Optimizing Aspect-Oriented Product Line Architectures with Search-based Algorithms

THAINÁ MARIANI[1], SILVIA REGINA VERGILIO[1] AND THELMA ELITA COLANZI[2]

[1] FEDERAL UNIVERSITY OF PARANA (UFPR) - BRAZIL

[2] STATE UNIVERSITY OF MARINGA (UEM) - BRAZIL

SSBSE 2015, BERGAMO - ITALY

Introduction

A Software Product Line (SPL) defines a set of software products that share features to satisfy a domain;

Features are visible functionalities for the user;

A feature can be a variable functionality (variability) that may or not be present in a product;

A feature can be a mandatory functionality, being present in every product;

The Product Line Architecture (PLA) is used to derive the architecture of each product of a SPL and contains all the mandatory and variable features;

The use of Aspect-Oriented Modeling (AOM) concepts in the PLA design is useful to modularize crosscutting features;

Aspect-Oriented Product Line Architecture (AOPLA) contributes to improve modularity, stability and to reduce feature tangling and scattering.

Introduction

The PLA design needs to consider different principles, such as cohesion, coupling and feature modularization.

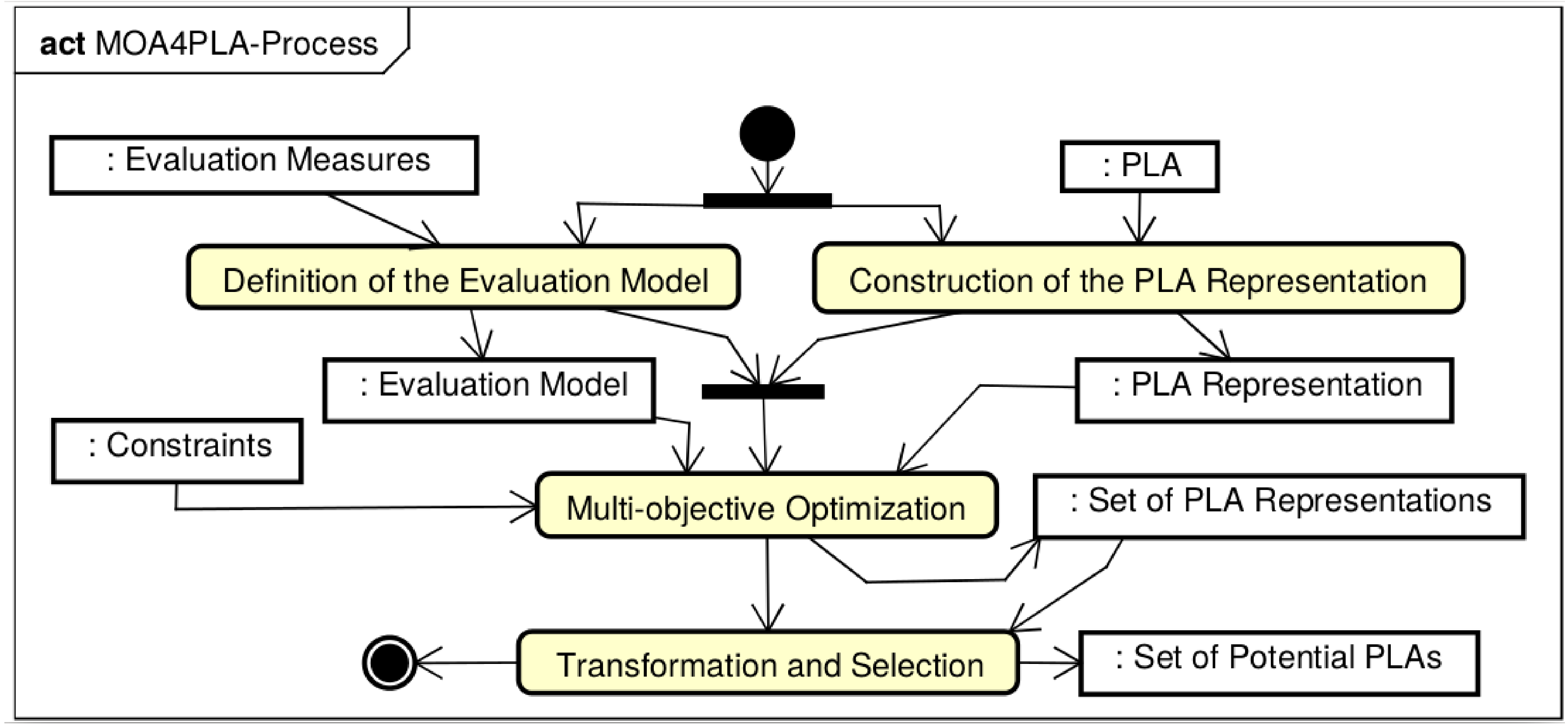
To help in the PLA design, Colanzi et al. introduced MOA4PLA (Multi-objective Optimization Approach for PLA Design), a search-based design approach for multi-objective optimization of PLAs.

MOA4PLA has some limitations when optimizing AOPLAs: i) the meta-model does not capture AOM particularities; and ii) the search operators may violate AOM rules;

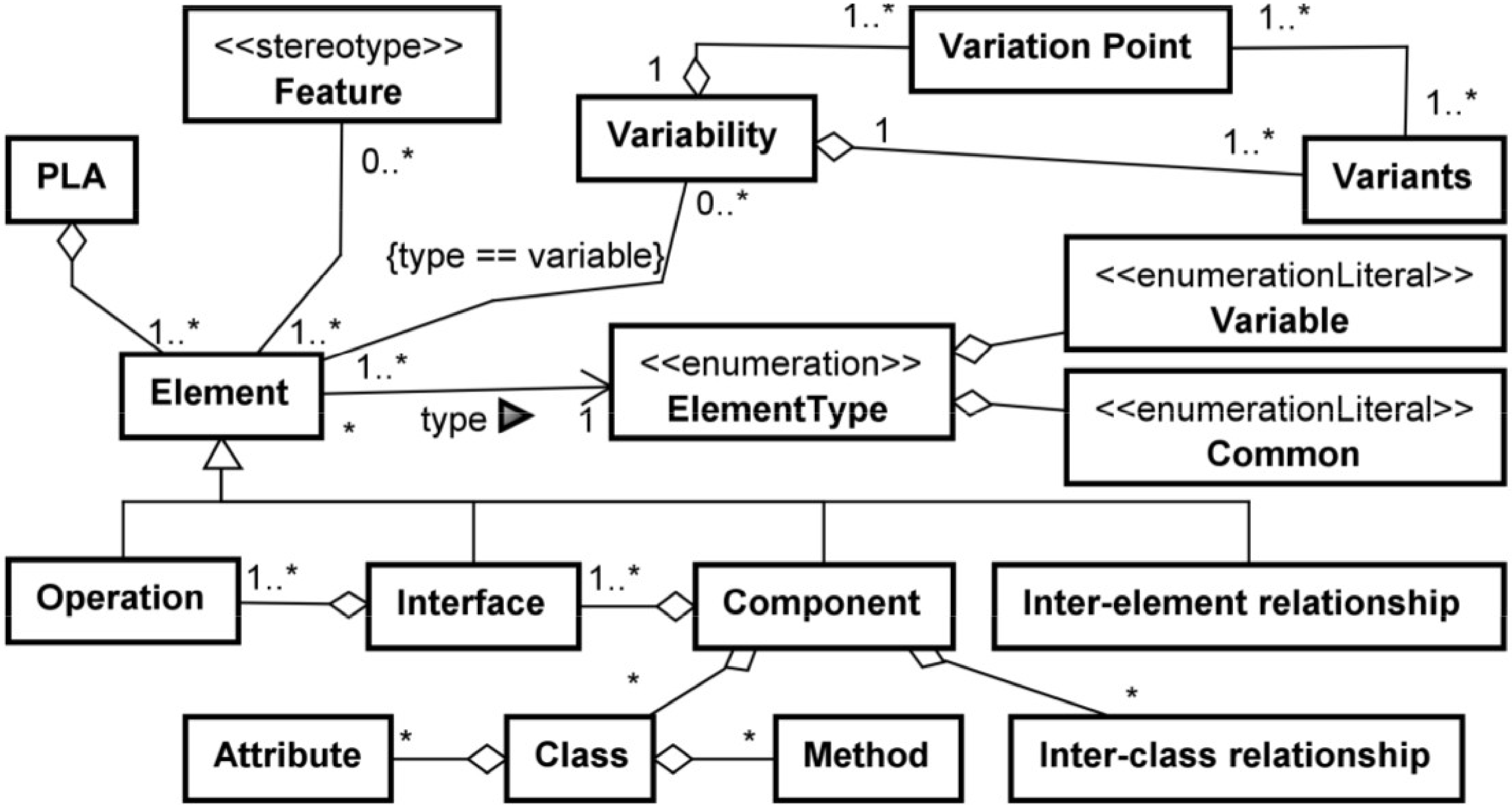
These violations reduce the AOM benefits and impact negatively in the architecture understanding;

To reduce such limitations and ensure the AOM benefits, this paper proposes a way to represent AOPLAs and a set of search-based operators that aggregate AOM rules.

MOA4PLA Activities



MOA4PLA Representation

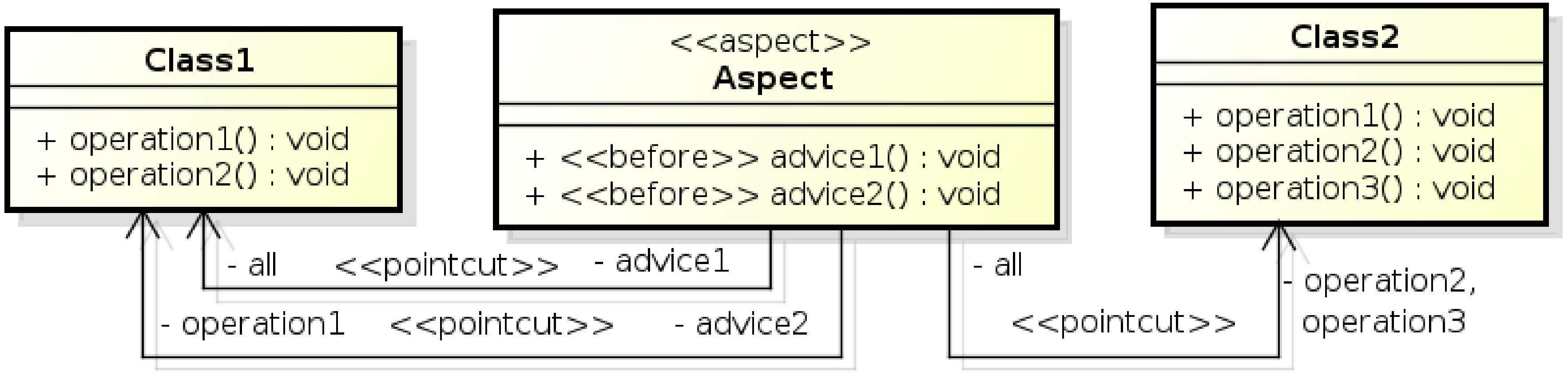


Representation of AOPLAs

Representation used to allow the optimization of AOPLAs by MOA4PLA;

Notation of Pawlak et al. was chosen to represent AOM elements in UML class diagrams;

AOM Elements: aspects, advices, join points, pointcuts and crosscutting relationships.



Search Operators for AOPLAs

Violations in an AOPLA can occur if an aspect is disconnected from its advices and crosscutting relationships;

To avoid such violations a set of search operators named Search Operators for Aspect-Oriented Architectures (SO4ASPAR) is proposed;

SO4ASPAR is derived from the MOA4PLA search operators by aggregating AOM rules.

Search Operators for AOPLAs

Move_Method4ASPAR (Move Method for Aspect-oriented Architectures): moves a method from a class to another;

Move_Attribute4ASPAR (Move Attribute for Aspect-oriented Architectures): moves an attribute from a class to another;

Add_Class4ASPAR (Add Class for Aspect-oriented Architectures): creates a class and moves to it a method or an attribute selected from a random class;

Move_Operation4ASPAR (Move Operation for Aspect-oriented Architectures): moves an operation from an interface to another;

Add_Package4ASPAR (Add Package for Aspect-oriented Architectures): creates a package, creates an interface inside this package and moves an operation from a random interface to the created one;

F**eature_Driven4ASPAR (Feature Driven for Aspect-oriented Architectures):** selects a crosscutting feature to be modularized and, to do this, all elements associated with the selected crosscutting feature are moved to a modularization package.

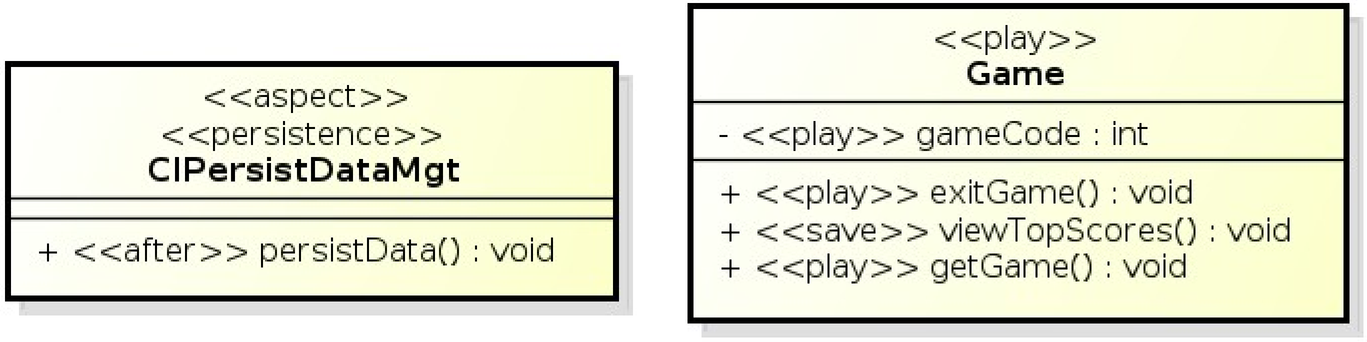
Aspect Rules

Advices, methods or attributes should not be moved from an aspect. If one of these elements is moved to other elements, a modularized concern can become scattered in the architecture and consequently, the aspect loses its benefits regarding modularization and cohesion;

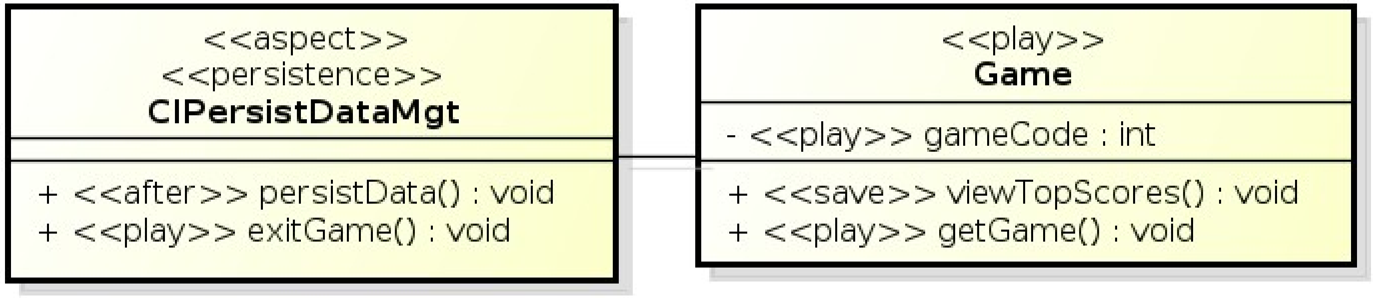
Methods and attributes should not be moved to aspects, because if they were, an aspect could be associated with additional concerns besides its main concern, losing its main modularization functionality.

Aspect Rules

a) Before



b) After

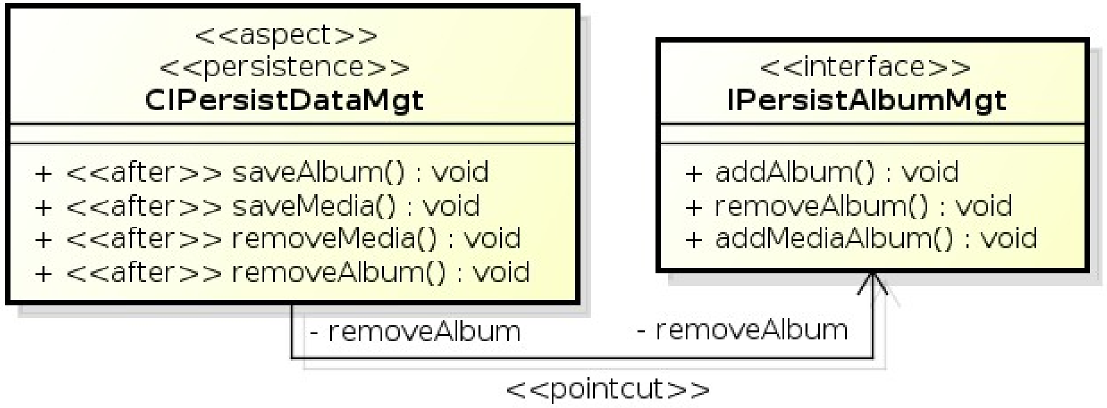


Join Point Rule

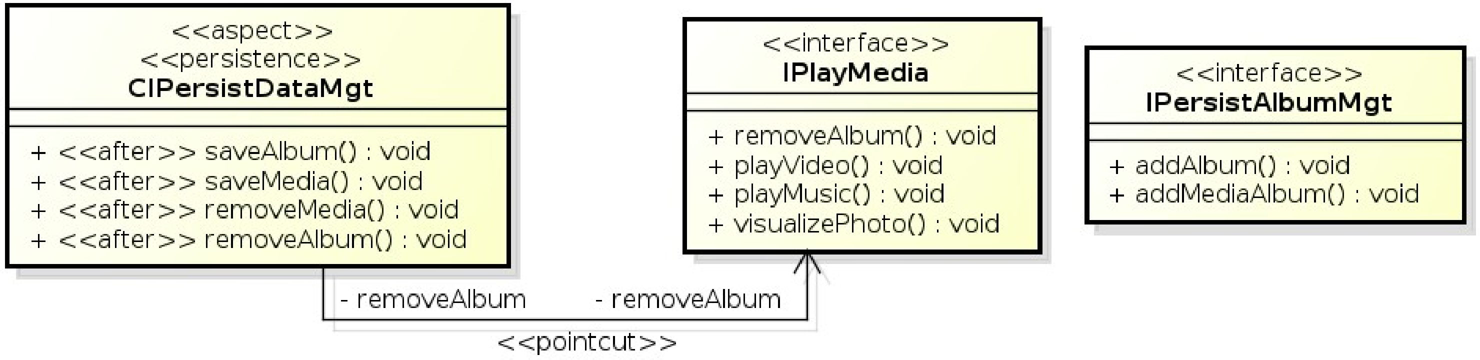
When a joint point is moved to an element, a crosscutting relationship must be added/adapted between the aspect and the target element, and removed/adapted between the aspect and the source element.

Join Point Rule

a) Before



b) After



Empirical Study Description

The experiments were conducted with MOA4PLA;

SO (search operators) and SO4ASPAR experiments;

The search operators were implemented as mutation operators in NSGA-II;

The fitness functions used are:

- CM (Conventional Metrics) to evaluate coupling and cohesion;
- FM (Feature Metrics) to evaluate feature modularization;

30 runs of each experiment were performed.

Empirical Study Description

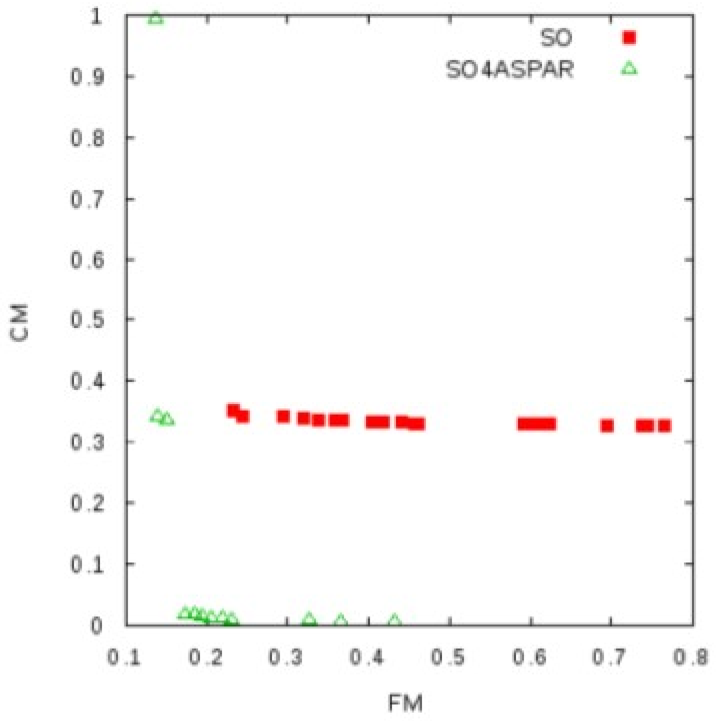
AOPLAs used: Arcade Game Maker (AGM), Mobile Media (MM), and Electronic Tickets in Urban Transportation (BET).

PLA	Fitness (FM, CM)	Packages	Interfaces	Classes	Features	Aspects	Pointcuts
AGM	(727, 6.1)	11	18	30	12	2	7
MM	(1074, 4.1)	10	18	14	14	2	8
BET	(1411, 90.0)	56	30	109	16	6	7

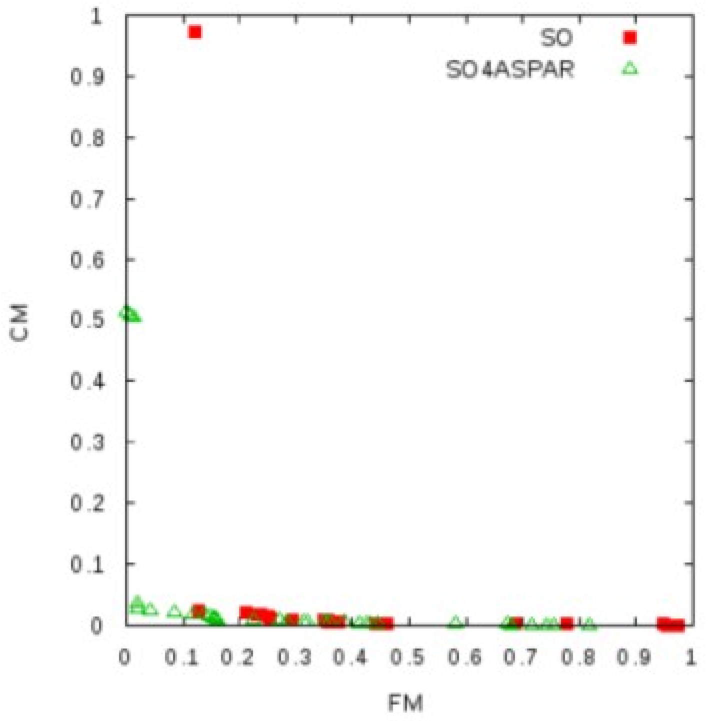
Pareto Front Results

PLA	PF_{true}	PF_{known}	
		SO	SO4ASPAR
AGM	12	19 (0)	12 (12)
MM	32	24 (0)	32 (32)
BET	13	13 (0)	13 (13)

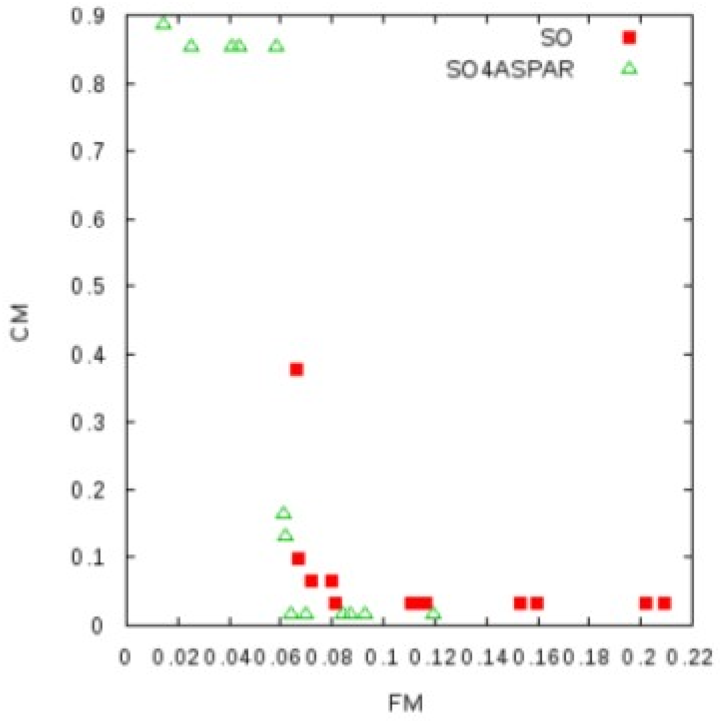
Pareto Front Results



(a) AGM



(b) MM

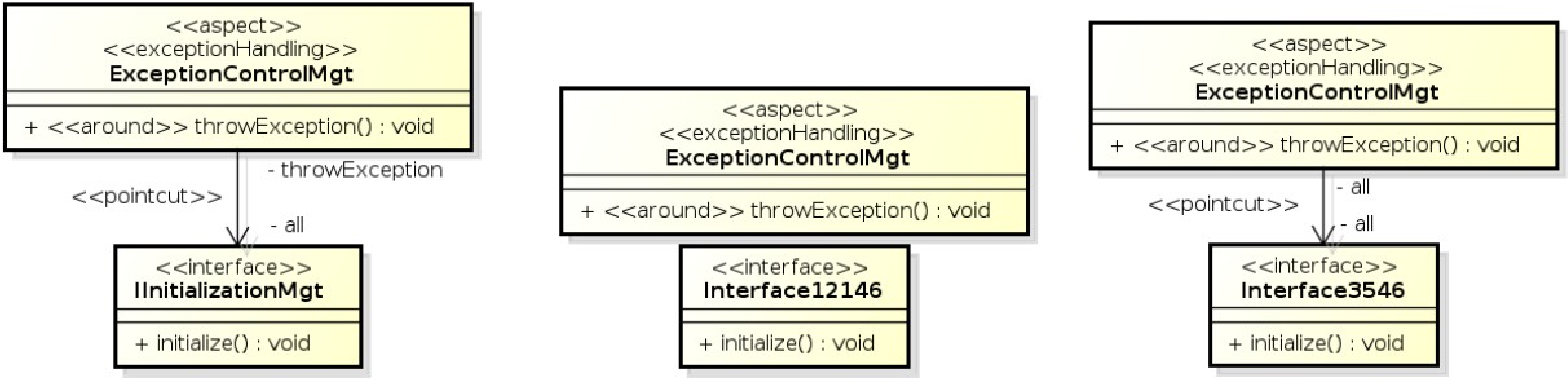


(c) BET

Hypervolume Results

PLA	**Experiment**	**Average (Std.Dev.)**	**Kruskal (p-value)**
AGM	SO	0.4734 (0.0157)	TRUE (4.28E-11)
	SO4ASPAR	**0.6126** (0.1191)	
MM	SO	0.7928 (0.0409)	TRUE (3.17E-11)
	SO4ASPAR	**0.9262** (0.0293)	
BET	SO	0.8668 (0.0375)	FALSE (0.06)
	SO4ASPAR	**0.8905** (0.0367)	

Qualitative Results



(a) Original (b) SO (c) SO4ASPAR

Concluding Remarks

This paper contributes to search-based design of AOPLAs by introducing a representation for AOPLA and search operators named SO4ASPAR;

The proposed representation takes into account the input AOPLA given by a class diagram, and allows representations of the AOM elements, such as aspects and crosscutting relationships;

Pareto results showed an improvement in the fitness values of the solutions generated by SO4ASPAR. Regarding the hypervolume quality indicator, SO4ASPAR presented better or statistically equivalent results to SO;

A qualitative analysis showed that the AOM rules were preserved in the SO4ASPAR experiments and, in most cases, violated in the SO experiments.

Future Works

Creation of new operators to aggregate rules of other modeling approaches;

Addition of AOM metrics in the fitness function to evaluate more accurately the AOM benefits;

Experiments should be conducted with other AOPLAs.

Thank you

AUTHOR EMAIL: MARIANITHAINA@GMAIL.COM