Software Product Line Process

Eduardo Figueiredo

http://www.dcc.ufmg.br/~figueiredo

Development Process

- Most processes of traditional software engineering target a single system
  - The life cycle of a software product
- For Software Procut Lines (SPL), we must change our way of thinking
  - We have to look at a variety of systems that are similar, but not identical

Domain in SPL

- The proper scoping of the target domain is essential in SPL
- The broader the domain, the smaller is the set of similarities
- SPL development has two phases
  - Domain Engineering
  - Application Engineering

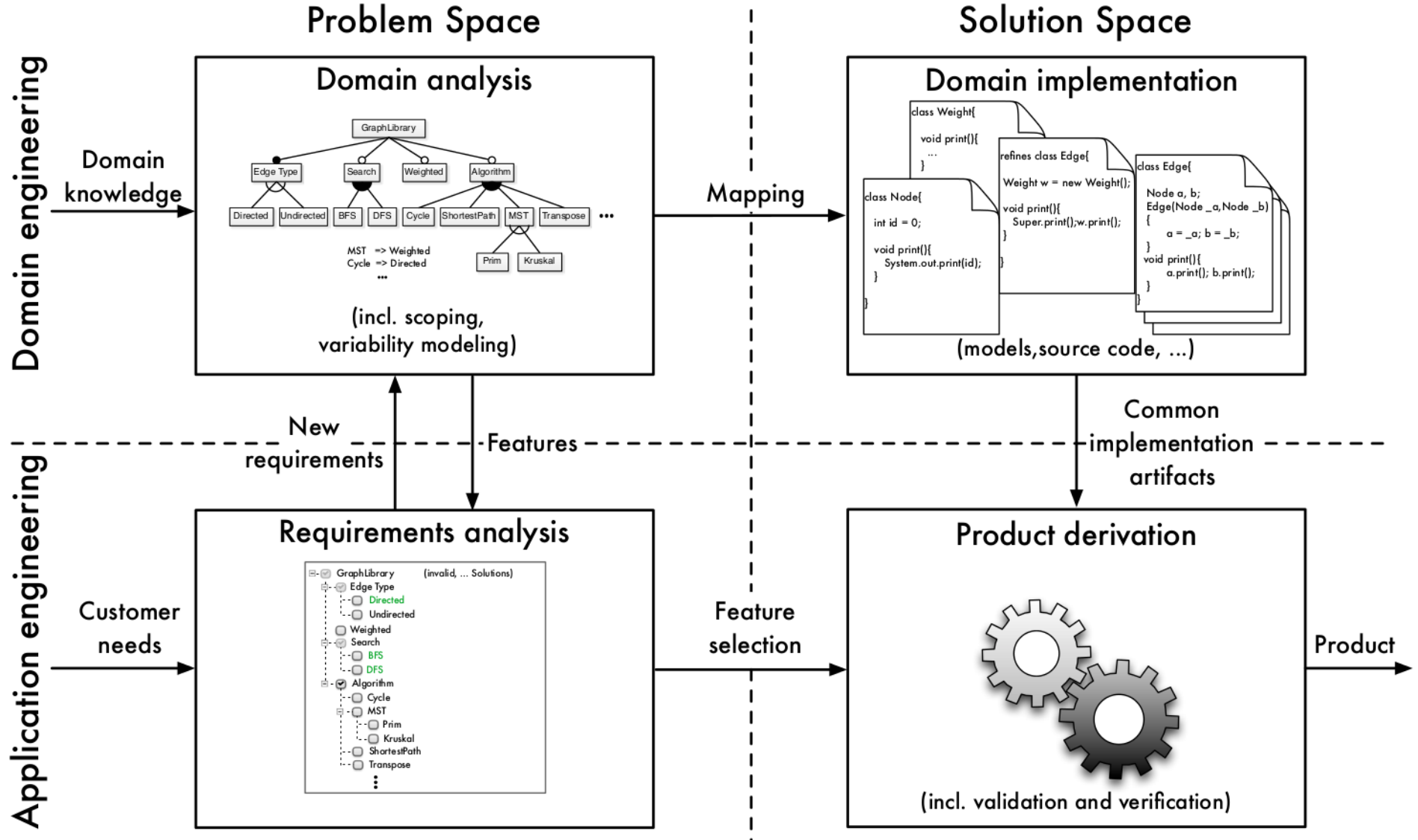
Domain Engineering

- The process of analyzing the domain of a product line and developing reusable artifacts
- Domain Engineering does not result in a specific product
  - It prepares artifacts to be used in multiple products
- It targets development **for reuse**

Application Engineering

- Application engineering is the process of developing a specific product
  - It addresses the needs of a customer
  - It repeats for every product
- Similar to a development process in traditional software engineering
  - But, in SPL, it reuses artifacts from domain engineering
- It targets development **with reuse**

Engineering Process of SPL

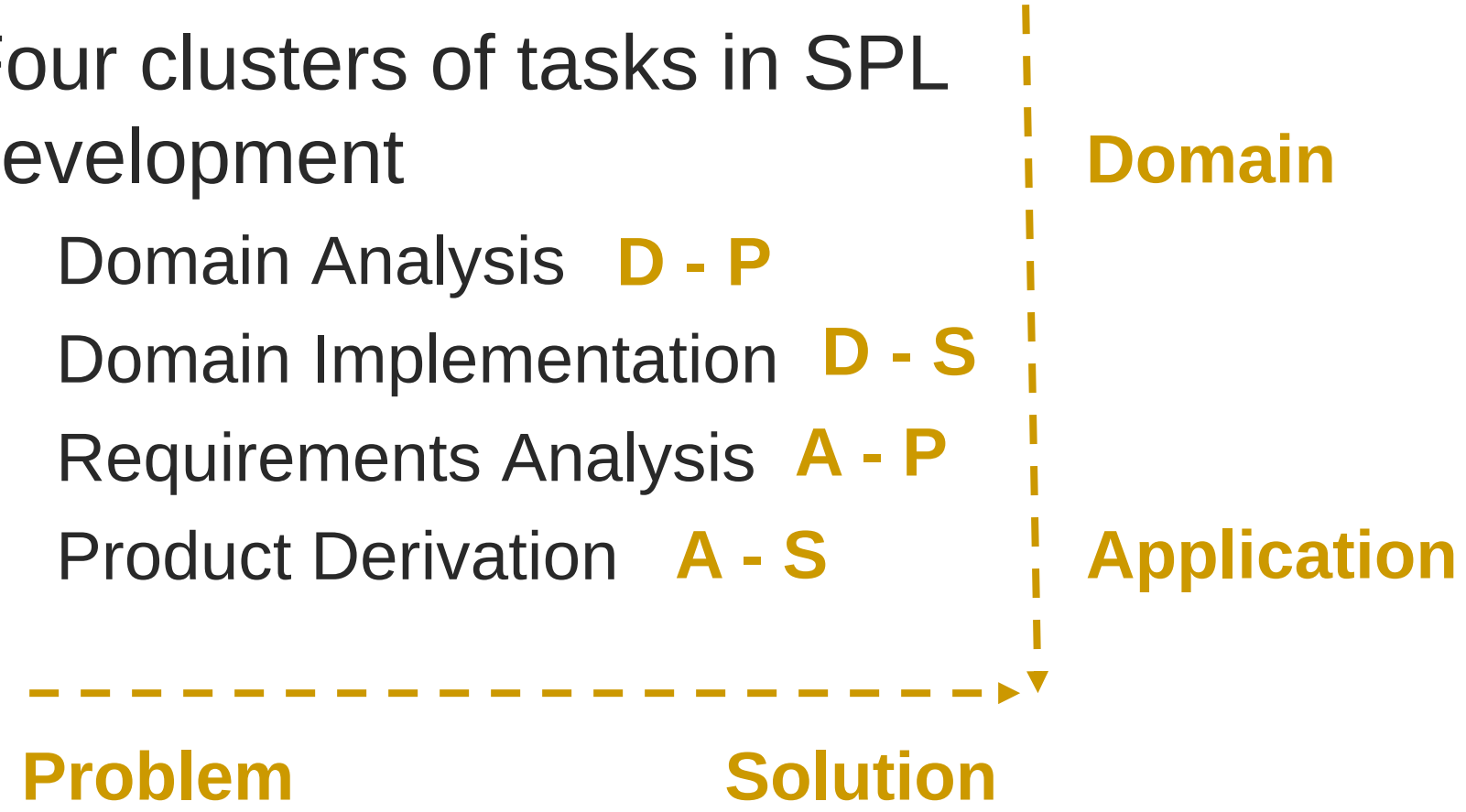


Problem vs. Solution Space

- Problem space is the perspective of stakeholders and their problems
  - Features are domain abstractions in the problem space
- Solution space deals with design, implementation, and testing
  - In SPL, features and their combinations are considered to facilitate systematic reuse

Tasks in SPL Development

- Four clusters of tasks in SPL development
  1. Domain Analysis **D - P**
  2. Domain Implementation **D - S**
  3. Requirements Analysis **A - P**
  4. Product Derivation **A - S**



Domain Analysis

- It is similar to requirements engineering for an entire SPL
- In domain analysis, we need to decide the scope of the domain
- The result of domain analysis is usually a feature model

Domain Implementation

- It is the process of developing reusable artifacts
  - Development for reuse
- In domain implementation, many kinds of artifacts can be developed
  - UML design models, source code, automated tests, documentation, etc.

Requirements Analysis

- It investigates the needs of a customer
- In requirements analysis, customer requirements are mapped to feature selection
- If new requirements are discovered, the feature model can be modified

Product Derivation

- Reusable artifacts are combined according to the results of requirements analysis
- This process can be automated, but it may require customization tasks

Bibliography

- S. Apel, D. Batory, C. Kastner, G. Saake. **Feature-Oriented Software Product Lines: Concepts and Implementation**. Springer; 2013.
  - Section 2.2