FODA: Twenty Years of Perspective on Feature Models

Kyo Chul Kang

SPLC 2009
August 24-28, 2009
San Francisco, CA, USA

Pohang University of Science and Technology
(POSTECH)
Prologue

Stand on the shoulders of giants

Google Home - About Google - About Google Scholar - Go to Google Scholar

©2009 Google
Stand on the shoulders of giants

Google Home - About Google - About Google Scholar - Go to Google Scholar

©2009 Google
1533 Citations! (August 12, 2009)
Prologue

Feature-Oriented Domain Analysis (FODA)
Feasibility Study

Kyo C. Kang
Sholom G. Cohen
James A. Hess
William E. Novak
A. Spencer Peterson
Domain Analysis Project

Technical Report
CMU/SEICN-TR-21
ESD-89-TR-222
November 1990

Approved for public release.
Distribution unlimited.

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
Agenda

➢ Introduction

➢ Looking Back
  • Number of Citations
  • Feature Model Genealogy
  • Why Popular?
  • Salient Features of FODA Report

➢ Looking Forward
  • Future Works
  • Other Issues

➢ Acknowledgement
Agenda

- Introduction
- Looking Back
  - Number of Citations
  - Feature Model Genealogy
  - Why Popular?
  - Salient Features of FODA Report
- Looking Forward
  - Future works
  - Other Issues
- Acknowledgement
Feature-Oriented Domain Analysis (FODA) Feasibility Study

Kang, K.
Cohen, S.
Hess, J.
Novak, W.
Peterson, A.

Technical Report
CMU/SEI-90-TR-021

This document is unavailable online. Please refer to the instructions for purchasing paper copies of SEI documents.

Successful software reuse requires the systematic discovery and exploitation of commonality across related software systems. By examining related software systems and the underlying theory of the class of systems they represent, domain analysis can provide a generic description of the requirements of that class of systems and a set of approaches for their implementation. This report will establish methods for performing a domain analysis and describe the products of the domain analysis process. To illustrate the application of domain analysis to a representative class of software systems, this report will provide a domain analysis of window management system software.

© 2000 Carnegie Mellon University

Terms of Use | Privacy Statement
Agenda

- Introduction

- Looking Back
  - Number of Citations
  - Feature Model Genealogy
  - Why Popular?
  - Salient Features of FODA Report

- Looking Forward
  - Future Works
  - Other Issues

- Acknowledgement
Prologue

1314 Citations! (June 06, 2009)
Number of Citations

1990~2009 (June 6)
Total: 1314
Unknown/Overlapping: 442
Number of Citations

1990~2008 (June 6)
Total: 1280
Unknown/Overlapping: 442

\[ y = 1.1754e^{0.2704x} \]
\[ R^2 = 0.9255 \]
Regional Differences

1990~2009 (June 6)
Total: 1314
Unknown/Overlapping: 456
Number of Citations

Industry vs. Academia

- **Academia**: 714
- **Industry**: 53
- **Academia & Industry**: 84

1990~2009 (June 6)
Total: 1314
Unknown/Overlapping: 467
Source Differences

Number of Citations

1990~2009 (June 6)
Total: 1314
Unknown/Overlapping: 425
Not-English: 81
Subject Differences

Subject Categories

- Feature Modeling
- Feature Model Formalization
- Feature Model Extension
- Generative Programming
- Domain-Specific Language
- Feature-Oriented Design/Programming
- Product Configuration
- Variability Management
- Product Line Methodology
- Product Line Adoption
- Domain Analysis
- Requirements Analysis
- Domain-Specific Architecture
- Reusable Component Development
- Refactoring/Reengineering
- Tool Development
- Business Application
- Embedded Application
- SOA
- Other
Subject Differences

<table>
<thead>
<tr>
<th>Subject</th>
<th>Citations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain Analysis/Requirements Analysis</td>
<td>166</td>
</tr>
<tr>
<td>Variability Management</td>
<td>129</td>
</tr>
<tr>
<td>Business Application</td>
<td>75</td>
</tr>
<tr>
<td>Reusable Component Development</td>
<td>67</td>
</tr>
<tr>
<td>Tool Development</td>
<td>62</td>
</tr>
<tr>
<td>Generative Programming</td>
<td>59</td>
</tr>
<tr>
<td>Product Line Methodology/Product Line Adoption</td>
<td>53</td>
</tr>
<tr>
<td>Feature Model Formalization</td>
<td>49</td>
</tr>
<tr>
<td>Domain-Specific Architecture</td>
<td>47</td>
</tr>
<tr>
<td>Embedded Application</td>
<td>46</td>
</tr>
<tr>
<td>Product Configuration</td>
<td>42</td>
</tr>
<tr>
<td>Feature Modeling</td>
<td>34</td>
</tr>
<tr>
<td>Feature Model Extension</td>
<td>33</td>
</tr>
<tr>
<td>Feature-Oriented Design/Programming</td>
<td>31</td>
</tr>
<tr>
<td>Domain-Specific Language</td>
<td>22</td>
</tr>
<tr>
<td>Refactoring/Reengineering</td>
<td>16</td>
</tr>
<tr>
<td>SOA</td>
<td>8</td>
</tr>
</tbody>
</table>

1990~2009 (June 6)

Total: 1314
Unknown/Overlapping: 450
Not-English: 56
Others: 51
Top-30 People

Number of Citations

Authors

1990~2009 (June 6)
Total: 1314
Unknown/Overlapping: 415
Survey: Top-30 People

1. What are the contributions of FODA?

2. What are the remaining problems for FODA?

3. What is your own one-sentence definition/feeling about the feature model?

4. Anything else you would suggest for inclusion?
The Contributions of FODA:

Systematic domain analysis method used in software product line engineering. (C Kastner, S Apel, C Lengauer)

Strongly influencing how software configuration is seen in software product line, and popularizing the domain analysis concept. (K Schmid)

Laying the groundwork for feature analysis and feature modeling. (J Van Gurp)

Feature modeling:
- Essential technique for defining the space of programs that define a software product line. (D Batory)
- Simple but comprehensive way to modeling commonalities and variabilities in a domain. (H Zhang, J Lee, S Jarzabek, T Asikainen)
- Easy to use and communicate between stakeholders. (C Kastner)
- Giving a name to a fundamental form of modularity in context of product lines. (D Batory)
The Remaining Problems for FODA:

Addressing other parts of the life cycle (Especially application engineering). (K Schmid)

Clear mapping between features and software artifacts. (J Van Gurp, K Schmid)

Standardization of feature model extensions; Trade-off between expressiveness and simplicity. (C Kastner, S Apel, C Lengauer, T Asikainen, K Schmid)

Scalability of feature model. (C Kastner, D Batory)

Managing Complexity in view of many inter-dependent features. (J Van Gurp, S Jarzabek)

Feature model evaluation. (J Lee)

Integration with UML Model. (S Jarzabek)

Good teaching materials. (D Batory)
Own One-Sentence Definition/Feeling about the Feature Model:

A model that provides the **foundations of software reuse**. (J Lee)

A **simple means to describe the commonalities and variabilities of a domain / product line**. (S Apel, C Lengauer)

**Good and easy to understand, practical and fundamental.** (H Zhang)

**Easy to use** (management-compatible) graphical model to describe variability. (C Kastner)

**Great notation, accepted by all in the field.** (S Jarzabek)

One of the **useful ways to analyze requirements (required features)** as well as a means to describe given software in terms of features (**provided features**). (J Van Gurp)

**An idea well received by the community but still lacking content meeting the scientific standards.** (T Asikainen)

**Great approach to make one understand the core idea of product line engineering** (But I’m not sure whether it is the right way of looking at variability for the actual development). (K Schmid)
Feature Model Genealogy

Original Feature Model (FODA)
(KC Kang et al., 1990)

FORM Feature Model
(KC Kang et al., 1998)

FeatuRSEB Feature Model
(ML Griss et al., 1998)

Hein et al. Model
(A Hein et al., 2000)

Riebisch et al. Feature Model
(M Riebisch et al., 2002)

GP-Extended Feature Model
(K Czarnecki et al., 2002)

Cardinality-Based Feature Model
(K Czarnecki et al., 2004)

PLUSS Feature Model
(M Eriksson et al., 2005)

Van Gurp et al. Feature Model
(J van Gurp et al., 2001)

Generating Programming (GP) Feature Model
(K Czarnecki et al., 2000)

Benavides et al. Feature Model
(D Benavides et al., 2005)
Feature Model Genealogy

Original Feature Model
(KC Kang et al., 1990)

- FORM Feature Model
  (KC Kang et al., 1998)

- FeatuRSEB Feature Model
  (ML Griss et al., 1998)

- Hein et al. Model
  (A Hein et al., 2000)

- Van Gurp et al. Feature Model
  (J van Gurp et al., 2001)

- Riebisch et al. Feature Model
  (M Riebisch et al., 2002)

- GP-Extended Feature Model
  (K Czarnecki et al., 2002)

- Cardinality-Based Feature Model
  (K Czarnecki et al., 2004)

- PLUSS Feature Model
  (M Eriksson et al., 2005)

- Generative Programming (GP) Feature Model
  (K Czarnecki et al., 2000)

- Benavides et al. Feature Model
  (D Benavides et al., 2005)
Feature Model Genealogy

Original Feature Model
(KC Kang et al., 1990)

- Feature diagram: A Graphical And/Or Hierarchy of Features
  - Mandatory / Optional / Alternative Feature
  - Composed-of Relationship
- Composition Rules: Mutual Dependency (Requires) and Mutual Exclusion (Mutex-with) Relationships
- Issues and Decisions: Record of Trade-offs, Rationales, and Justifications
- System Feature Catalogue: Record of Existing System Features
Feature Model Genealogy

Extensions from Original Feature Model

- Diagram Shape
- Layer
- Relationship Type
- Feature Type
- Feature Attribute
- Feature Cardinality
- Feature Group and Group Cardinality
- Constraint Notation
- Binding Time Notation
Feature Model Genealogy

Original Feature Model
(KC Kang et al., 1990)

FORM Feature Model
(KC Kang et al., 1998)

FeatuRSEB Feature Model
(ML Griss et al., 1998)

Hein et al. Model
(A Hein et al., 2000)

Riebisch et al. Feature Model
(M Riebisch et al., 2002)

GP-Extended Feature Model
(K Czarnecki et al., 2002)

Cardinality-Based Feature Model
(K Czarnecki et al., 2004)

PLUSS Feature Model
(M Eriksson et al., 2005)

Benavides et al. Feature Model
(D Benavides et al., 2005)
Feature Model Genealogy

FORM Feature Model (KC Kang et al., 1998)

- Layer
  - Capability
  - Operating Environment
  - Domain Technology
  - Implementation Technology

- Relationship Type:
  - Implemented-by
Feature Model Genealogy

- Original Feature Model (KC Kang et al., 1990)
  - FORM Feature Model (KC Kang et al., 1998)
    - FeatuRSEB Feature Model (ML Griss et al., 1998)
      - Hein et al. Model (A Hein et al., 2000)
        - Riebisch et al. Feature Model (M Riebisch et al., 2002)
          - GP-Extended Feature Model (K Czarnecki et al., 2002)
          - Cardinality-Based Feature Model (K Czarnecki et al., 2004)
      - Van Gurp et al. Feature Model (J van Gurp et al., 2001)
        - PLUSS Feature Model (M Eriksson et al., 2005)
        - Benavides et al. Feature Model (D Benavides et al., 2005)
  - Generative Programming (GP) Feature Model (K Czarnecki et al., 2000)
Feature Model Genealogy

FeatuRSEB Feature Model
(ML Griss et al., 1998)

Combine FODA and the Reuse-Driven Software Engineering Business (RSEB)

- Feature Type
  - Alternative Feature
    → Variation Point Feature / Variant Feature

- Constraint Notation (with Dashed Arrow)

- Bound Time Notation
  - Reuse Time Bound (XORed-disjunction)
  - Use Time Bound (ORed-disjunction)
Van Gurp et al.
Feature Model
(J van Gurp et al., 2001)

- Feature Type
  - External Feature
  - Alternative Feature
    → OR Specialization/ XOR Specialization
- Binding Time Notation
Feature Model Genealogy

PLUSS Feature Model (M Eriksson et al., 2005)

- Diagram Shape
  - Feature as Circle (Black, White, ‘S’, ‘M’)

- Feature Type
  - Alternative Feature
    → Single Adapter / Multiple Adapter

- Constraint Notation
Feature Model Genealogy

Original Feature Model
(KC Kang et al., 1990)

- FORM Feature Model
  (KC Kang et al., 1998)

- FeatuRSEB Feature Model
  (ML Griss et al., 1998)

- Hein et al. Model
  (A Hein et al., 2000)

- VAN Gurp et al. Feature Model
  (J van Gurp et al., 2001)

- Riebisch et al. Feature Model
  (M Riebisch et al., 2002)

- GP-Extended Feature Model
  (K Czarnecki et al., 2002)

- Cardinality-Based Feature Model
  (K Czarnecki et al., 2004)

- PLUSS Feature Model
  (M Eriksson et al., 2005)

- Benavides et al. Feature Model
  (D Benavides et al., 2005)
Hein et al.
Feature Model
(A Hein et al., 2000)

Use of UML

- Relationship Type
  - Arrow for Secondary Structure

UML Feature Meta Model

“f6” is “Optional Compound” in Primary Structure and “Alternative” in Secondary Structure

“f6” is “Optional Compound” in Primary Structure and “Alternative” in Secondary Structure
Feature Model Genealogy

Original Feature Model
(KC Kang et al., 1990)

FORM Feature Model
(KC Kang et al., 1998)

FeatuRSEB Feature Model
(ML Griss et al., 1998)

Hein et al. Model
(A Hein et al., 2000)

Van Gurp et al. Feature Model
(J van Gurp et al., 2001)

Riebisch et al. Feature Model
(M Riebisch et al., 2002)

GP-Extended Feature Model
(K Czarnecki et al., 2002)

Cardinality-Based Feature Model
(K Czarnecki et al., 2004)

PLUSS Feature Model
(M Eriksson et al., 2005)

Benavides et al. Feature Model
(D Benavides et al., 2005)

Generative Programming (GP) Feature Model
(K Czarnecki et al., 2000)
Feature Model Genealogy

Generative Programming
Feature Model
(K Czarnecki et al., 2000)

- Diagram Shape
  - Feature Name in a Box
  - Box with Circle
    - Black = Mandatory
    - White = Optional

- Feature Type
  - OR Feature (Black Triangle)
Feature Model Genealogy

Riebisch et al. Feature Model (M Riebisch et al., 2002)

- Diagram Shape / Feature Type
  - Inherit GP Feature Model
- Feature Group and Group Cardinality
- Constraint Notation

GP Extended Feature Model (K Czarnecki et al., 2002)

- Diagram Shape / Feature Type
  - Inherit GP Feature Model
- Feature Attributes
- Feature Cardinality
Cardinality-Based Feature Model
(K Czarnecki et al., 2004)

- Diagram Shape / Feature Type
  - Inherit GP Feature Model
- Relationship Type
  - Feature Diagram Reference (Dashed Line)
- Feature Cardinality
- Feature Group and Group Cardinality
Feature Model Genealogy

Original Feature Model
(KC Kang et al., 1990)

- FORM Feature Model
  (KC Kang et al., 1998)
- FeatuRSEB Feature Model
  (ML Griss et al., 1998)
- Hein et al. Model
  (A Hein et al., 2000)

- Van Gurp et al. Feature Model
  (J van Gurp et al., 2001)

- Generative Programming (GP) Feature Model
  (K Czarnecki et al., 2000)

- Riebisch et al. Feature Model
  (M Riebisch et al., 2002)

- GP-Extended Feature Model
  (K Czarnecki et al., 2002)

- Cardinality-Based Feature Model
  (K Czarnecki et al., 2004)

- PLUSS Feature Model
  (M Eriksson et al., 2005)

- Benavides et al. Feature Model
  (D Benavides et al., 2005)
Feature Model Genealogy

Benavides et al. Feature Model
(D Benavides et al., 2005)

- Diagram Shape / Feature Type
  - Inherit GP Feature Model

- Relationship Type
  - Attribute Relationship (with Dashed Line)

- Feature Attribute
Why Popular?

- Codification of the Most Critical Information for Reuse
  - Commonality and Variability
  - Medium for Identifying Variation Points and Variants
- Simplicity
- Understandability
  - Intuitive
- Practicality
- Applicability
Salient Features of FODA Report

- "For example, features from the window manager domain such as `constrainedMove` and `zapEffect` could have been specified more precisely using a formal specification technique."
  ➔ Formalization

- "If the domain is well-defined and is expected to remain stable, a preprocessor or an application generator development technique might be appropriate to process the compile-time features."
  ➔ Generative Programming

- "The description should also indicate whether it is a compile-time, an activation-time, or a runtime feature."
  ➔ Binding time, Dynamic Product Line
Salient Features of FODA Report

- “The classification of the features can be used in the components construction for modularization and for selection of appropriate development techniques.” → Component Development

- “A record of the issues and decisions that arise in the course of the feature analysis must be incorporated into the feature model to provide the rationale for choosing options and selecting among several alternatives.” → Configuration Decision Support
Agenda

- Introduction
- Looking Back
  - Number of Citations
  - Feature Model Genealogy
  - Why Popular?
  - Salient Features of FODA Report
- Looking Forward
  - Future Works
  - Other Issues
- Acknowledgement
Feature Modeling and Analysis

Future Works

Feature analysis for different aspects at different phases of the life-cycle

<table>
<thead>
<tr>
<th>PL Contexts</th>
<th>Requirements</th>
<th>Design</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legal / Social Constraint</td>
<td>Domain model</td>
<td>Architecture</td>
<td>Modules / Algorithms</td>
</tr>
<tr>
<td>User Profile</td>
<td>Capability</td>
<td>Component</td>
<td></td>
</tr>
<tr>
<td>Market / Business</td>
<td>Use Case</td>
<td>Variation Point / Variants</td>
<td></td>
</tr>
<tr>
<td>Usage</td>
<td>Configuration View</td>
<td>FM-VP Decision Dependency Analysis (Consistency)</td>
<td>Platform Conformance Analysis</td>
</tr>
<tr>
<td>Operating Environment</td>
<td>Structural View</td>
<td>QA Conformance Analysis</td>
<td>Variation Support Analysis</td>
</tr>
<tr>
<td>Computing Resource / Platform</td>
<td>Dependency View</td>
<td>View Consistency Analysis</td>
<td>Binding Time Support</td>
</tr>
<tr>
<td>Standards</td>
<td>Binding View</td>
<td></td>
<td>Connector Support</td>
</tr>
</tbody>
</table>

Decision Modeling (Rationales), Variability/Integrity Management
Domain analysis

- Different domains may require different approaches
  - Service analysis may be good for business applications domains
  - How about goal analysis?
  - “Goal -> Service->feature” as a unified method?

- Modeling mechanisms
  - Feature model is popular but many extensions
    - Should it be standardized?
  - Formalization

- Deciding the right level of abstraction; how to structure

- Feature explosion problem
  - How to model, analyze, and manage
  - High level of complex dependencies among them

- Feature interaction problem
Other Issues

- Goal-oriented (value-based) configuration of features
  - Knowledge-based configuration
  - Quality attributes or user-goals

- Going from domain analysis to architecture and component design
  - Designing architectures and components based on the analysis results (commonality and variability information)
    - SOA vs. agent-based vs. other architecture models
  - Building variability into architectures and components
  - Selecting appropriate mechanisms for the problem
  - Dealing with complex dependencies between features
Technology

Other Issues

- Specification of models
  - Reuse contexts and assumptions

- Verification of quality attributes of integrated systems
  - Safety, reliability, etc.
  - Detecting feature interaction problems

- Configuration management
  - Version control of components and architectures with multi-product nature
  - Evolution of the product line itself
Technology

- PL for systems in the newly emerging computing environments
  - Service Oriented Architecture
  - Ubiquitous computing environment/cloud computing
    - Dynamic binding of features
    - Run-time verification
  - From compile-time engineering to run-time engineering
    - Embedment of SE knowledge in running systems

- Tools!
Process

Other Issues

How to change to PL-based organization

• How to evolve: staged process model for reuse adoption
• Key process areas
  − Best practices
• Metrics
  − Key indicators: cost of production, time-to-market, project completion time, etc.
  − Relationship between reuse, quality, and productivity
  − Relationship between reuse and ROI for sustainability of a reuse program

Process models

• Proactive vs. reactive vs. extractive models
  − Best practices
• PL process vs. agile methods
Other Issues

Management

- Asset management (How to make PL-based development happen in an organization)- "Institutionalize” PL
  - Who should develop assets (with variation points)
  - Who should maintain assets (variability management)
  - Who will be responsible for quality assurance
  - Who should enforce the use of assets (policies)
  - Models (best practices)
    - Centralized vs. distributed

- Product line engineering in the context of a business strategy
  - “High option potentials”
  - ROI analysis
    - Estimating ROI from a reuse program
    - Estimating benefits from strategic market position

- Product line engineering in the global development environment
  - Component development outsourcing
    - Variability specification
    - Variability management
Embedment of SE Knowledge

Other Issues

Technical Trends

Manual

Ad-Hoc Approach

Systematic Approach

Context-aware Self-adaptive Software

Automatic

Incidental Application of Engineering Principles

Methods and Tools

Embedment of Software Engineering Knowledge in the System

- Modulization
- Information hiding
- Encapsulation
- ... 

- Commonality and variability analysis
- Parameterization
- Template framework
- ... 

- Monitor and dynamic reconfigurator
- Dynamic binding
- Architecture reconfiguration
- Run-time verification
- ...
Agenda

- Introduction
- Looking Back
  - Number of Citations
  - Feature Model Genealogy
  - Why Popular?
  - Salient Features of FODA Report
- Looking Forward
  - Future Works
  - Other Issues
- Acknowledgement
Acknowledgement

Special thanks to my students:

Hyesun Lee
Hyunsik Choi
Daesik Ham
Yoonho Cho
Yeonho Kim

for data analysis!
The End

Thank You!