SSC 5904 – SOFTWARE REUSE

ICMC - UNIVERSITY OF SÃO PAULO
PROF. DR. ROSANA T VACCARE BRAGA

SOFTWARE PRODUCT LINES

Slides by Prof. Dr. Jaejoon Lee – Lancaster University - UK

Lancaster University

Software Product Line Engineering: Concepts, Approaches and Feature Modelling

Dr Jaejoon Lee

School of Computing and Communications
Lancaster University

InfoLab21 | Lancaster University | Lancaster | LA1 4WA | UK
www.infolab21.lancs.ac.uk

InfoLab21
Computing Department

Today's objectives

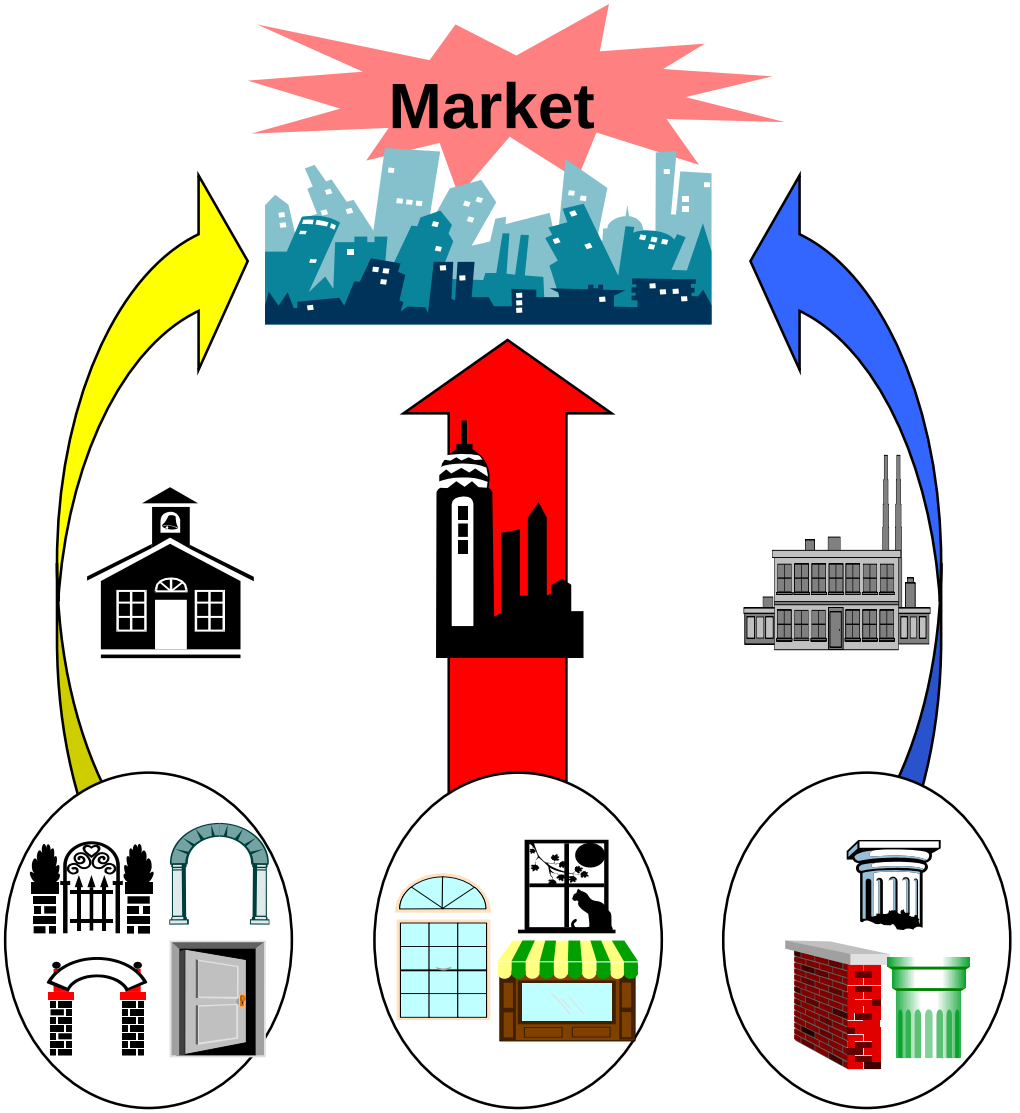
- After this lecture, you should be able to:
  - Understand the definition of a software product line (SPL)
  - Understand the software product line engineering process and three approaches
  - Understand SPL scoping and feature modeling
  - Explain the role of feature modeling in the software product line engineering process
  - Describe commonality and variability in terms of features

What is a Software Product Line ?

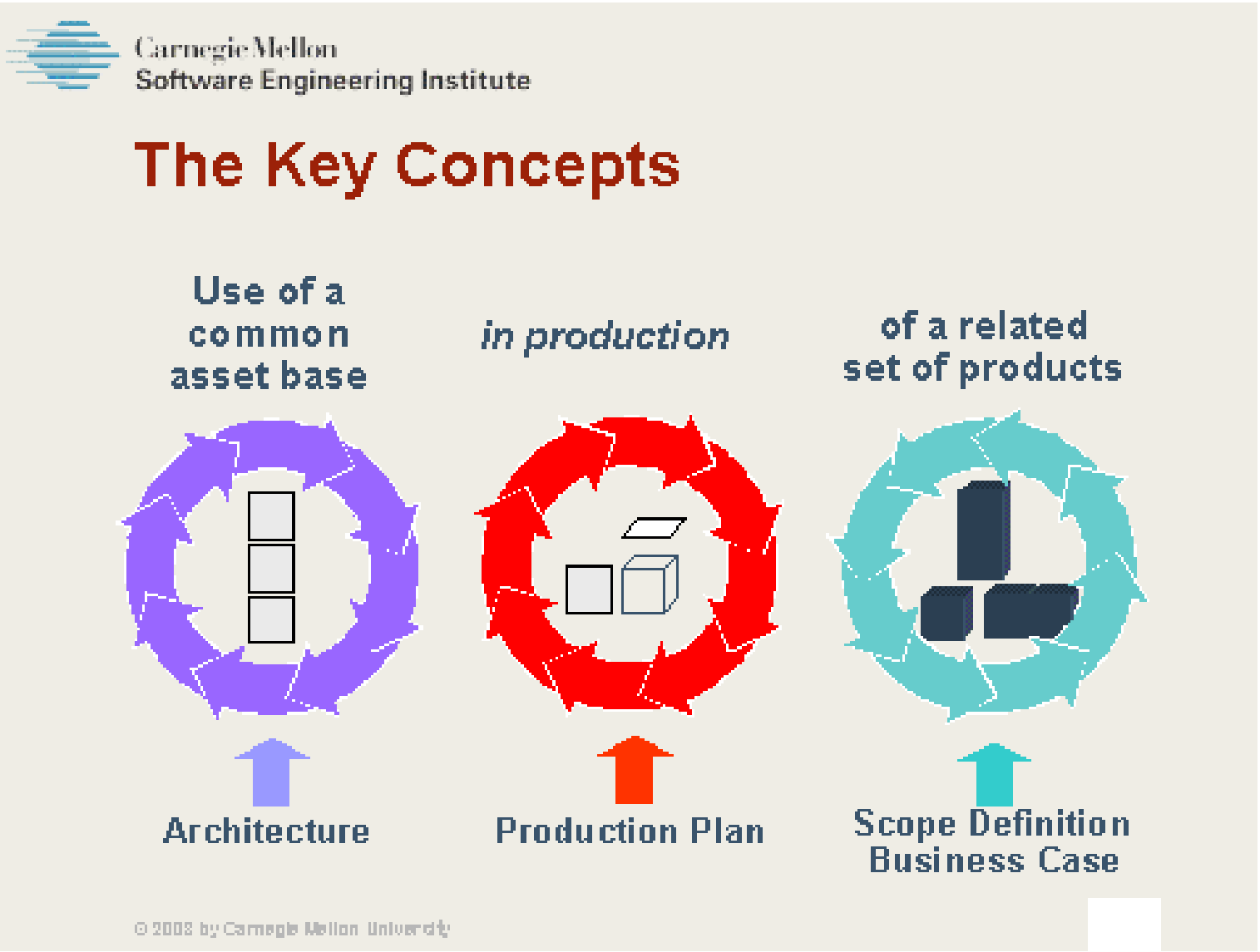
- A software product line is a *set* of software-intensive systems sharing a *common, managed set of features* that satisfy the specific needs of a *particular market* segment or mission and that are *developed from a common set of core assets in a prescribed way*.

Lawrence G. Jones and Linda M. Northrop, Software Product Lines: Capitalizing on Your Process Improvement Investment, European Software Engineering Process Group Conference, Amsterdam, Netherlands, June 2001

- Software Product Line Engineering (SPLE) is a software engineering paradigm, which guides organizations toward the development of products from core assets rather than the development of products one by one from scratch.



What is a Software Product Line ?



http://www.sei.cmu.edu/plp/essentials/

Carnegie Mellon
Software Engineering Institute

Product Line Development

http://www.sei.cmu.edu/plp/essentials/

Fraunhofer IESE: PuLSE

Fraunhofer Institut Experimentelles Software Engineering

PuLSE : Product Line Software Engineering

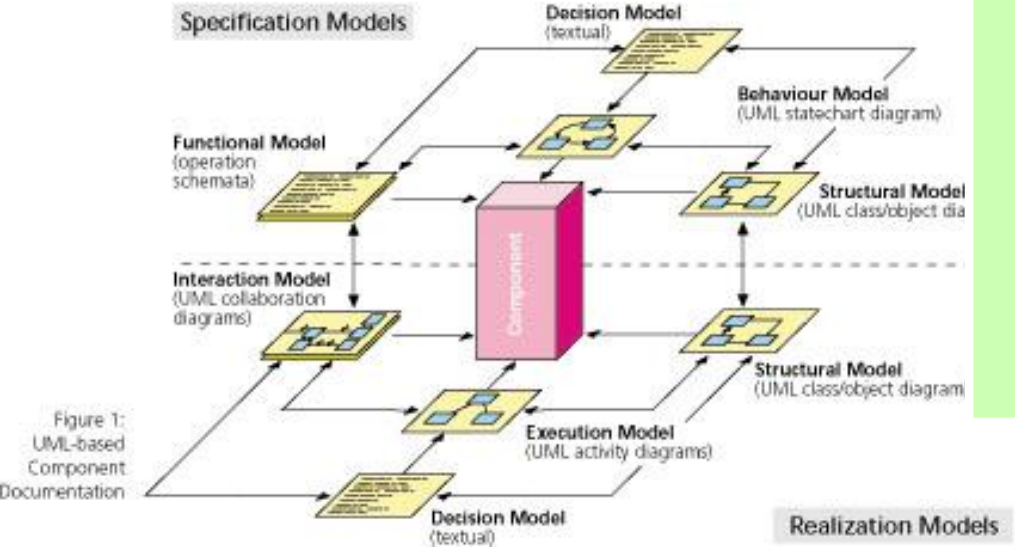
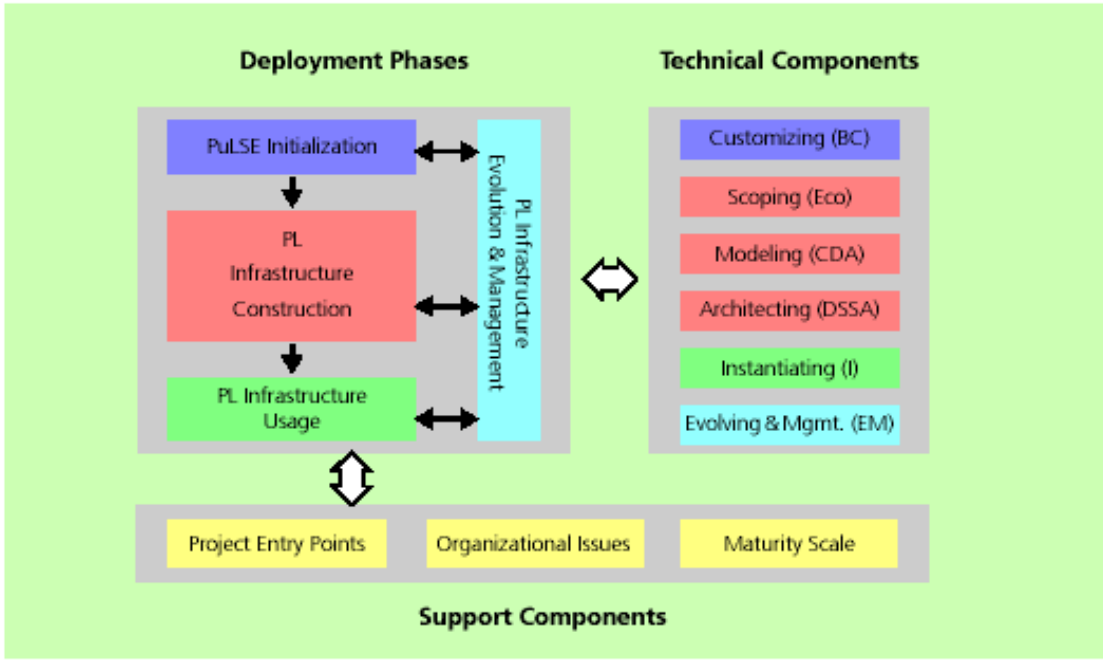


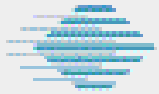
Figure 1: UML-based Component Documentation



KobrA: Komponent based application development

LANCASTER UNIVERSITY

http://www.iese.fhg.de/Products_Services/

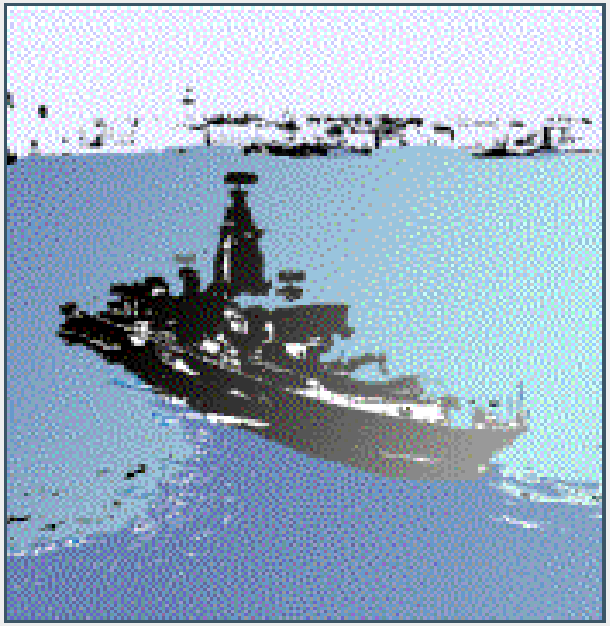


Carnegie Mellon
Software Engineering Institute

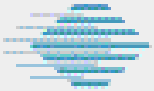
CelsiusTech: Ship System 2000

A family of 55 ship systems

integration test of 1-1.5 million SLOC requires 1-2 people
rehosting to a new platform/OS takes 3 months
cost and schedule targets are predictably met
performance/distribution behavior known in advance
customer satisfaction is high
hardware-to-software cost ratio changed from 35:65 to 80:20



http://www.sei.cmu.edu/plp/essentials/



Carnegie Mellon
Software Engineering Institute

Cummins Inc.: Diesel Engine Control Systems

Over 20 product groups with over 1000 separate engine applications

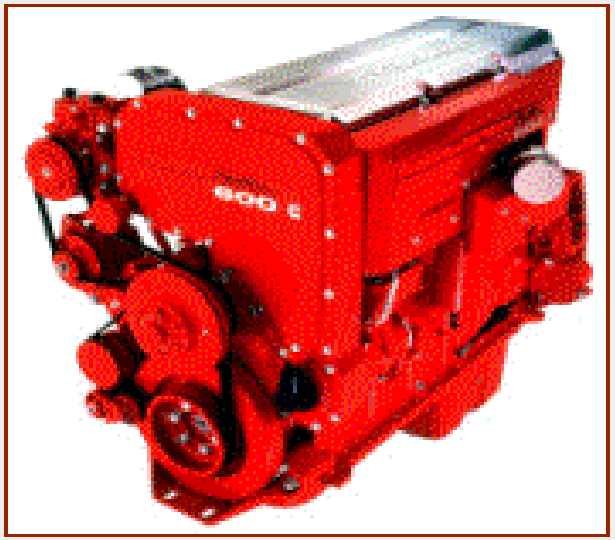
product cycle time was slashed from 250 person-months to a few person-months

Build and integration time was reduced from one year to one week

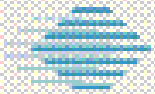
quality goals are exceeded

customer satisfaction is high

product schedules are met



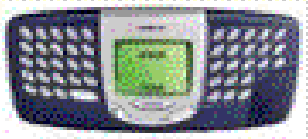
http://www.sei.cmu.edu/plp/essentials/



Carnegie Mellon
Software Engineering Institute

Nokia Mobile Phones

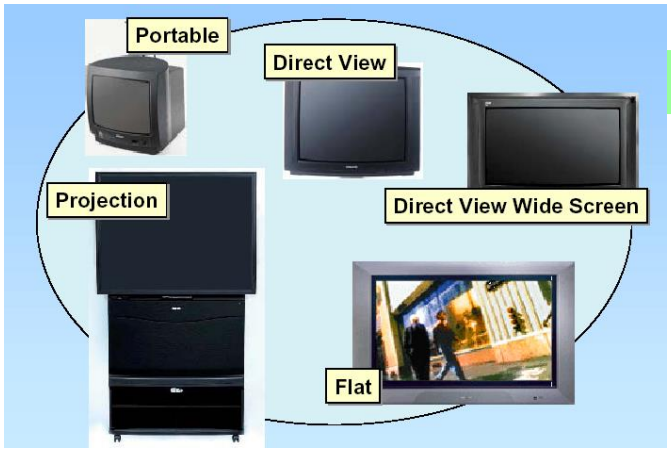
Product lines with 25-30 new products per year



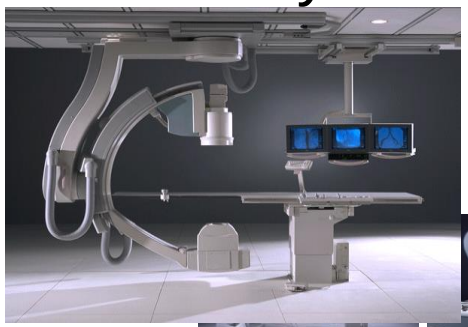
Across products there are
- varying number of keys
- varying display sizes
- varying sets of features
- 58 languages supported
- 130 countries served
- multiple protocols
- needs for backwards compatibility
- configurable features
- needs for product behavior change after release

http://www.sei.cmu.edu/plp/essentials/

TV



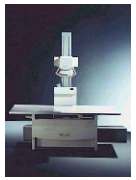
Medical Systems



X-Ray

Cardio/Vascular

Universal Radiography and Fluoroscopy



Radiography

Surgery

The Koala Component Model

14

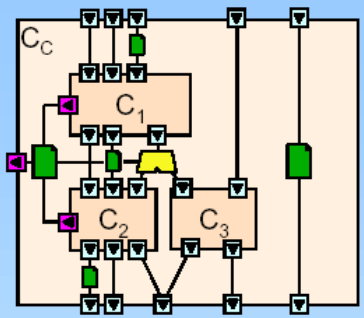
Koala offers:

a. Provides interfaces and interfaces as first class citizens

b. Requires interfaces and 3rd party binding

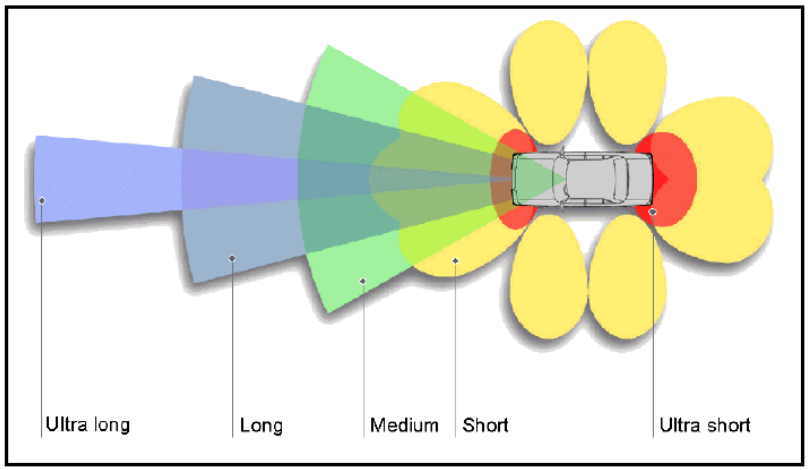
c. Aggregation and Gluing

d. Parameterization, optional interfaces and 'dynamic' binding

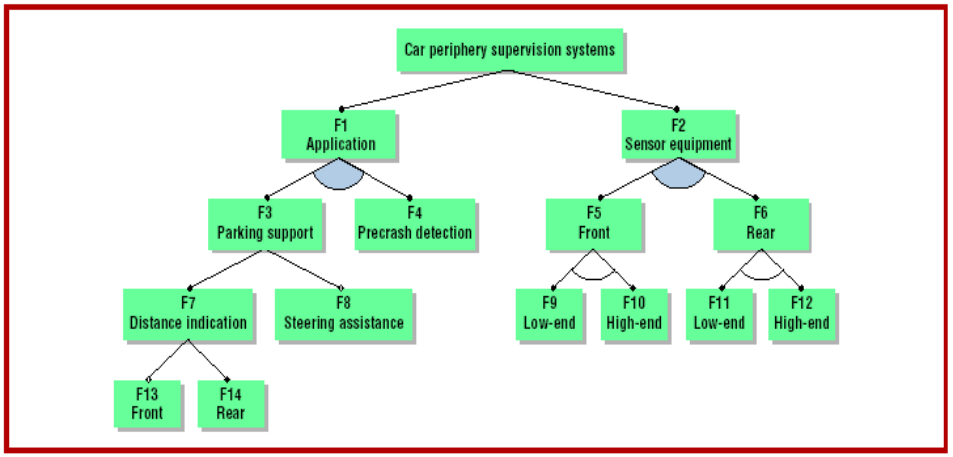


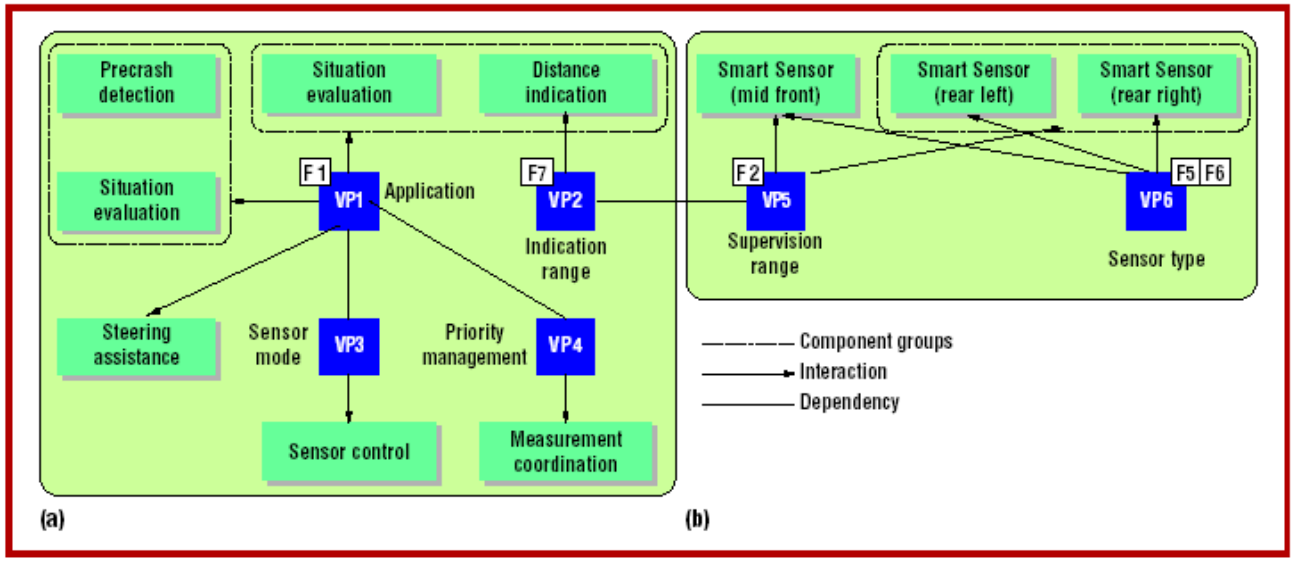
IEEE Computer March 2000

Car Periphery Supervision



Feature Modelling





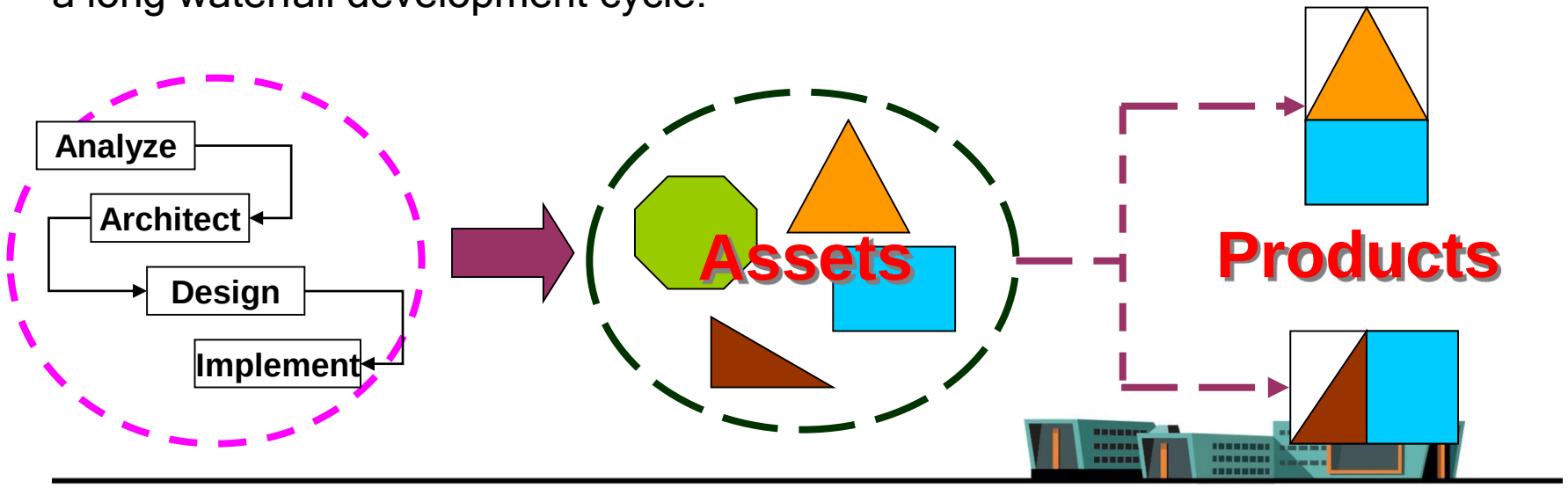
Architecture Modelling

Steffen Thiel and Andreas Hein, Modeling and Using Product Line Variability in Automotive Systems, IEEE Software, July/August 2002, pp.66-72

Product Line Approaches

* Charles Krueger, "Eliminating the Adoption Barrier", IEEE Software, Jul/Aug, 2002, pp. 29-31

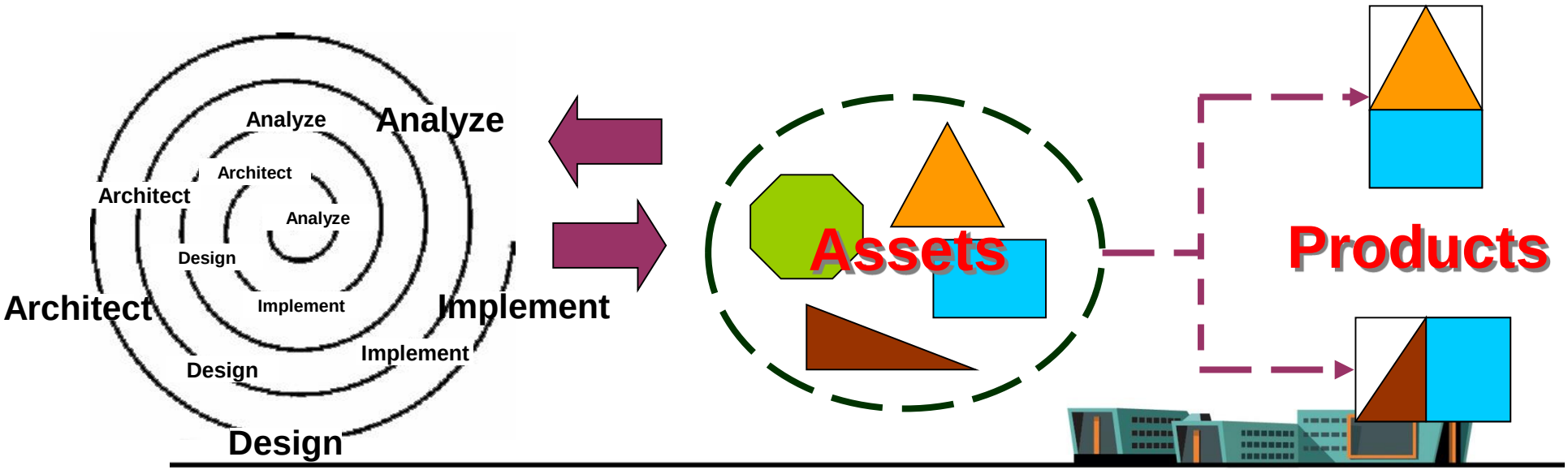
- Proactive approach*
  - The proactive approach to software product lines is like the waterfall approach to conventional software. You analyze, architect, design, and implement all product variations on the foreseeable horizon up front.
  - This approach might suit organizations that can predict their product line requirements well into the future and that have the time and resources for a long waterfall development cycle.



Product Line Approaches

*Charles Krueger, "Eliminating the Adoption Barrier", IEEE Software, Jul/Aug, 2002, pp. 29-31

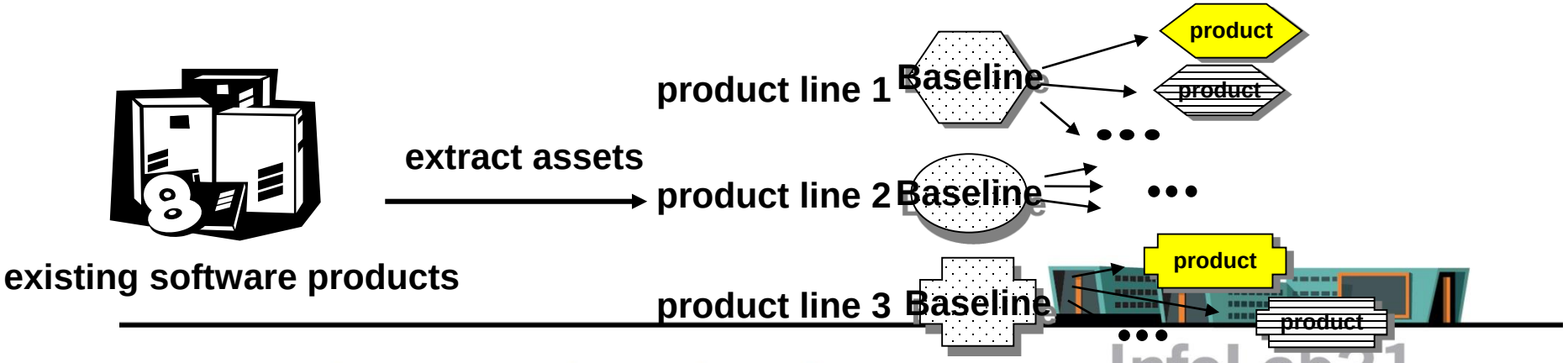
- Reactive approach*
  - The reactive approach is like the spiral or extreme programming approach to conventional. You analyze, architect, design, and implement one or several product variations on each development spiral.
  - This approach works in situations where you cannot predict the requirements for product variations well in advance.



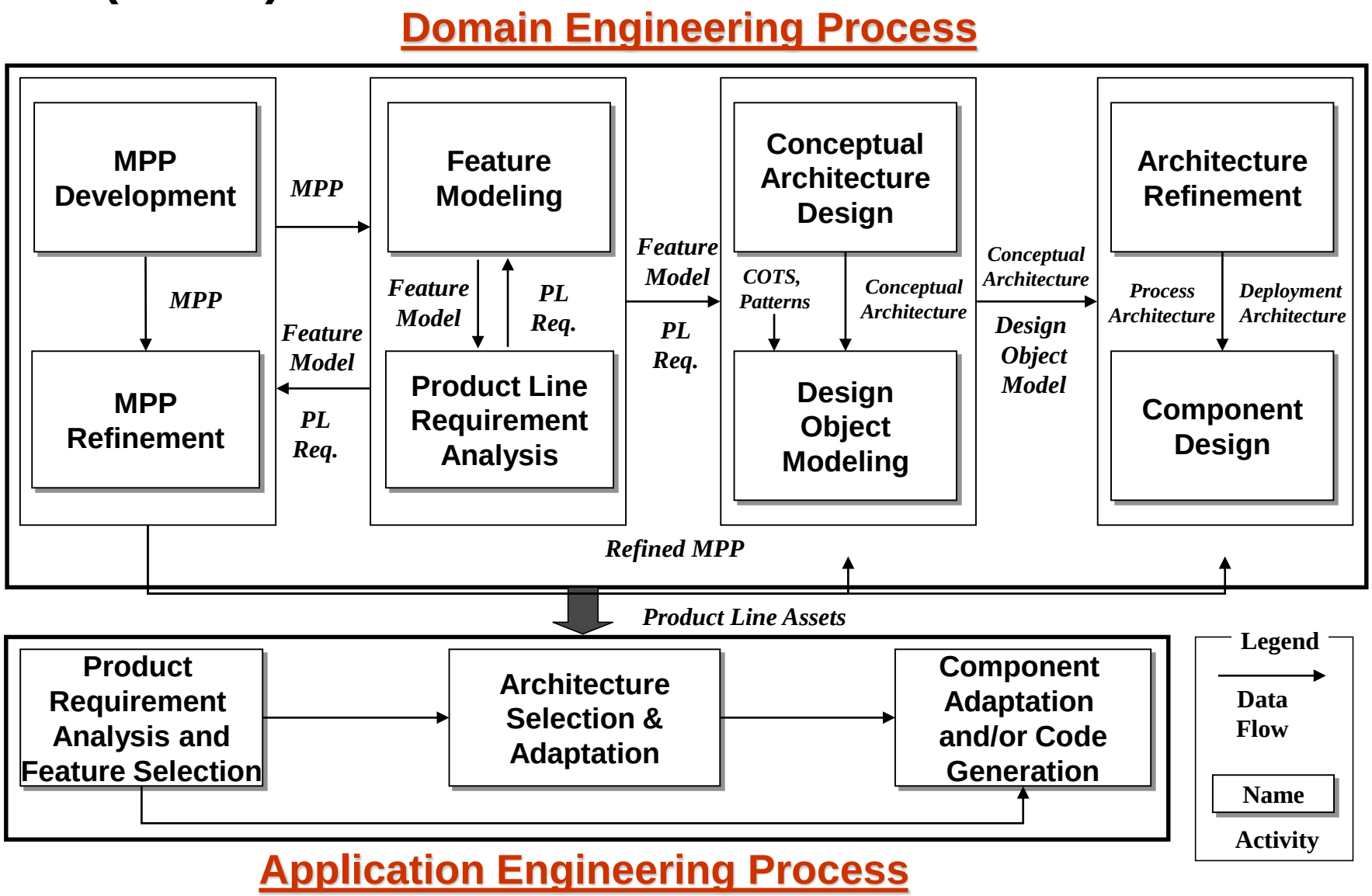
Product Line Approaches

*Charles Krueger, "Eliminating the Adoption Barrier", IEEE Software, Jul/Aug, 2002, pp. 29-31

- Extractive approach*
  - The extractive approach reuses one or more existing software products for the product line's initial baseline.
  - Require lightweight software product line technology and techniques that can reuse existing software without much reengineering.
  - Effective for an organization that wants to quickly transition from conventional to software product line engineering
  - This approach *<u>does not</u>* support the possibility of one organization developing the core assets and a separate organization developing the products based on the core assets.



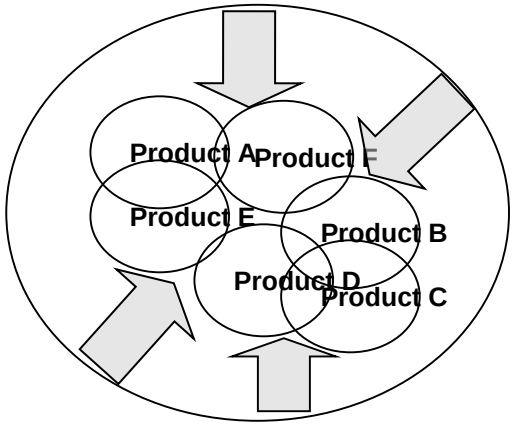
Product Line Engineering Processes: Feature-Oriented Reuse Method (FORM)



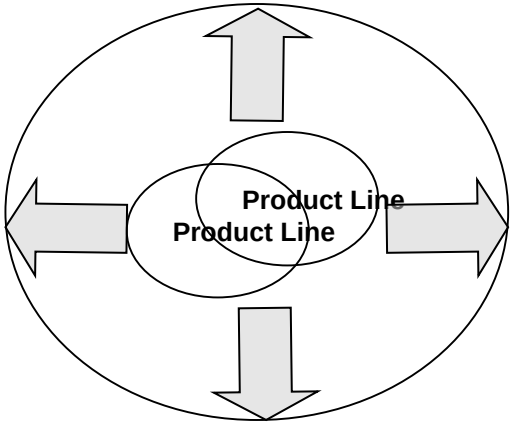
* MPP: Marketing and Product Plan * PL: Product Line * Req.: Requirements

Domain Engineering vs. Software Product Line Engineering

- They both attempt to exploit commonalities to build reusable core asset.
- SPLE is founded on a marketing and a product plan that specifies target products and their features from a market analysis is the primary input.
- The scope of analysis and development in SPLE is determined considering "time to market," "market evolution," and "technology evolution."

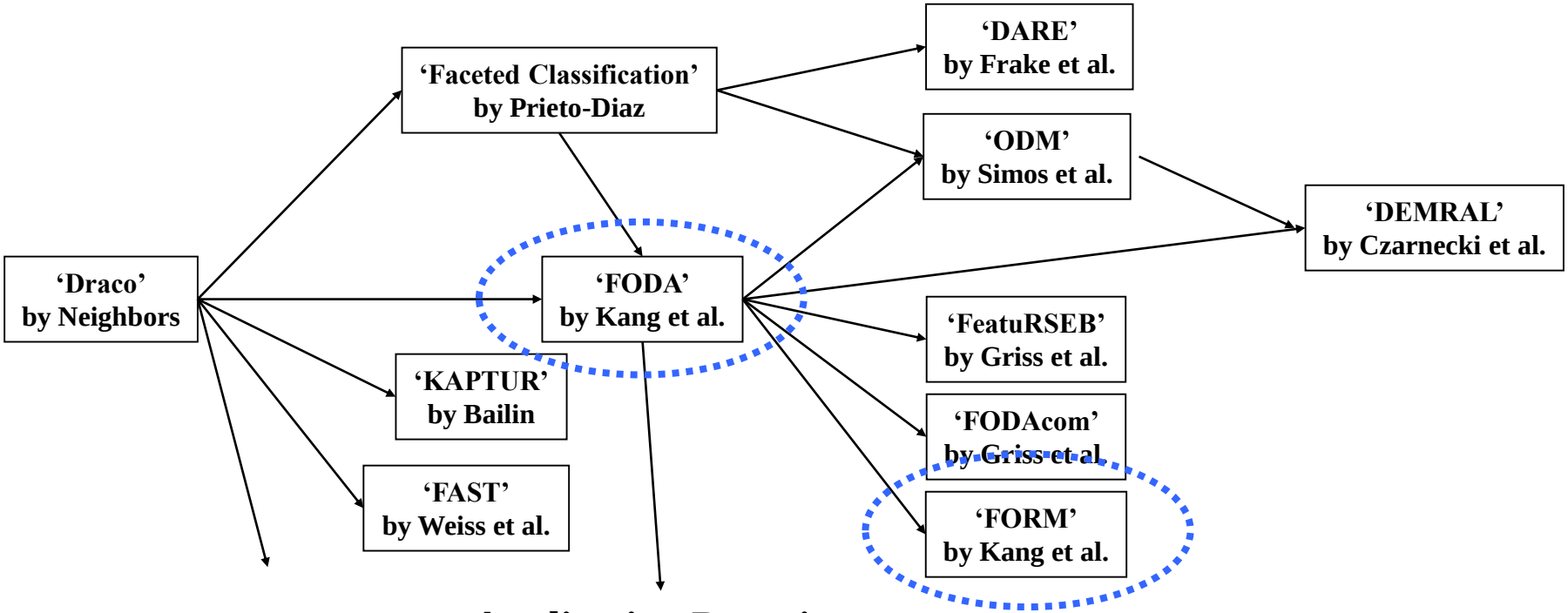


Domain Engineering Approach



Product Line Engineering Approach

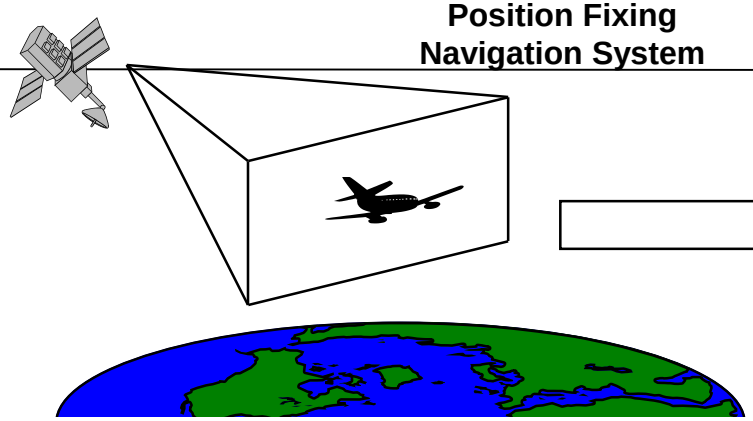
Domain Analysis Technology Evolution



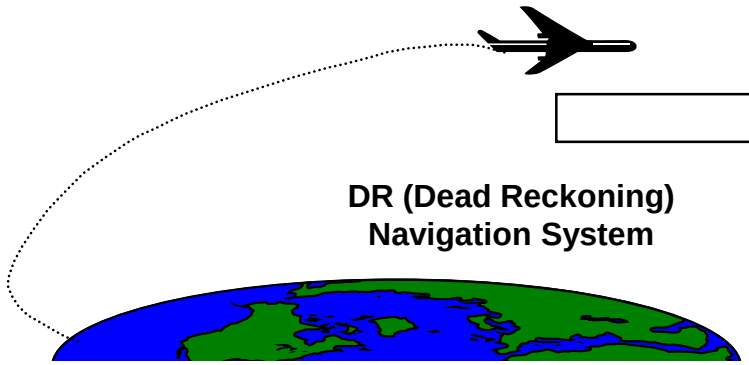
Application Domains

- The Army Movement Control Domain [Cohen et al., 1991]
- The Automated Prompt Response System Domain [Krut et al., 1996]
- The Telephony Domain [Vici et al., 1998]
- The Private Brach Exchange Systems Domain [Kang et al., 1999]
- The Car Periphery Supervision Domain [Hein et al., 2000]
- The Elevator Control Software Domain [Lee et al., 2000]
- The E-Commerce Agents Systems Domain [Griss, 2000]
- The Algorithmic Library Domain [Czarnecki et al., 2000]

Concept of Domain Language



Abstraction by Naming



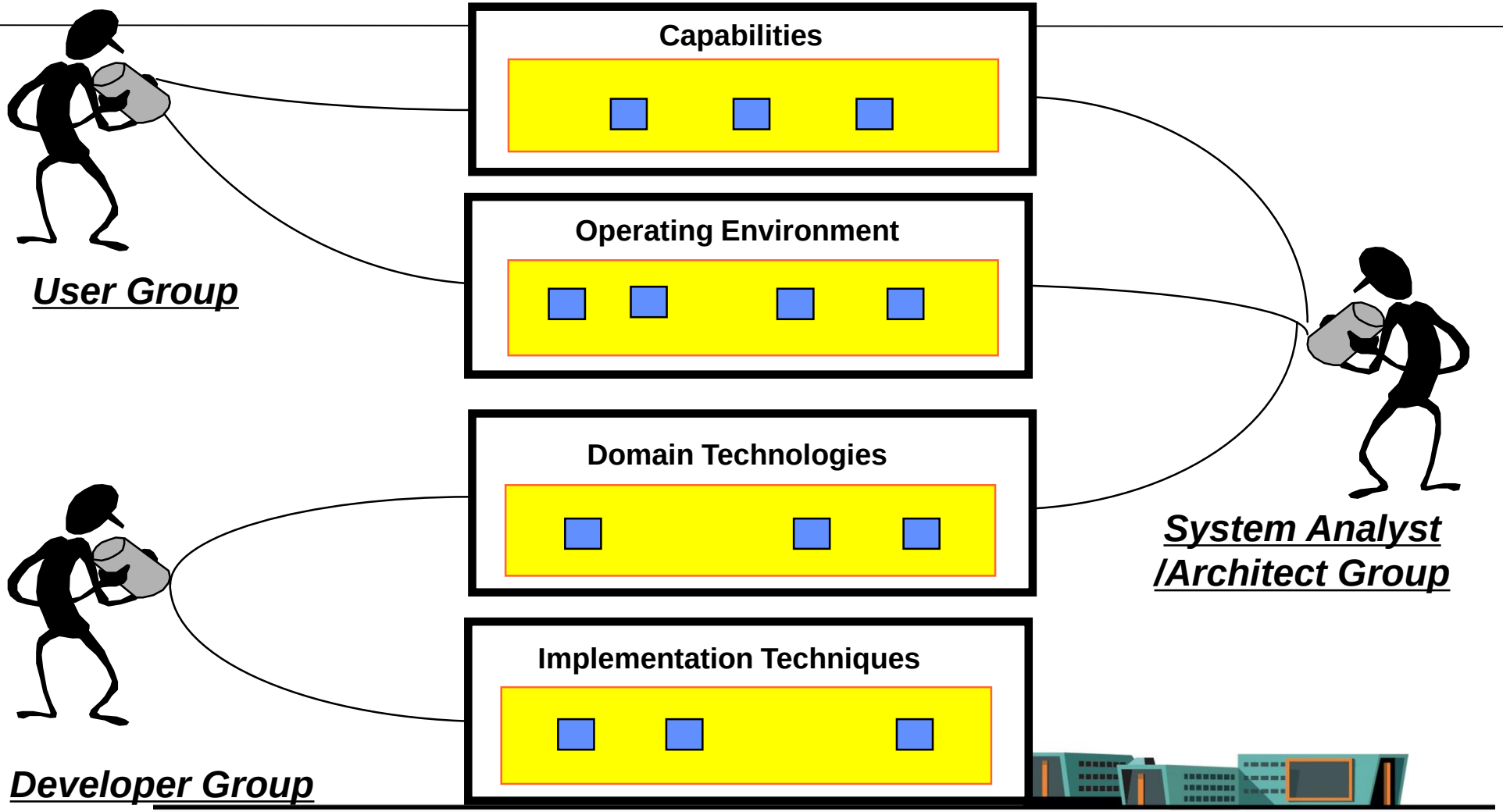
Navigation Method

- **Position Fixing Navigation System**
  - Range and bearing radio navigation aids
  - Hyperbolic radio navigation systems
  - Satellite Navigation systems - GPS (Global Position System)
  - Terrain reference navigation (TRN) systems
- **DR (Dead Reckoning) Navigation System**
  - Air data based DR navigation
  - Doppler/heading reference systems
  - Inertial navigation systems
  - Doppler/inertial navigation systems

Real World : Navigation System

Conceptual World : Domain Terminology

Identification of Features through Domain Language Analysis



Feature Model

What is Feature?

Various definitions of “feature”:

***- Features are "abstractions" of user or developer visible characteristics of an application domain* [FODA90].**

***- A feature refers to an attribute or characteristics of a system that is meaningful to, or directly affects, the users, developer, or other entity that interacts with a system* [NIST94].**

***- A feature is an essential “property” for its associated concept* [ODM98].**

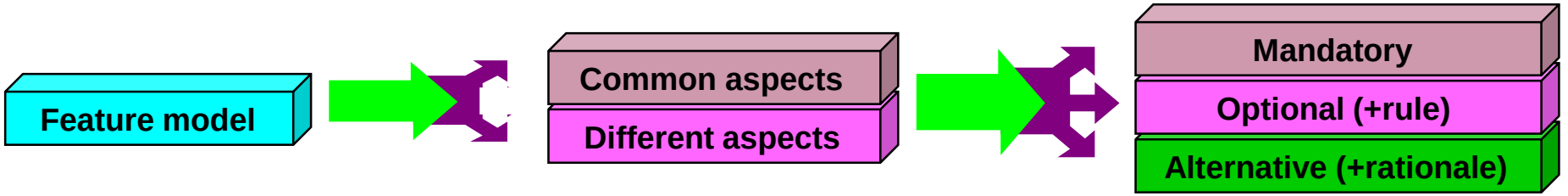
[FODA90] K. Kang, S. Cohen, J, Hess, W. Nowak, and S. Peterson, “Feature-Oriented Domain Analysis (FODA) Feasibility Study,” Technical Report, CMU/SEI-90-TR-21, Software engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, November 1990.

[NIST94] National Institute of Standard and Technology Special Publication 500-222, MD 20899-001, Gaithersburg, December 1994.

[ODM98] M. Simos and J. Anthony. “Weaving the Model Web: A Multi-Modeling Approach to Concepts and Features in Domain Engineering,” Proceedings of the Fifth International Conference on Software Reuse, IEEE Computer Society Press, 1998.

Overview of Feature and Feature Model

Feature : a prominent or distinctive user-visible aspects, quality, or characteristics of a S/W system or systems.



Capabilities — of applications from the end-user's perspective.

Operating Environment — in which applications are used and operated.
(H/W, S/W platform, O/S, interfaces with different types of devices)

Domain Technologies — that are commonly used in a domain (e.g., navigation methods in the avionics domain).

Implementation Techniques — designer's decision on algorithms and data structures.

References

- [FODA90] K. Kang, S. Cohen, J, Hess, W. Nowak, and S. Peterson, “Feature-Oriented Domain Analysis (FODA) Feasibility Study,” Technical Report, CMU/SEI-90-TR-21, Software engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, November 1990.

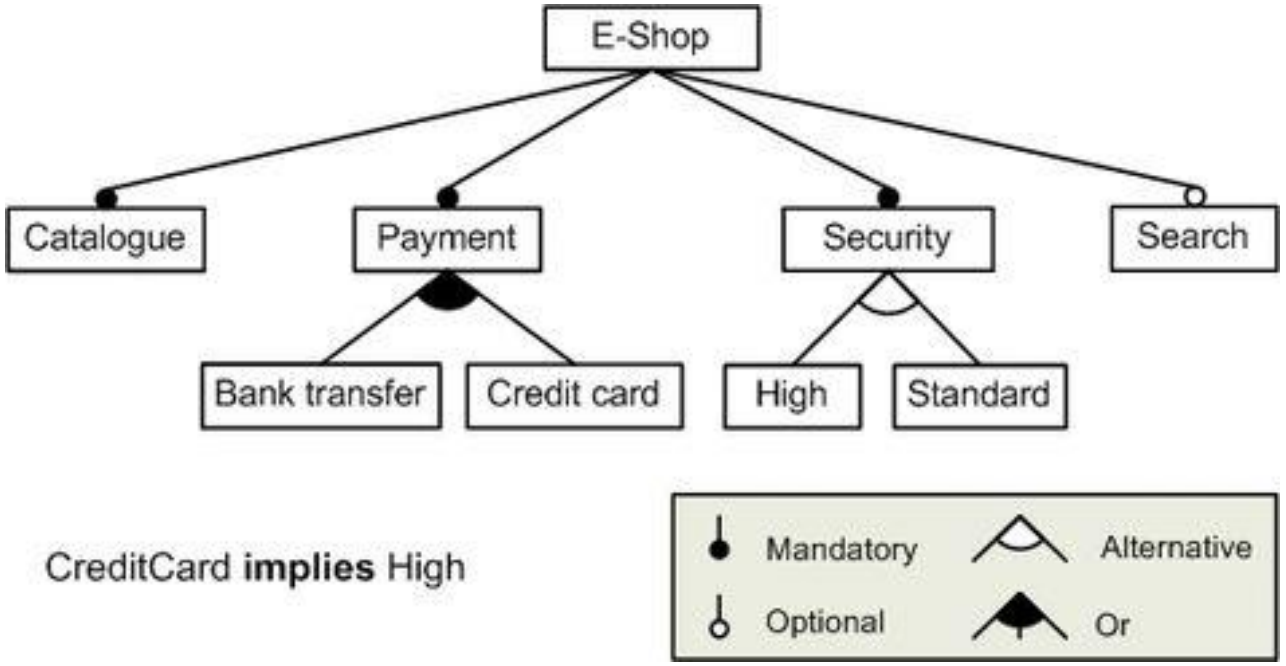
- [NIST94] National Institute of Standard and Technology Special Publication 500-222, MD 20899-001, Gaithersburg, December 1994.

- [ODM98] M. Simos and J. Anthony. “Weaving the Model Web: A Multi-Modeling Approach to Concepts and Features in Domain Engineering,” Proceedings of the Fifth International Conference on Software Reuse, IEEE Computer Society Press, 1998.

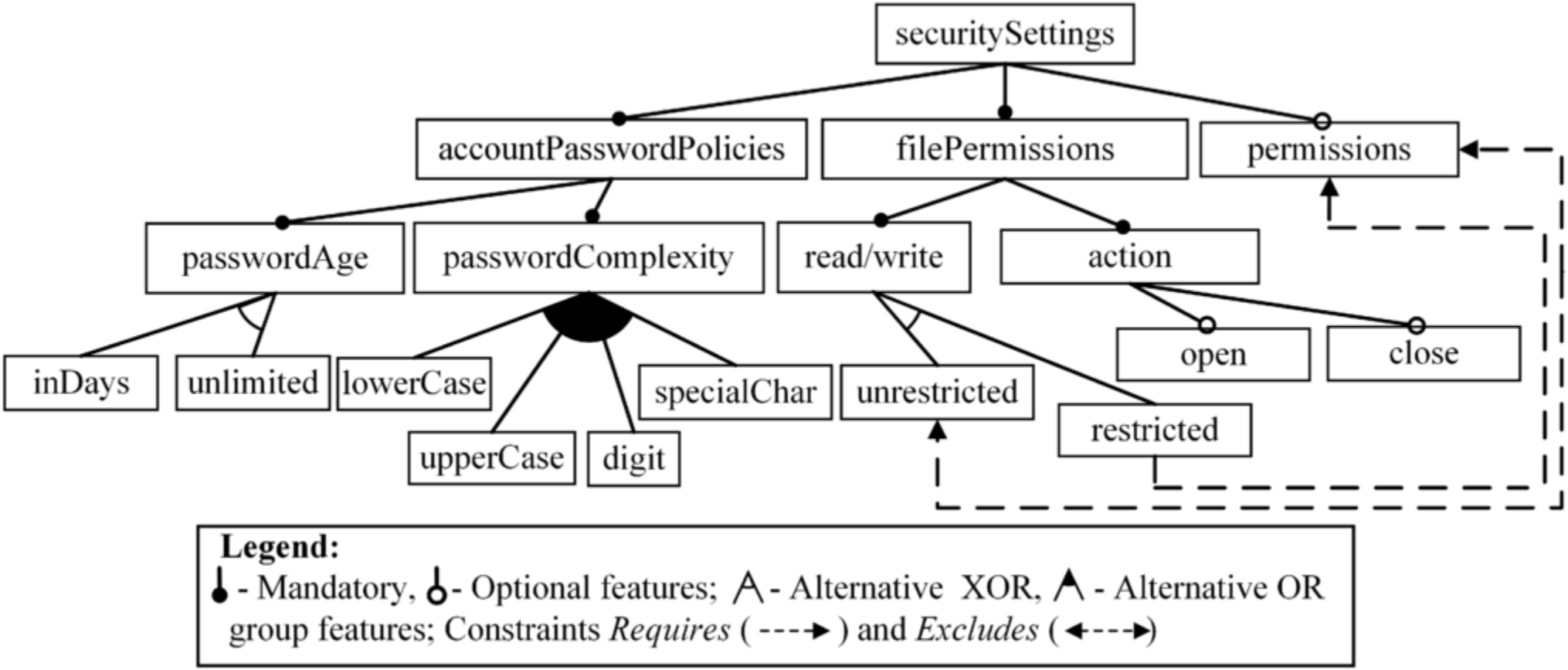
- K. Lee, K. Kang, and J. Lee, "Concepts and Guidelines of Feature Modeling for Product Line Software Engineering," Cristina Gacek, editor, Software Reuse: Methods, Techniques, and Tools: Proceedings of the Seventh Reuse Conference (ICSR7), Austin, U.S.A., Apr.15-19, 2002, Heidelberg, Germany: Springer Lecture Notes in Computer Science Vol. 2319, 2002, pp. 62-77.

- K. Kang, J. Lee, and P. Donohoe, "Feature-Oriented Product Line Engineering," IEEE Software, Vol. 9, No. 4, July/August 2002, pp. 58-65.

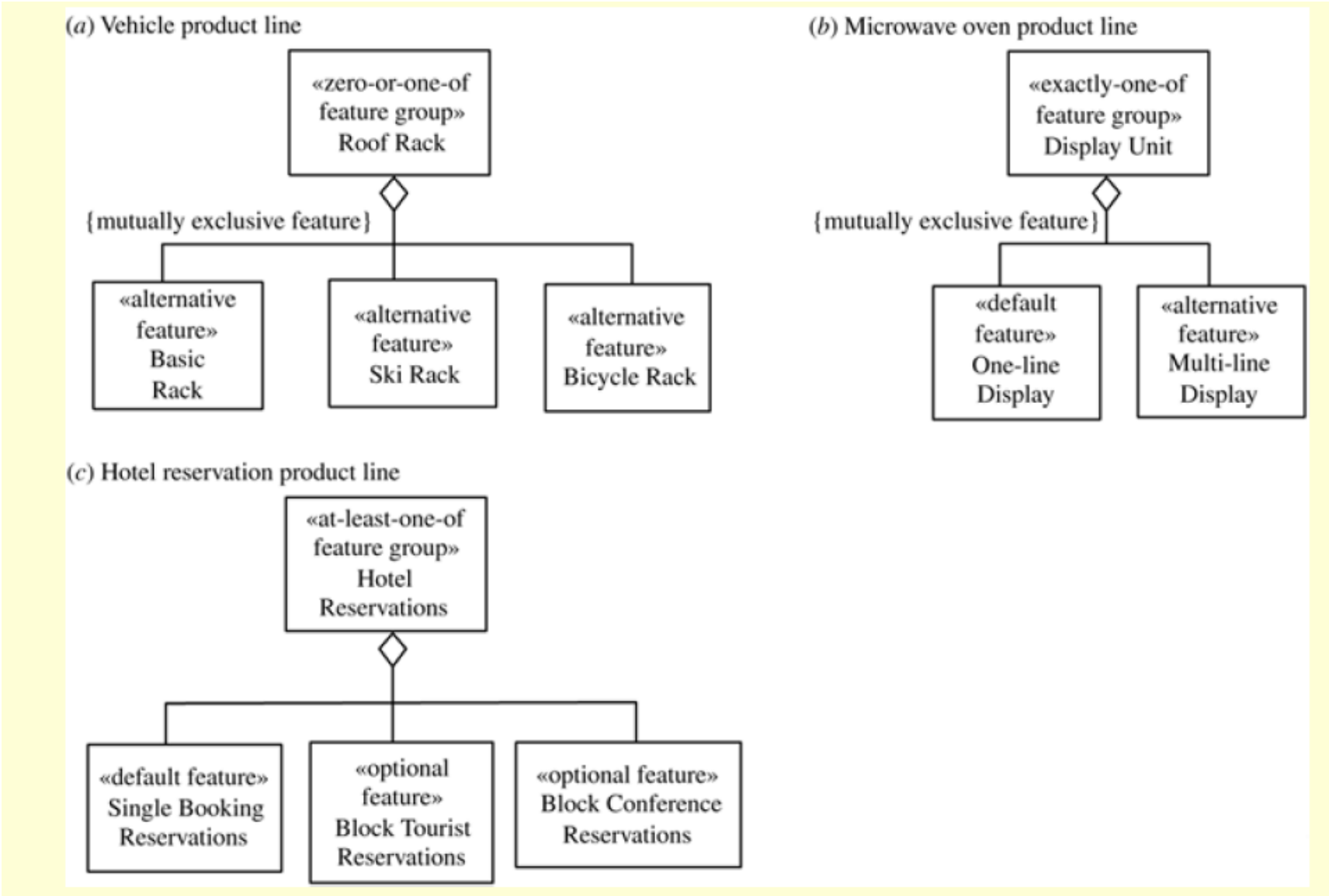
Feature Model Examples (Kang notation)



Feature Model Examples (Kang notation)



Feature Model Examples (UML Notation)



Feature Model Examples (UML Notation)

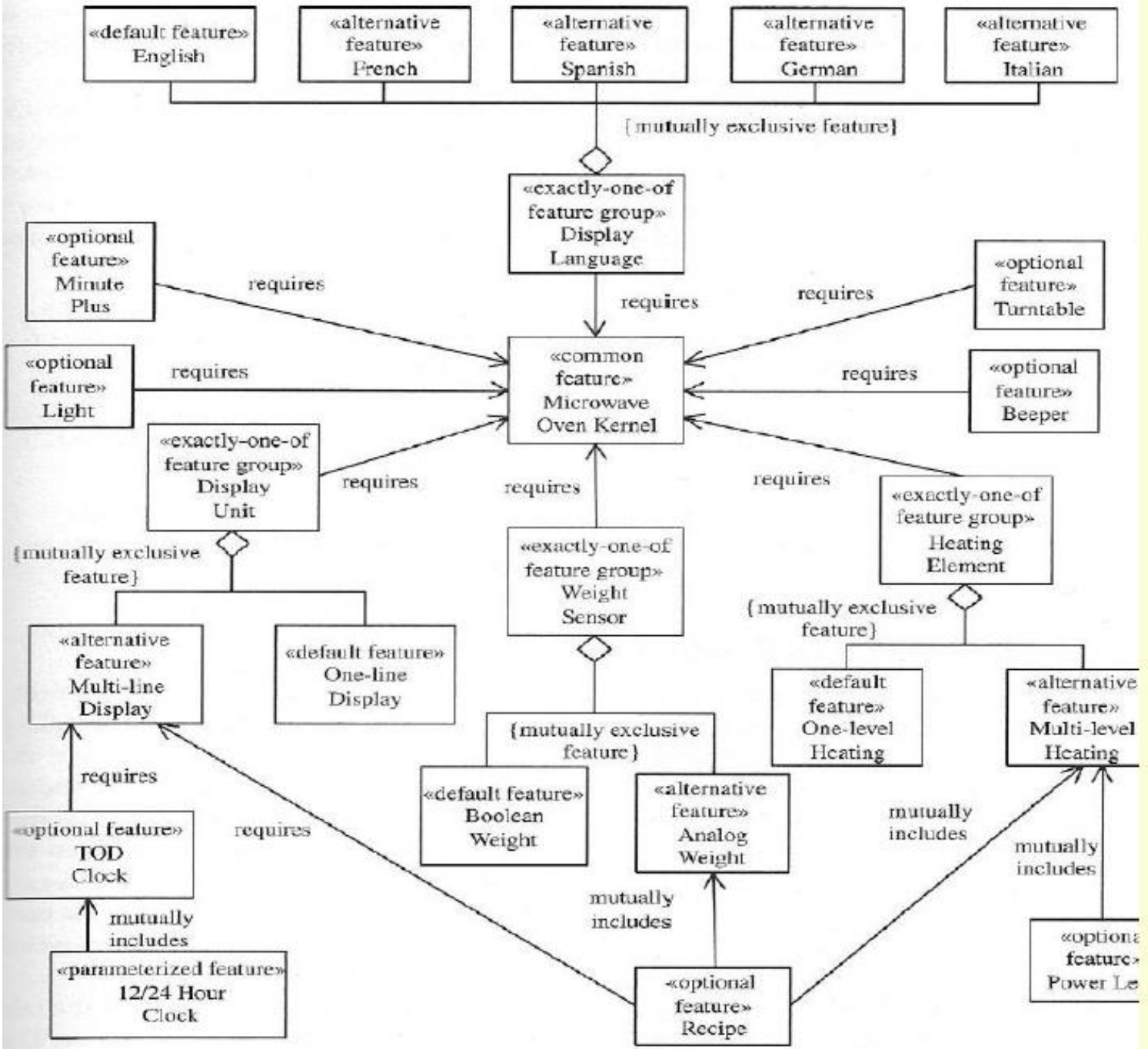


Figure 13.2 *Feature dependency diagram for the microwave oven software product line*