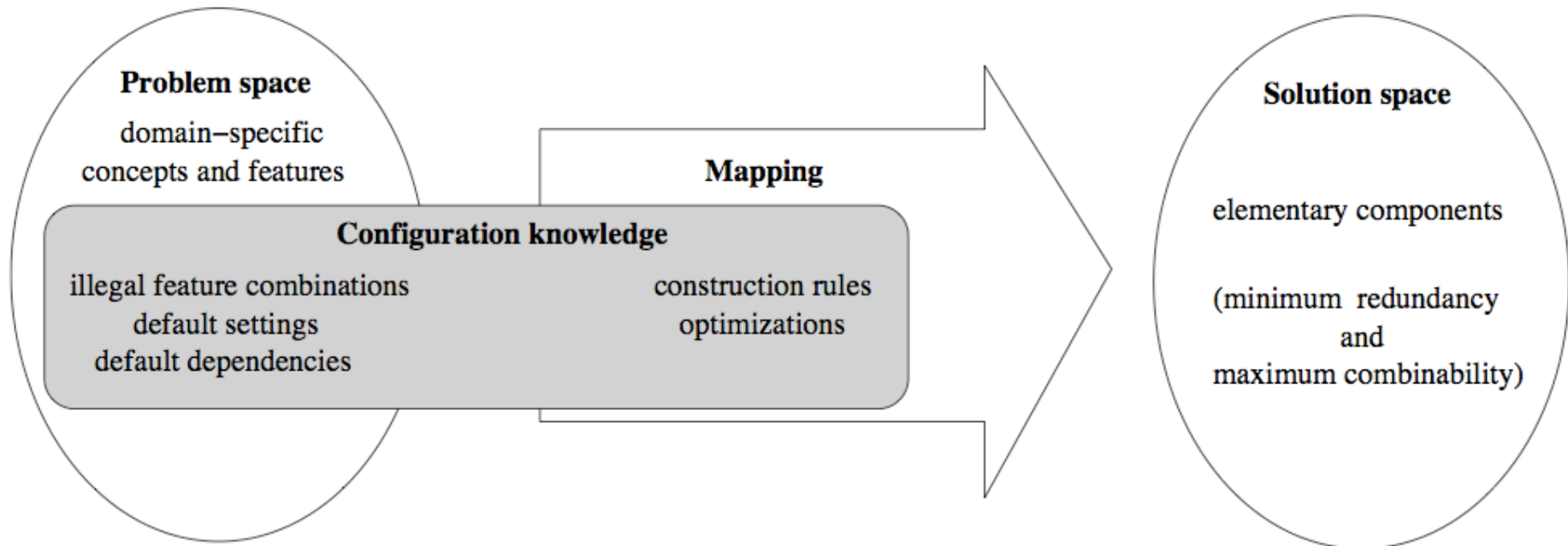
Rodrigo Bonifácio (UFPE)

Uirá Kulesza (UFRN)

Objetivo

- Comparar diferentes abordagens de representação de modelos de configuração
 - Ênfase em identificar vantagens e desvantagens das técnicas MSVCM e GenArch

Configuration Knowledge



Técnicas

- GenArch
- MSVCM
- Colored IDE (CIDE)
- pure::variants
- VML4*
- Model Template

Critérios

- Representação (expressividade)
- Tipos de artefatos de linha suportados
 - Como a técnica influencia o CK
- Rastreabilidade, entendimento e legibilidade
- Mecanismos de variabilidade
- Mecanismos de checagem estática
- Modularidade
 - modelos nem sempre são realmente independentes dos PLAs

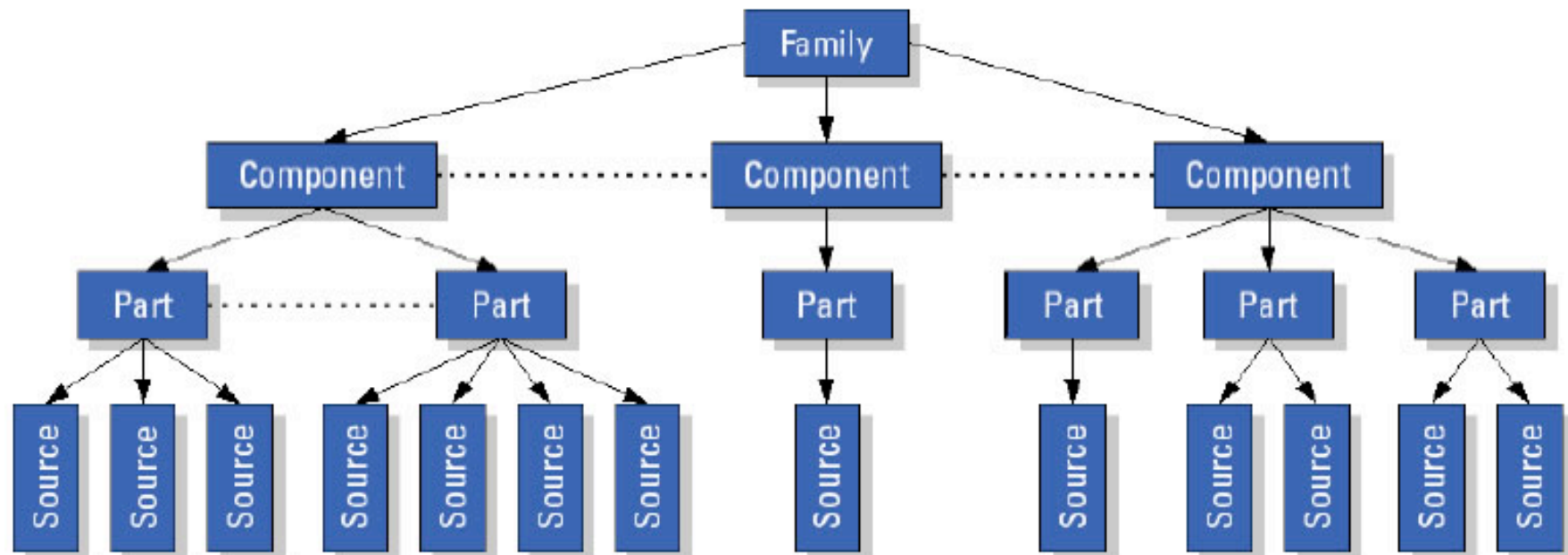
MSVCM

Feature Expression	Transformations
ReproduceMedia	selectScenario SC01
Streaming Content	bindParameter SC
Optional Controls	bindParameter OC
Favorite	evaluateAdvice AV01, selectScenario SC02
Sorting and Favorite	selectComponent SortingAndFavorite
Copy and Video	selectComponent CopyAndVideo
...	...

GenArch



pure::variants



Colored IDE (CIDE)

The screenshot displays the Colored IDE (CIDE) interface within the Eclipse SDK. The main editor shows the source code for `Stack.java` with several lines highlighted in different colors to represent features:

- Hidden Code:** A callout box points to the `loggingTarget` parameter in the constructor signature and its assignment in the constructor body.
- Feature Code:** A callout box points to the `push` method signature.
- Overlapping Features:** A callout box points to the `lock` variable declaration and its use in the `push` method body.

The ASTView on the right shows the abstract syntax tree for the selected code, with nodes like `SingleVariableDeclaration` and `Block` highlighted in red, corresponding to the feature annotations in the source code.

```
import java.io.PrintStream;

public class Stack {

    public Stack(int maxSize, PrintStream loggingTarget) {
        elementData = new Object[maxSize];
        logTarget = loggingTarget;
    }

    private int size = 0;

    private Object[] elementData;

    private PrintStream logTarget;

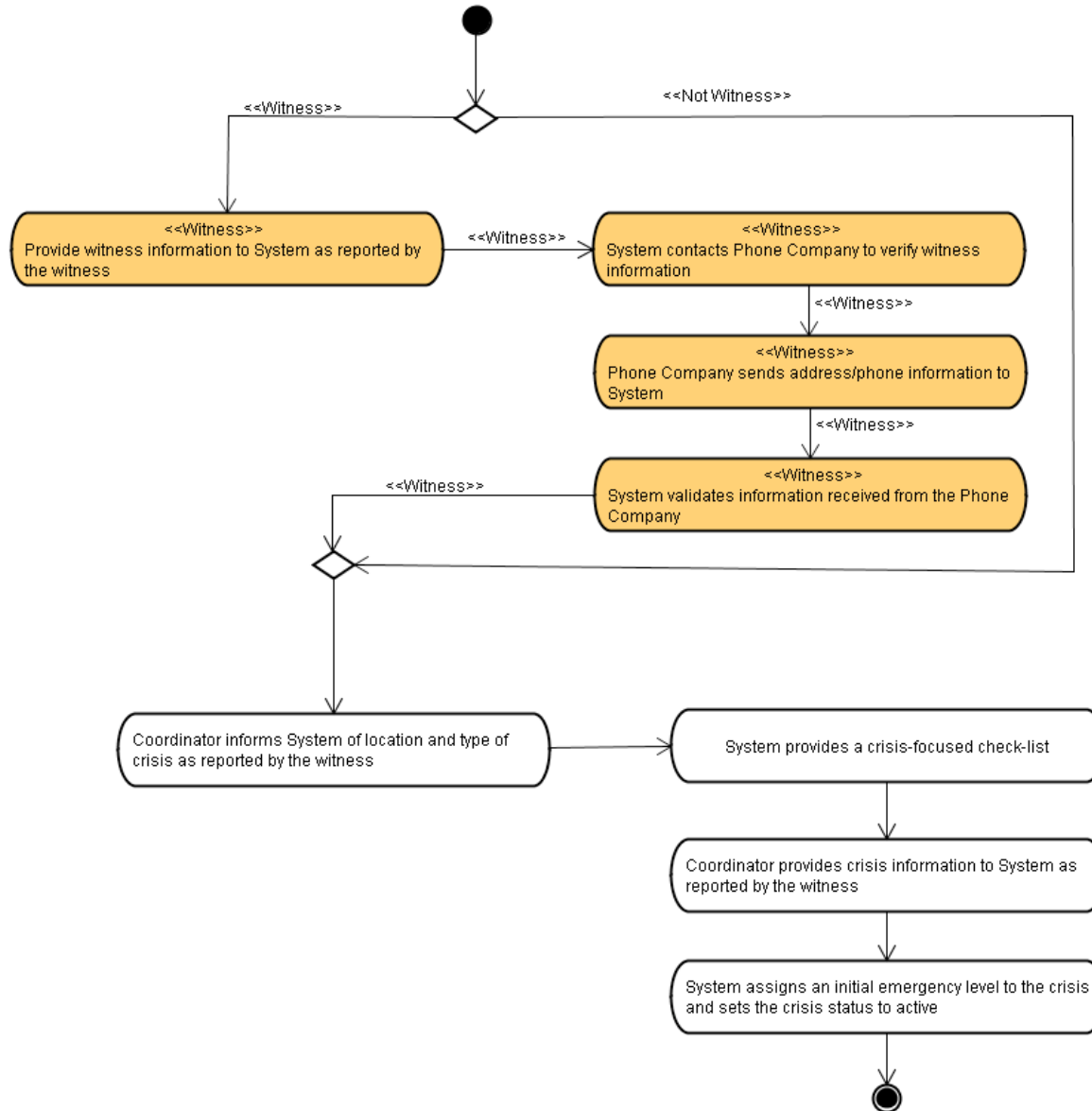
    public boolean push(Object o) {
        Lock lock = lock();
        if (lock == null) {
            log("lock failed for " + o);
        }
    }
}
```

Problems: PreviewView Feature Interactions

- All interactions (1)
- Interactions by Feature (2)
- All derivatives (3)

Stack.java - Stack 15M of 31M

Model Template



VML4*

```
variant Security {  
  
    insertPackage ("Security", "");  
    insertUseCase ("SecureTheHouse", "Security");  
    insertUseCase ("ActivateSecureMode", "Security");  
    createActorToUseCaseLink (  
        "Inhabitant", "Security::*");  
    createInclude (  
        "Security::SecureTheHouse",  
        or (  
            "Notification::SendSecurityNotification",  
            "WindowsManagement::"  
            "OpenAndCloseWindowsAutomatically"));  
}
```

Rastreabilidade

- Visualização é geralmente através do modelo
 - MSVCM
 - GenArch
 - pure::variants
 - VML4*
- Suporte ferramental fornece visões
 - CIDE: interações entre features, relatórios

Qual definição de modularidade
é suficiente/aplicável ao
contexto de modelos de
configuração?

Legibilidade, entendimento

- Modelo independente facilita o entendimento?
- Model Template e CIDE espalham o conhecimento pelos artefatos

annotative approach, but avoids the pollution of source code. Developers start with a fully composed application with all features implemented in a single code base, typically a legacy application.

Modularidade

- Artefato independente
 - MSVCM
 - GenArch
 - pure::variants
 - VML4*
- Conhecimento espalhado pelos artefatos
 - CIDE
 - Model Template

Mecanismo de Checagem

- Model Template
 - Avaliação de constraints OCL usando SAT
- CIDE
 - Garante boa-formação sintática para todos (parse)
 - Safe composition para Java e Bali (compile)
- pure::variants
 - Mecanismo de extensão
- MSVCM
 - Verificação de CK (Em desenvolvimento)

Questões em aberto

- (apenas) Lógica proposicional não parece ser suficiente

(Music and Photo) or (Music and Video) or (Photo and Video) or (Music and Photo and Video)	SelectClass(alternative/SelectMediaController.java) SelectClass(alternative/SelectTypeOfMedia.java) SelectAspect(alternative/TwoAlternativeFeatures.aj)
(Photo and not(Music) and not(Video)) or (Music and not(Video) and not(Photo)) or (Video and not(Music) and not(Photo))	SelectAspect(alternative/OneAlternativeFeature.aj)

- Usar quantificadores parece interessante em alguns casos

Questões

- Outras técnicas?
 - CrossMDA
 - Captor-AO
 - MetaEdit
 - openArchitectureWare
- Mais algum critério?

Proposta de tarefas

- Comparação de CK
 - VAMOS 2010
 - 10 de Novembro de 2009
- Extrair visualizações a partir do modelo
 - VisPLE 2010
 - Mid-2010??
- Métricas
 - Modelar o estudo de caso nas diferentes abordagens

Comparação de Modelos de Configuração

Elder Cirilo (PUC-Rio)

Leopoldo Teixeira (UFPE)

Rodrigo Bonifácio (UFPE)

Uirá Kulesza (UFRN)