

A Bottom Up SPL Design Method

—

Luciano Marchezan

Roteiro

- Objetivo
- Processo
 - Harmonização de nomes
 - Engenharia reversa
 - Extração de Feature Model

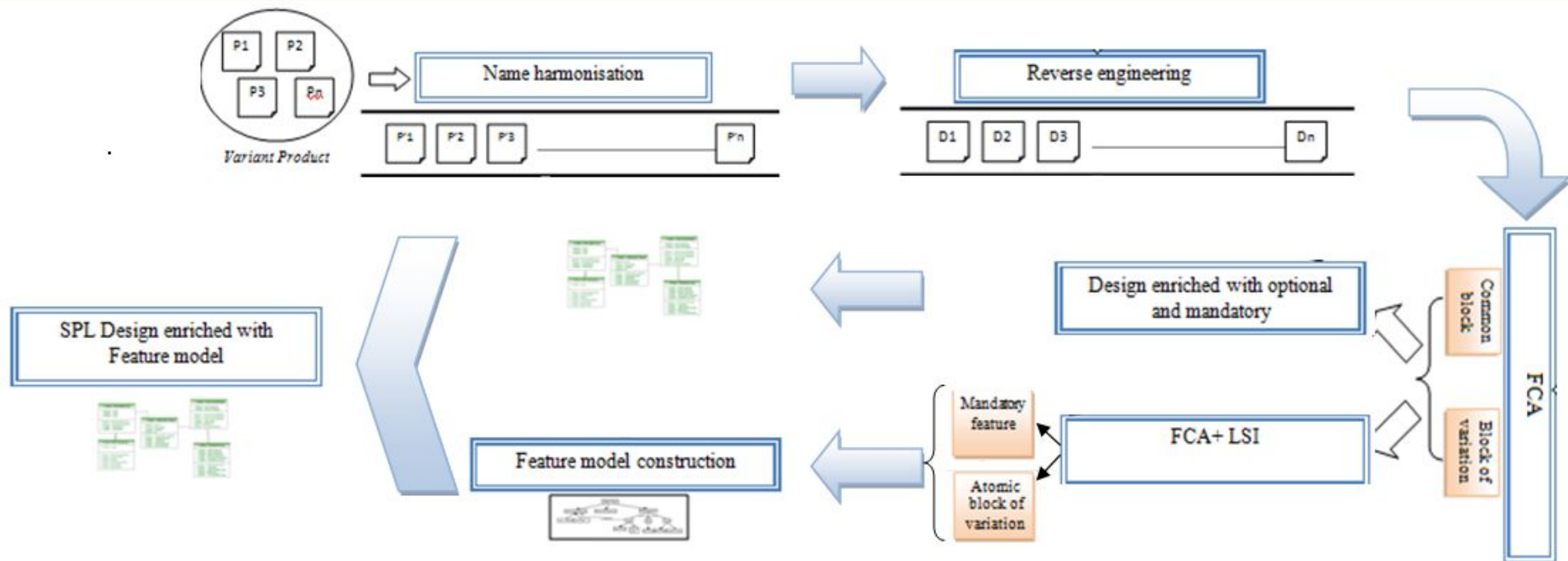
Objetivo

- Extrair do código fonte um *feature model* de um SPL.
- Modelar com um perfil UML que ajuda na compreensão, reuso e evolução do SPL.

O Processo

- Harmonização de nomes
- Engenharia reversa
- Extração de feature model
- Extração de elementos de projeto
- Construção do projeto SPL

0 Processo



Definições

- WordNet é um banco de dados para o idioma Inglês. Agrupa palavras em inglês em grupos de sinônimos chamados *synsets* e fornece definições curtas e exemplos de uso além de gravar relações entre esses sinônimos.
- Meronymy denota uma parte que faz parte ou é membro de algo. Ex: dedo é um meronym de mão.
- Um hyponym compartilha uma relação *type-of* com seu hypernym. Ex: águia e corvo são hyponyms de pássaro(hypernym).

Harmonização de nomes

- Identificar as correspondências semânticas entre nomes de pacotes, classes, métodos e atributos utilizando o WordNet.
- As relações são examinadas na seguinte ordem: equivalência (sinônimo), extensão (meronyms), e por fim a generalização (hypernyms).
- Caso exista uma ontologia de domínio, é preferível utilizá-la à um dicionário (Wordnet).
- Ao final dessa etapa, toda a semântica relacionada aos nomes deve estar harmonizada para ser posteriormente analisada.

Engenharia Reversa

- É aplicada a engenharia reversa do código fonte para se gerar diagramas de classe e sequência.

Extração de Feature Model

- É utilizada FCA (Formal Concept Analysis) e LSI (Latent Semantic Indexing) para se extrair os pontos em comum e as variantes.
- FCA é um método de análise de dados que busca analisar os dados descritos através de relações entre um conjunto particular de elementos. Neste caso, seriam os dados representando as variantes sendo analisadas. Esses dados são representados em uma tabela.
- As linhas da tabela são as variantes do produto e as colunas são representadas por métodos, classes, etc.

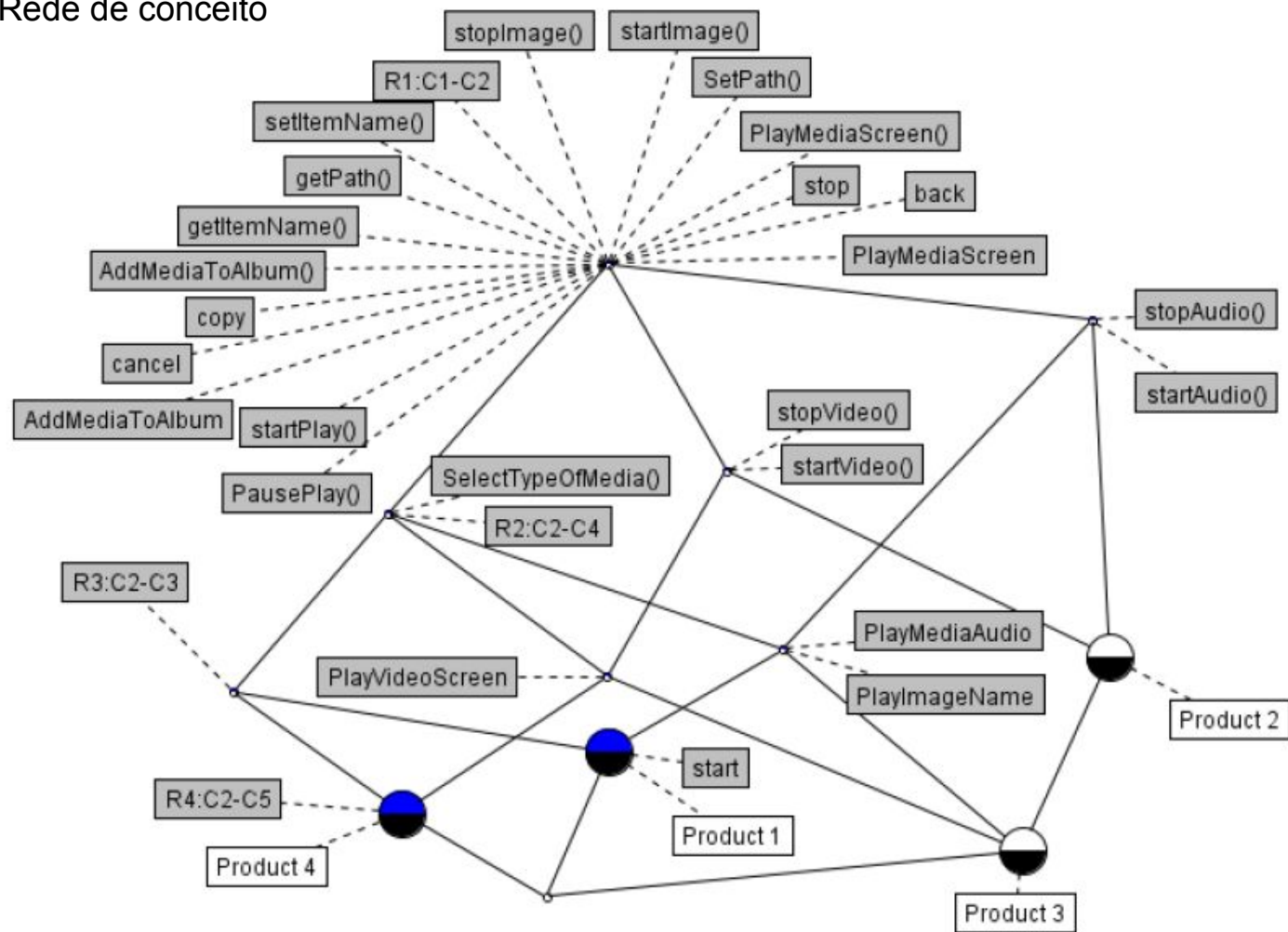
	PlayMediaScreen	Back	Start	Stop	PlayMediaScreen ()	PausePlay ()	startPlay ()	SelectTypeOfMedia ()	AddMediaToAlbum	Cancel	Copy	AddMediaToAlbum ()	getItemName ()	getPath ()	setItemName ()
<i>Product1</i>	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
<i>Product2</i>	X	X		X	X	X	X		X	X	X	X	X	X	X
<i>Product3</i>	X	X		X	X	X	X	X	X	X	X	X	X	X	X
<i>Product4</i>	X	X		X	X	X	X	X	X	X	X	X	X	X	X

Figura 1: tabela representado as variantes sendo analisadas

Extração de Feature Model

- Desta tabela é derivada uma rede de conceito que permite definir pontos em comum e variações entre todos os produtos.
- O elemento no topo da rede indica que certos objetos têm elementos em comum.
- O elemento na base indica que certos atributos se encaixam em objetos comuns(variações).
- Além disso, a rede ainda indica a relação entre elementos. Dessa relação serão derivadas as relações do *Feature Model*(mandatory, optional, Xor, etc).

Rede de conceito



Extração de Feature Model

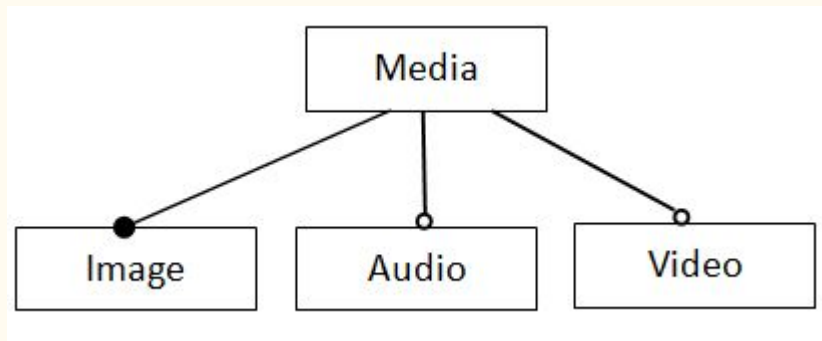
- LSI é utilizado para medir o grau de similaridade entre nomes de pacotes, classes, etc.
- De maneira informal, LSI assume que as palavras que sempre aparecem juntas são relacionadas.
- FCA e LSI são usadas em conjunto para se identificar *features* baseado na similaridade textual.
- Essa similaridade é descrita por uma matriz de similaridade.

Extração de Feature Model - Determinar a hierarquia e restrições entre as features

- As *features* (classes, pacotes, etc) são renomeadas baseado na frequência do nome dos elementos.
 - A organização e estrutura das *features* é recuperada baseada nos critérios de semântica.
 - Hypernoms permitem construir a Hierarquia do feature model.
 - Str_extension e Synonyms permitem Determinar restrições.
- *Hypernoms*(FeatureN1, FeatureN2) — — — — > *FeatureN1 is the parent of FeatureN2*
 - *Str_extention*(FeatureN1, FeatureN2) — — > *FeatureN1 OR FeatureN2*
 - *Synonyms*(FeatureN1, FeatureN2) — — — — > *FeatureN1 XOR FeatureN2*

Extração de Feature Model

- As restrições entre diferentes *features* que são extraídas utilizando FCA e LSI são verificadas e outras restrições são adicionadas baseado nos critérios da semântica.
- Por fim, é criado um *feature model* para representar as variações dos produtos.



Extração de elementos de projeto e Construção do projeto

A organização e estrutura do projeto é recuperada baseada nas seguintes regras de construção:

- R1: cada classe obrigatória será apresentada com seus elementos obrigatórios (atributos e métodos)
- R2: se uma relação é obrigatória, então o final da associação é obrigatório e será apresentado no projeto.
- R3: se uma relação tem uma *startAssociation* ou uma *endAssociation* obrigatória, ela será apresentada no projeto e as *startAssociation* ou *endAssociations* opcionais serão apresentadas com o estereótipo “recommended”.
- R4: os demais elementos serão apresentados de acordo com o grau de presença em todos os diagramas de classe.

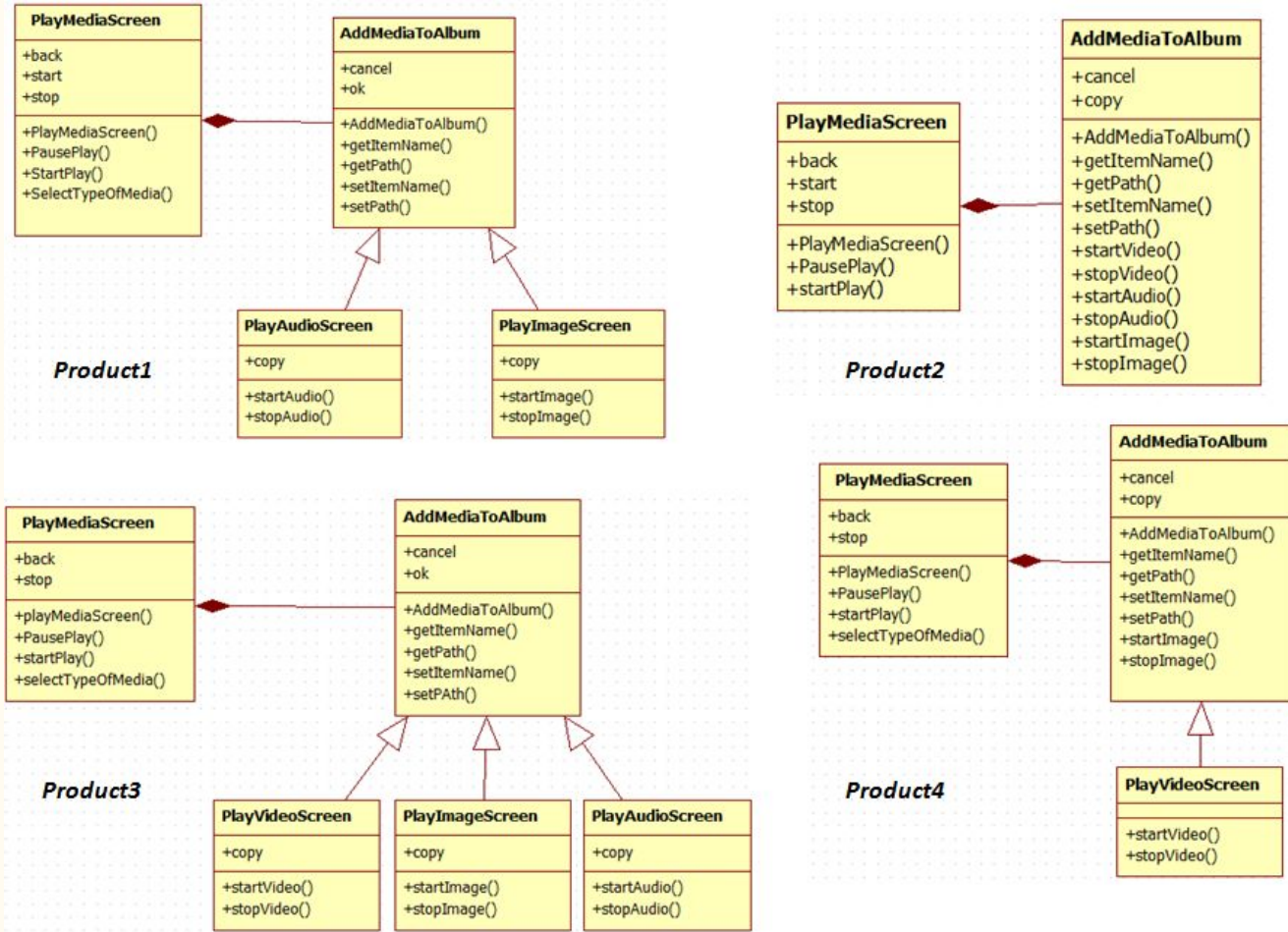


Diagrama de classes de 4 variantes de produto

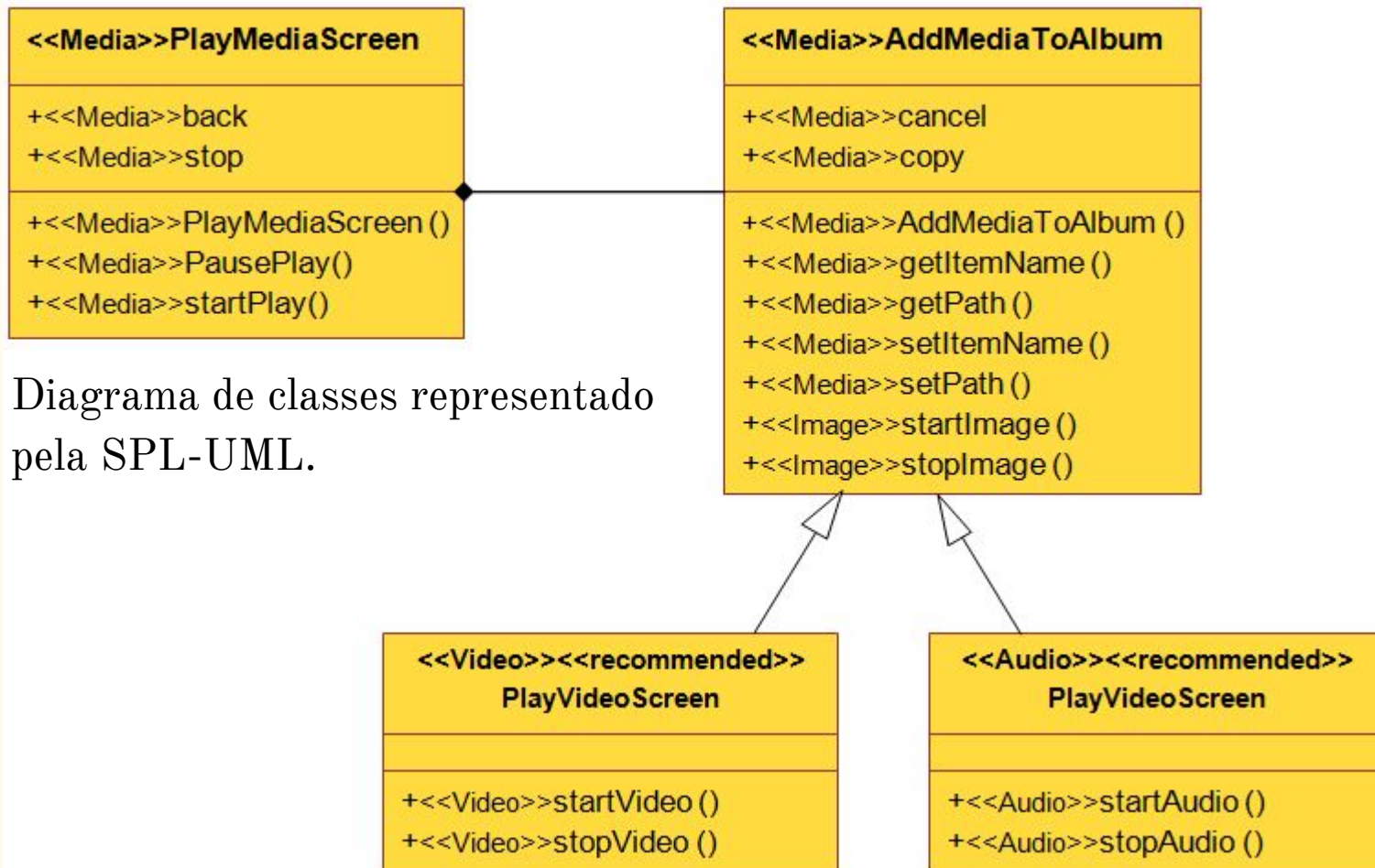


Diagrama de classes representado pela SPL-UML.

Referências

Maazoun, Jihen, Nadia Bouassida, and Hanêne Ben-Abdallah. "A bottom up SPL design method." Model-Driven Engineering and Software Development (MODELSWARD), 2014 2nd International Conference on. IEEE, 2014.

Ganter, Bernhard, and Robert Godin. Formal concept analysis. Springer Berlin/Heidelberg., 2005.