



PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DE PERNAMBUCO

Comparing GORE modeling languages using the Variability Metamodel of the Klaus Pohl's Orthogonal Variability Framework

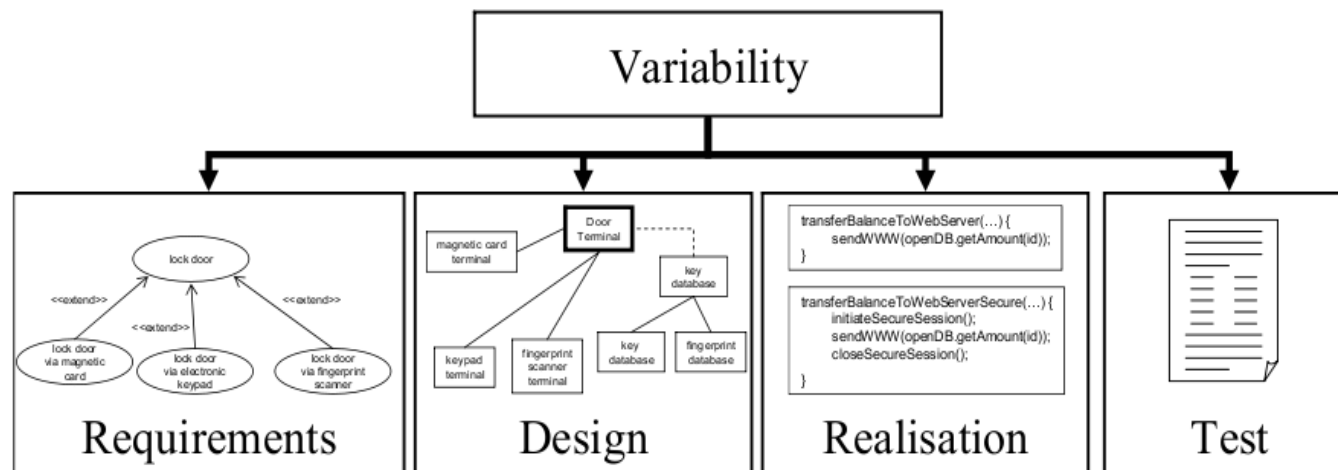
Carlos Diego Lima

Agenda

- Contextualização e Problema
- Objetivos
- Comparativo entre abordagens GORE e SPL
- Conclusões

Contextualização

- Modelos Ortogonais de Variabilidade
 - Alternativa para construir produtos específicos de software baseando-se em informações do domínio e mantendo a rastreabilidade entre a Variabilidade do Domínio e os Artefatos de Software



Contextualização

- Modelos Ortogonais de Variabilidade
 - Permitem explorar a informação de variabilidade do domínio sem espalhá-la por diversos modelos
 - Provê o rastreamento entre a informação de variabilidade do domínio e os artefatos de desenvolvimento de software
 - Protege os artefatos de desenvolvimento de software da sobrecarga (por exemplo, com o uso de anotações) de informação sobre variabilidade

Problema

Os Modelos Ortogonais de Variabilidade existentes carecem de informações sobre a justificativa da escolha de um conjunto de componentes que estarão presentes numa configuração específica de produto de software

Objetivos

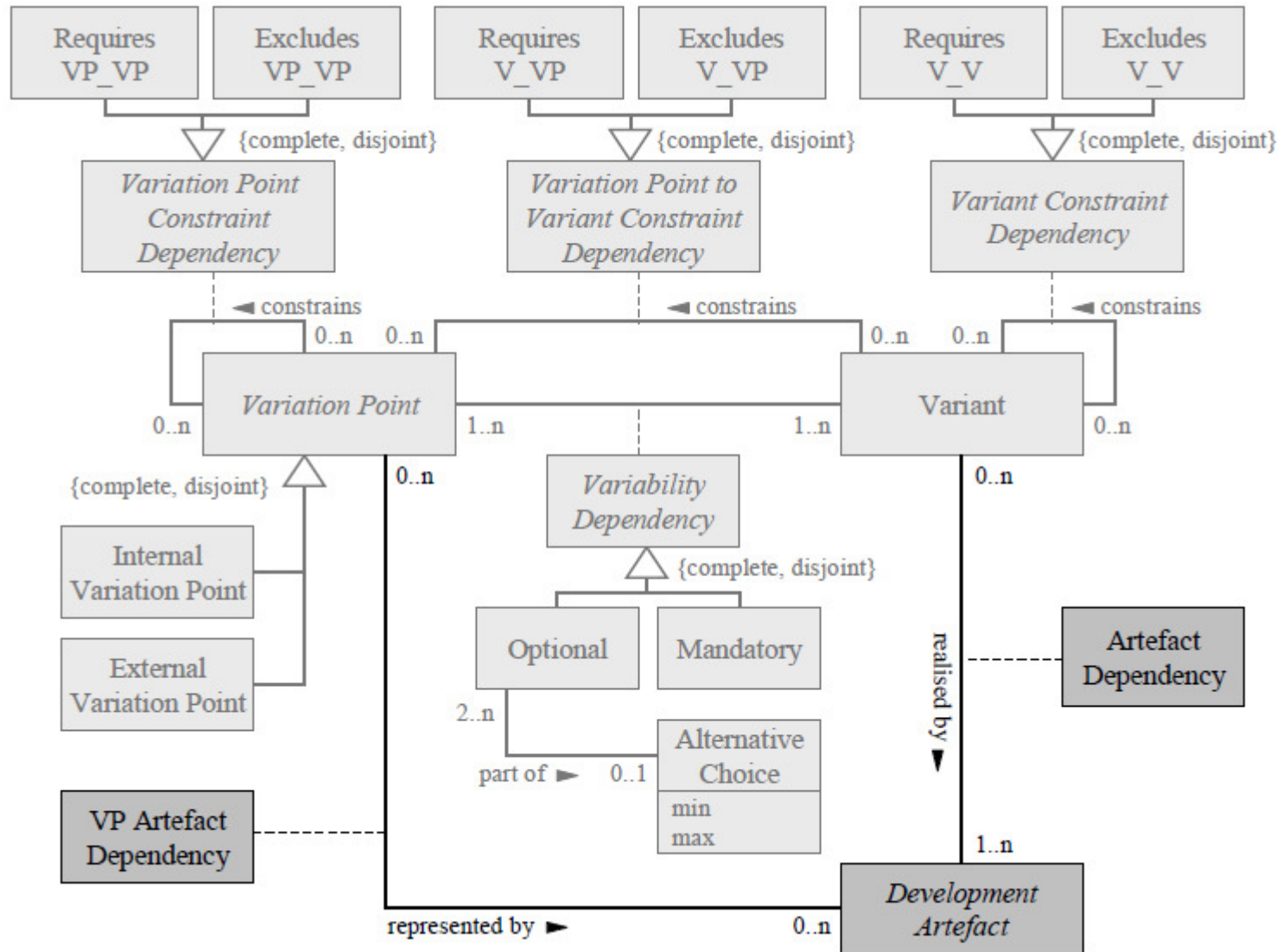
- Geral:
 - Propor uma linguagem GORE tão representativa quanto um OVM e, assim, capturar a informação de variabilidade de domínio e as intencionalidades dos stakeholders
 - Específicos:
 - Comparar abordagens GORE para SPL
 - Definir, a partir da comparação, uma nova Linguagem GORE com a expressividade de um OVM
-



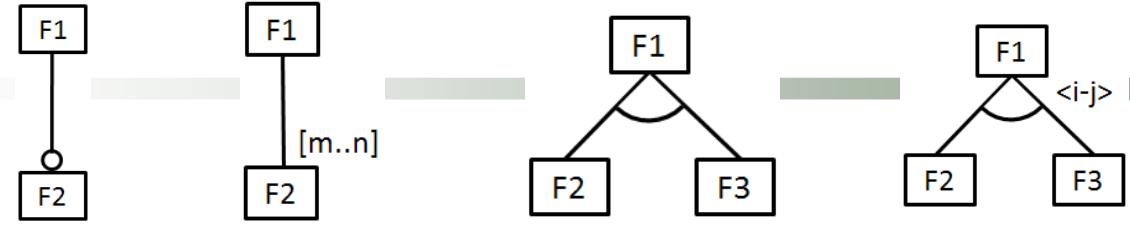
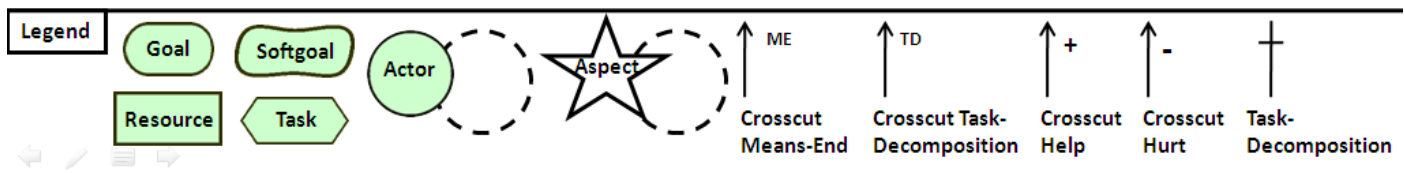
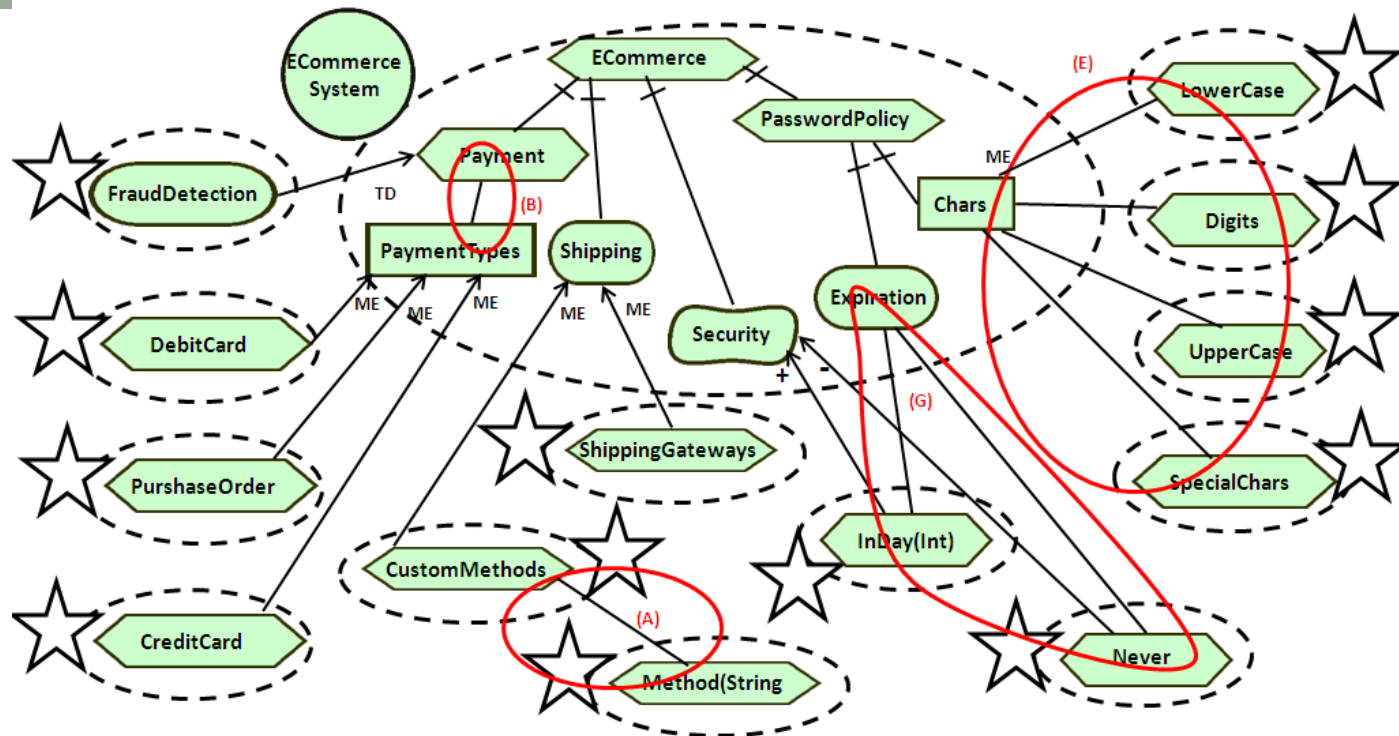
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DE PERNAMBUCO

Comparativo entre as linguagens GORE para SPL segundo o Metamodelo de Variabilidade

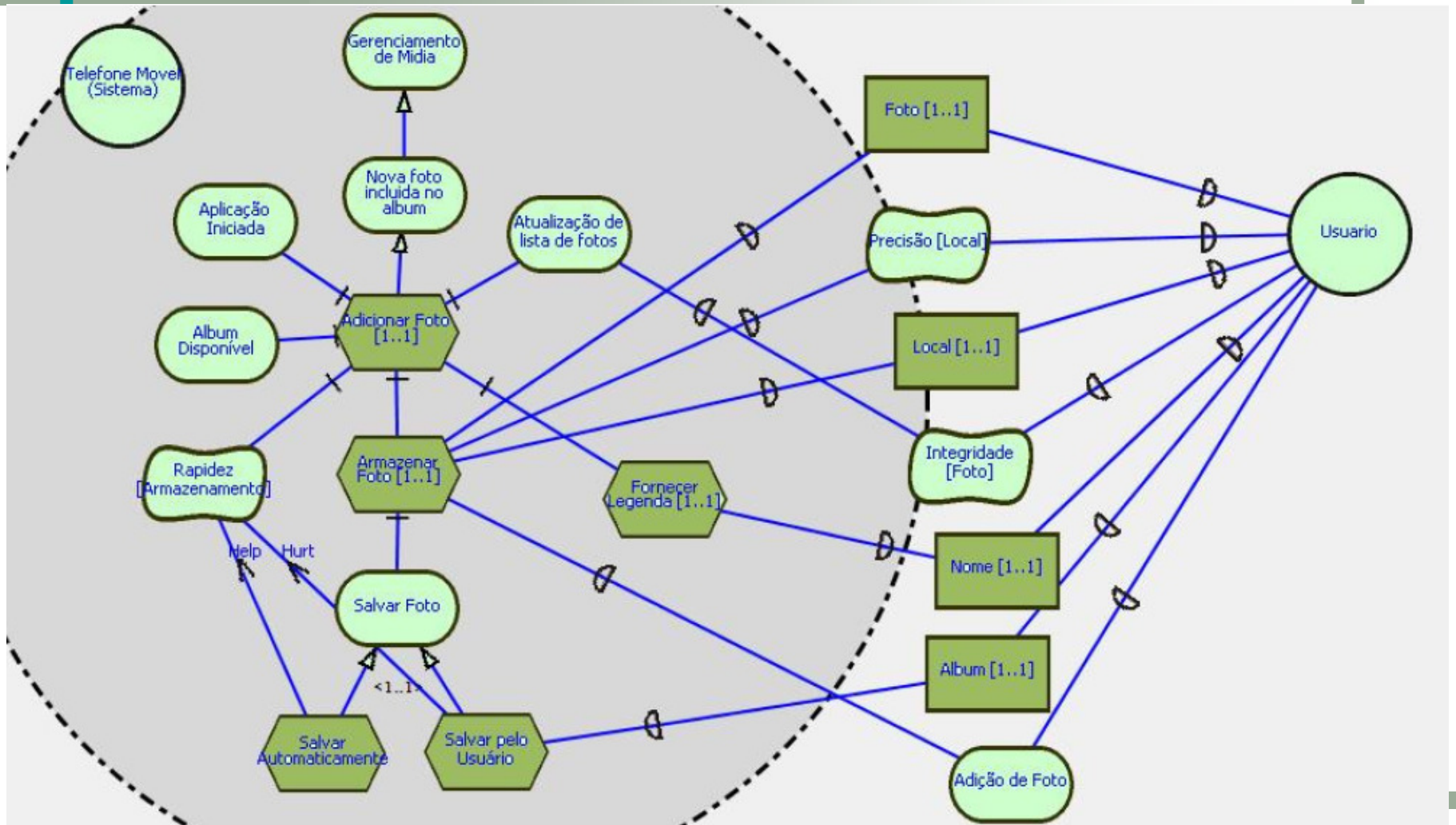
Metamodelo de Variabilidade



Aspectual i* for SPL

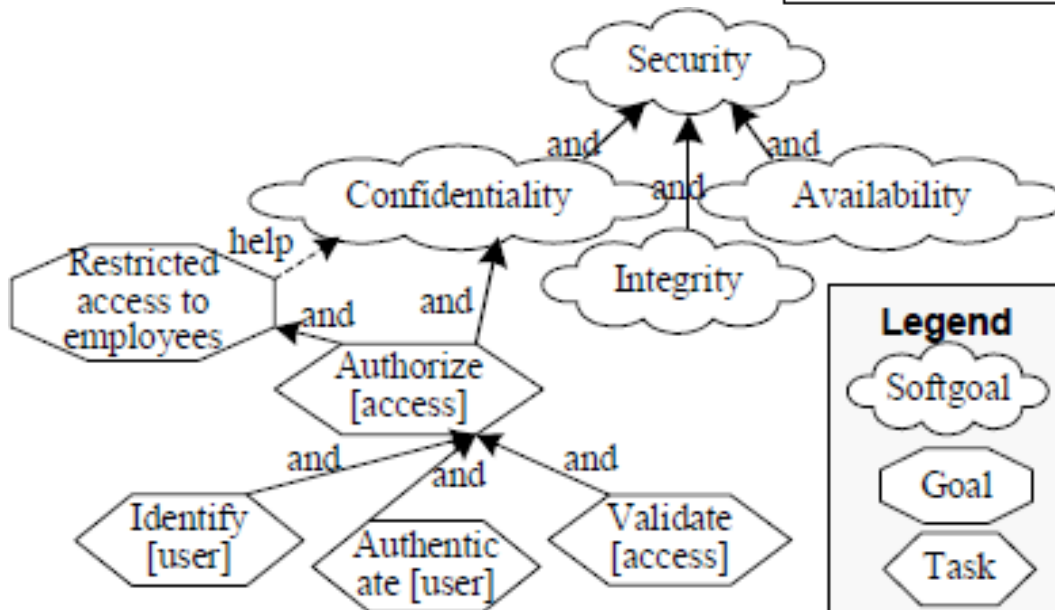


i*-C



PL-Aovgraph

Feature model		⇔	PL-AOVgraph	
Root of feature model			Name of goal model	
Hierarchy of features			Hierarchy of goals, softgoals, and tasks	
Feature			Task, goal or softgoal	
	Mandatory			Contribution = and Property [typeFeature=mandatory; cardinality=(1,n)]
	Optional			Contribution = or Property [typeFeature=optional; cardinality=(0,n)]
	Alternative xor/inc-or			Contribution = xor/inc-or Property [typeFeature=alternative; groupFeature = { }; cardinality=(1,1)/(1,n)]
Constraint = "implies the non-selection"			Correlation (hurt)	
Constraint = "implies"			Correlation (make)	
Feature reference			Task_ref, goal_ref, or softgoal_ref	



Conceitos contidos no Metamodelo de Variabilidade

- Sujeitos de Variabilidade (Pontos de Variação)
 - Objetos de Variabilidade (Variantes)
 - Dependências de Variabilidade
 - Restrições de Variabilidade
-

Comparativo entre linguagens GORE para SPL

Variability Classes	i^*-c	Aspectual i^*	PL-AOV graph
1. Variant	TA, REInf	GO; TA	GO, TA, SG
2. Variation Point	GO	GO; TA	GO, TA, SG
3. Variability Dependency	GO/TA with Card; ME with Card; TD	ME; TD; Crosscut Link	(And,Or,XOr, Inc-OR) CL; Card; Properties typeFeature/groupFeature
3.1. Optional	GO/TA with Card; ME with Card;	ME	[Or CL; Card=(0,n); Property typeFeature= optional]
3.2. Mandatory	GO/TA with Card; ME with Card; TD	TD; Crosscut Link	[And CL; Card=(1,n); Property typeFeature= mandatory]
3.3. Alternative	GO/TA with Card; ME with Card;	-	[(Inc-Or, Xor) CL; Card=(1,1), (1,n); Properties: typeFeature= alternative, groupfeature]
4. Constraint	-	-	-
4.1. Requires	-	-	Make CR
4.2. Excludes	-	-	Hurt CR

SG: Softgoal / GO: Goal / TA: Task / REInf: Informational Resource / REPhy: Physical Resource / TD: Task-Decomposition / ME: Means-end / CL: Contribution-Link / CR: Correlation / Card: Cardinality / Inc-Or: Inclusive Or / XOr: Exclusive Or

Conclusões

- Contribuições:
 - A comparação demonstrou as capacidades das linguagens GORE em relação aos conceitos presentes no OVF por Klaus Pohl.
 - A técnica PL-Aovgraph é a mais completa, segundo a comparação realizada usando o metamodelo de variabilidade.
 - O uso de links de correlação como restrições é algo que precisamos discutir.
 - Usar GORE ajuda a justificar a escolha de um conjunto de features que estarão presentes no produto de software em SPL (através de softgoals).

Trabalhos Futuros

- Construir o i^* Ortogonal, uma linguagem GORE que complemente o i^* -c e sirva como Modelo de Variabilidade Ortogonal
- Integrar i^* Ortogonal com um processo de desenvolvimento que utilize linguagem GORE (ex: Tropos)
- Fazer o rastreamento dos elementos presentes i^* Ortogonal com os artefatos de desenvolvimento de software.