The Class Responsibility Assignment Case

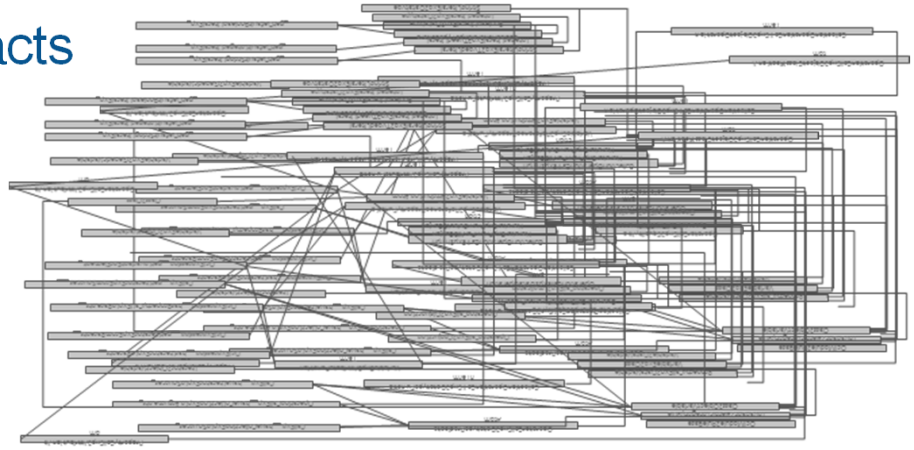
Martin Fleck, Javier Troya, and Manuel Wimmer

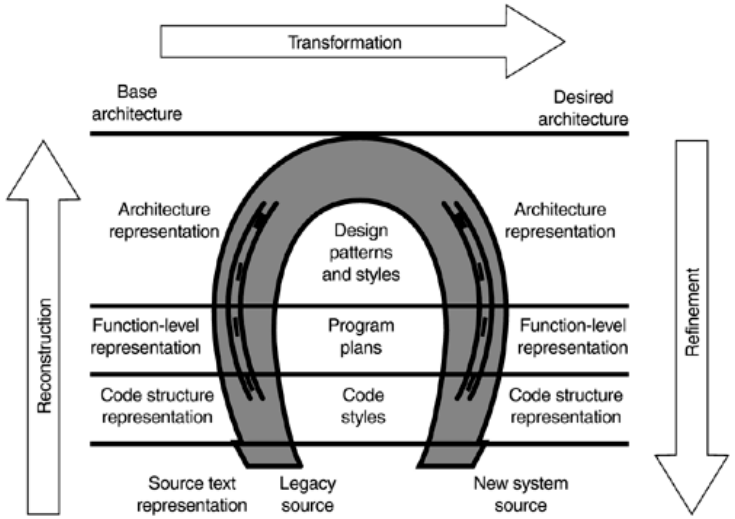
TTC, Vienna
8th July, 2016

Motivation

ARTIST Project: http://www.artist-project.eu

- Models are **human-oriented** artefacts
  - Maintainability
  - Testability
  - Readability
  - Understandability
  - …
- How to **improve** the **quality**?
- What is a **good solution**?





R. Kazman, S.G. Woods, S.J. Carrière: Requirements for integrating software architecture and reengineering models: CORUM II. In: Proc. WCRE, pp. 154-163, 1998.
Object Management Group: Architecture Driven Modernization, http://adm.omg.org/

Motivation

Class Responsibility Assignment (CRA) Problem

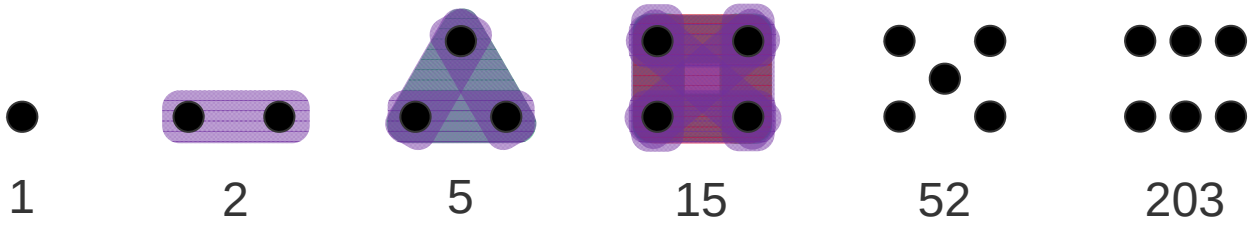
- **CRA** deals with the creation of **high-quality object-oriented models**
- For **solving** a particular CRA problem, one needs to decide **where responsibilities**, i.e., class operations and attributes, **belong**
- **When** do we have to deal with CRA problems?
  - **Generating class diagrams**: When migrating an application from a procedural language to an object-oriented language
  - **Optimizing class diagrams**: During the refactoring of an existing object-oriented model
- CRA is a **computationally challenging problem**
  - Huge search space!
  - Considered as an optimization problem

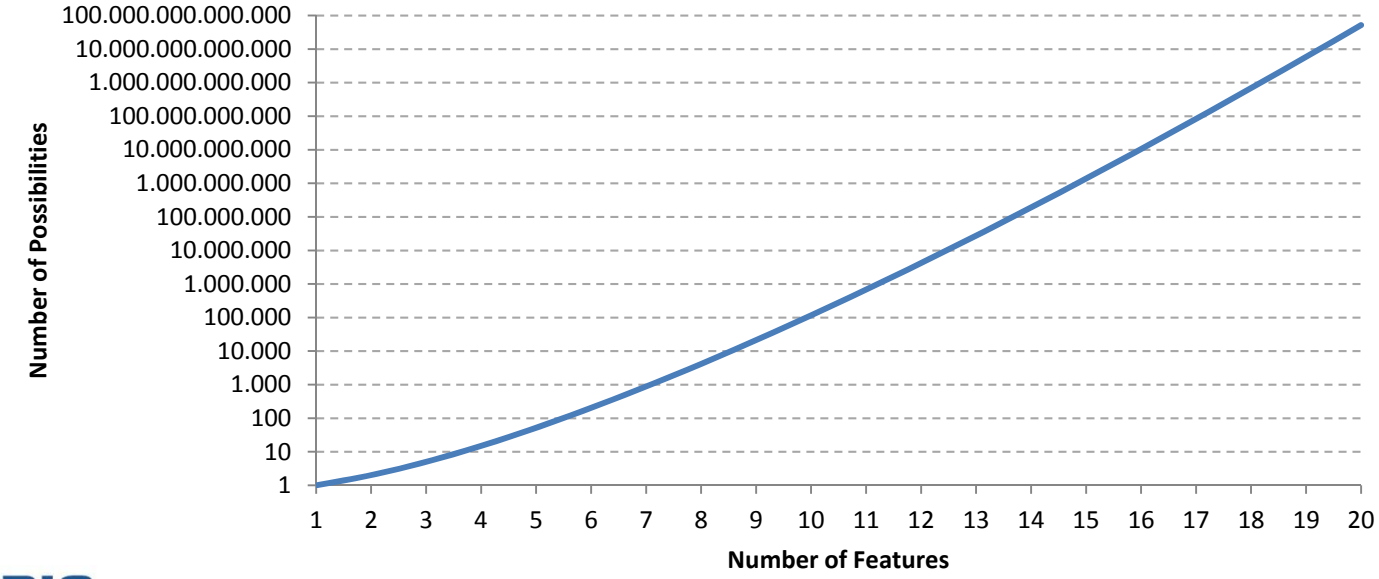
M. Bowman, L.C. Briand, and Y. Labiche. Solving the Class Responsibility Assignment Problem in Object-Oriented Analysis with Multi-Objective Genetic Algorithms. IEEE Transactions on Software Engineering, 36(6):817–837, 2010.

Motivation

Solving Complex Optimization Problems

- Class Responsibility Assignment Problem
  - Modularization of features into classes -> partitioning problem



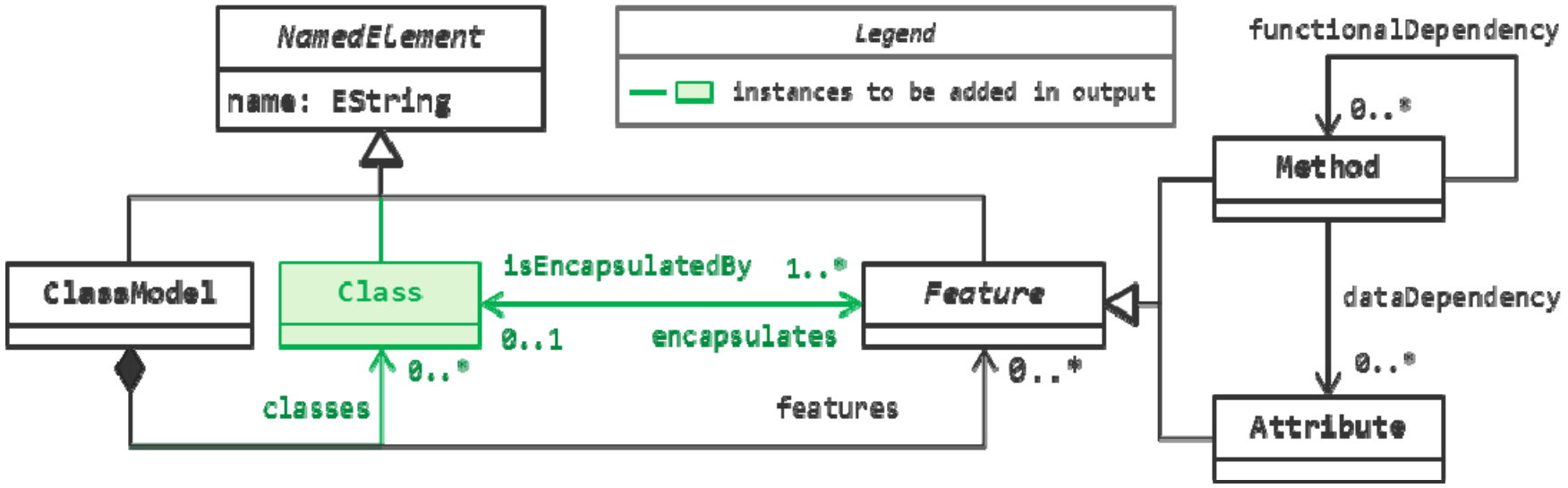


$$B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k$$

$$B_0 = 1$$

Modeling the CRA Problem

Input/Output Structures: RDG -> CD



- **Responsibility Dependency Graph (RDG) Language** (without green-colored elements)
  - One *ClassModel* contains all *Features*
  - *Features* have dependencies: functional and data
- **Class Diagram (CD) Language** (black-colored and green-colored elements)
  - *ClassModel* contains *Classes*
  - *Classes* encapsulate *Features*
  - No empty *Classes*/no unassigned *Features*

Modeling the CRA Problem

Fitness Function

$$CRA\text{-}Index = CohesionRation - CouplingRatio$$

$$CohesionRatio = \sum_{c_i \in Classes} \frac{MAI(c_i, c_i)}{|M(c_i)| \times |A(c_i)|} + \frac{MMI(c_i, c_i)}{|M(c_i)| \times |M(c_i) - 1|}$$

$$CouplingRatio = \sum_{\substack{c_i, c_j \in Classes \\ c_i \neq c_j}} \frac{MAI(c_i, c_j)}{|M(c_i)| \times |A(c_j)|} + \frac{MMI(c_i, c_j)}{|M(c_i)| \times |M(c_j) - 1|}$$

$$MMI(c_i, c_j) = \sum_{\substack{m_i \in M(c_i) \\ m_j \in M(c_j)}} DMM(m_i, m_j)$$

$$MAI(c_i, c_j) = \sum_{\substack{m_i \in M(c_i) \\ a_j \in A(c_j)}} DMA(m_i, a_j)$$

$$DMA(m_i, a_j) = \begin{cases} 1 & \text{if there is a dependency between method } m_i \text{ and attribute } a_j \\ 0 & \text{otherwise} \end{cases}$$

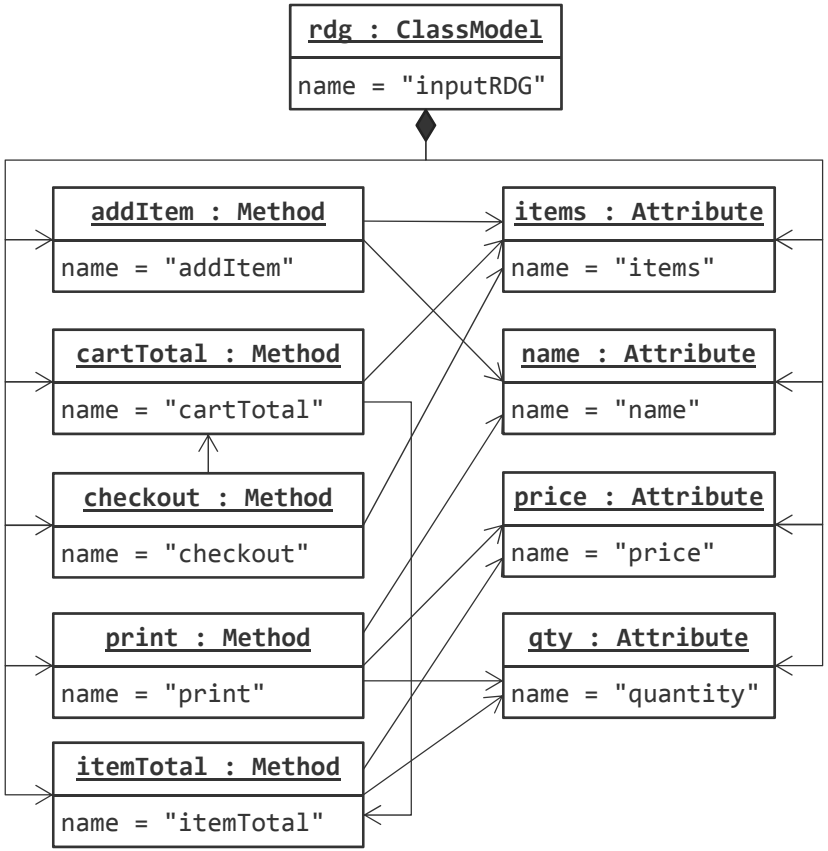
$$DMM(m_i, m_j) = \begin{cases} 1 & \text{if there is a dependency between method } m_i \text{ and } m_j \\ 0 & \text{otherwise} \end{cases}$$

Hamid Masoud and Saeed Jalili. A clustering-based model for class responsibility assignment problem in object-oriented analysis. Journal of Systems and Software, 93(0):110 – 131, 2014.

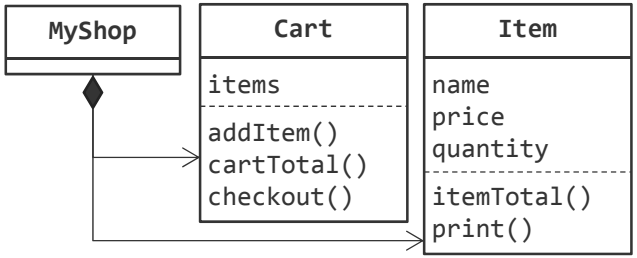
Modeling the CRA Problem

Example

RDG



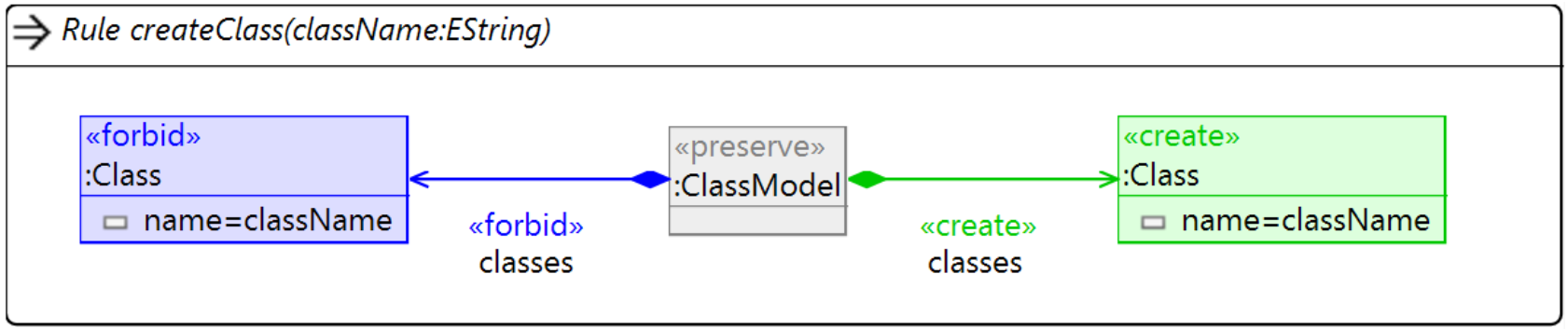
CD

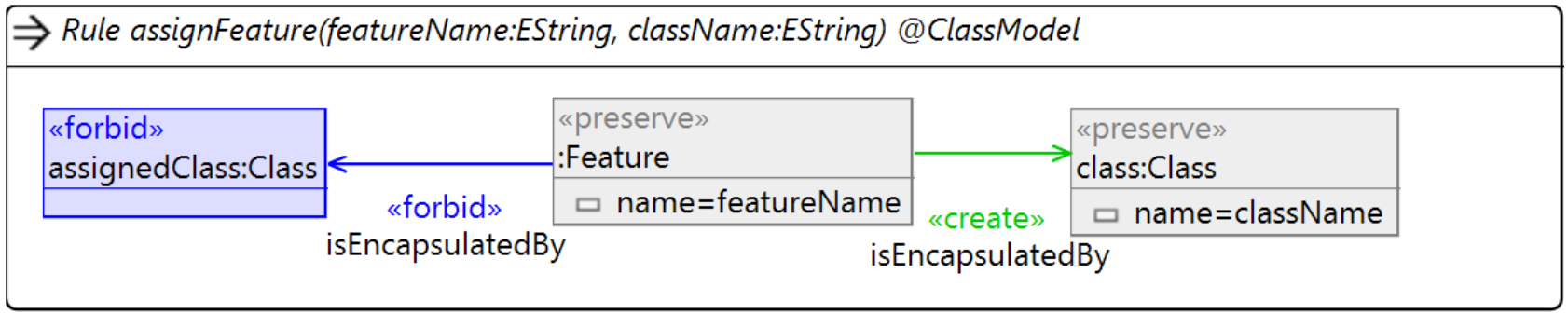


	Cart	Item
MAI(c_i, c_i)	3	5
MMI(c_i, c_i)	1	0
CohesionRatio	1.1667	0.8333
MAI(c_i, c_j)	1	0
MMI(c_i, c_j)	1	0
CouplingRatio	0.4444	0
∑ CohesionRatio	2	
∑ CouplingRatio	0.4444	
CRA-Index	1.5556	

Modeling the CRA Problem

- **Example rules**
  - Shown in Henshin syntax
  - Many other possibilities to solve CRA

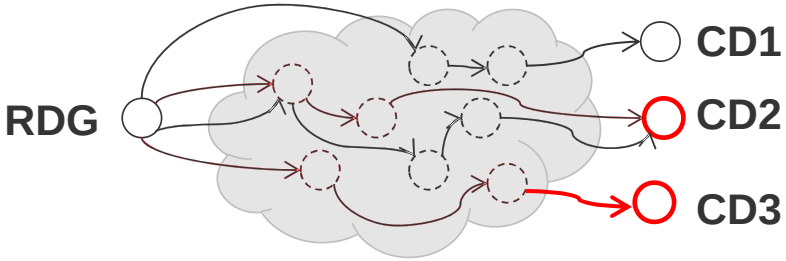




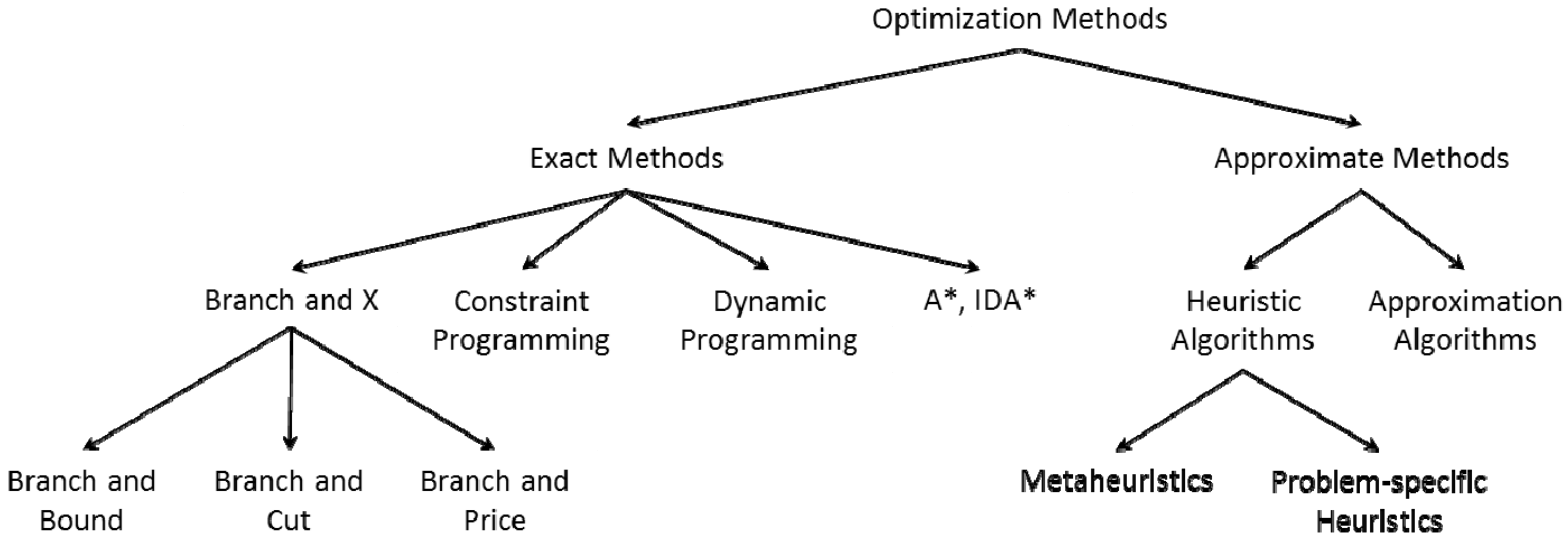
Solving the CRA Problem

Going beyond Random Search

- **Main challenge:** Finding the best output model –> finding the best rule application sequence



- **Optimization methods** to orchestrate transformation rules?



El-Ghazali Talbi, Metaheuristics: From Design to Implementation, Wiley Publishing, 2009.

Evaluation

Input Models and Evaluation Schema

- **Example Models**

	Input A	Input B	Input C	Input D	Input E
Attributes	5	10	20	40	80
Methods	4	8	15	40	80
Data Dep.	8	15	50	150	300
Functional Dep.	6	15	50	150	300

- **Evaluation Schema**

Criteria	Weight	MaxPoints	Total
Completeness & Correctness	1	10	10
Optimality	3	10	30
Complexity	2	10	20
Flexibility	2	10	20
Performance	2	10	20
Total			100

Solutions

http://www.transformation-tool-contest.eu/solutions_cra.html

- **NMF** by Georg Hinkel
- **VIATRA** by András Szabolcs Nagy et al.
- **UML-RSDS** by Kevin Lano et al.
- **ATL/Java** by Leif Arne Johnsen et al.
- **Excel** by Maximiliano Vela et al.
- **SDMLib** by Christoph Eickhoff et al.
- **MDEOptimiser** by Alexandru Burdusel et al.
- **Henshin** by Kristopher Born et al.
- **SIGMA** by Filip Krikava