

Domain-specific Engineering

Introduction

Course Motivation

- **What:** Institute an effective strategy for systematic reuse
- **Why:**
 - Too few people with needed experience or expertise
 - Unreliable schedules or inconsistent product quality
 - Time to market too long
 - Effort being duplicated across projects
 - Products that are expensive to tailor or modify
- **Who:** Organizations that expect to repeatedly build similar products
- **How:** *the subject of this course*

Making Reuse Pay

Do the right thing

Focus reuse investments on how to rapidly build similar products

Do it right

Adopt reuse as part of a systematic process improvement initiative

Making Reuse Pay

Doing the right thing

Domain-specific Engineering (DsE)

Doing it right

Reuse-driven Process Improvement (PI_r)

Basic Reuse Tenet

The only sound basis for reuse is a *family*
(an envisioned set of *similar* products or components)

Similarity

- **Commonality:** the basis for standardization (of work products and process)
- **Variability:** the flexibility needed to accommodate different needs

Adaptability

- **An explicit representation of similarity**
- **A characteristic set of deferred decisions that distinguish among the members of a family**

Origins of the 'Family' Concept 1

Edsger W. Dijkstra, "On Program Families,"
in *Notes on Structured Programming*, 1972.

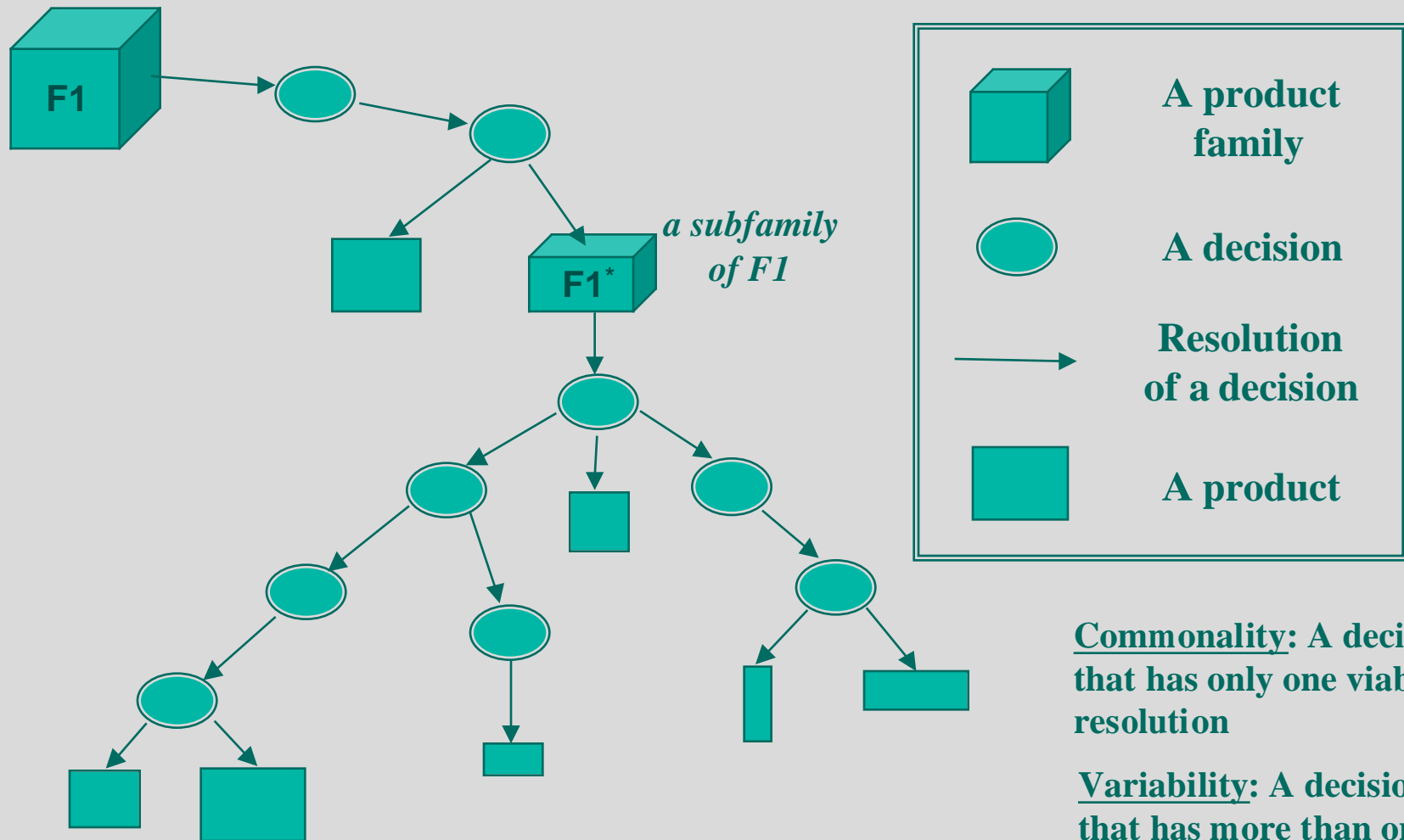
- I prefer to regard a program not so much as an isolated object, but rather as a member of a family of 'related programs'.
- We can think about related programs either as alternative programs for the same task or as similar programs for similar tasks.

Origins of the 'Family' Concept 2

David L. Parnas, "On the Design and Devel. of Program Families," in *IEEE Trans. on Software Eng.*, March 1976.

- A set of programs constitute a *family* whenever it is worthwhile to study programs from the set by *first* studying the common properties of the set and *then* determining the special properties of the individual family members.
- Software will *inevitably* exist in many versions. The differences between these versions are unavoidable and purposeful.
- If a designer/programmer pays conscious attention to the family rather than a sequence of individual programs, the overall cost of development and maintenance of the programs will be reduced.

Extracting Products from a Family



Commonality: A decision that has only one viable resolution

Variability: A decision that has more than one viable resolution

adapted from D. L. Parnas

A Simple Example of Decisions

Family: “collection” data structures

Decision 1: entry type? *Integer*

Decision 2: specific values can be repeated? *Yes*

Decision 2.1: order of stored values (byValue, asInserted)?
asInserted

Decision 2.1.1 a: position for insertions (first/last/indexed)?
last

Decision 2.1.1 b: position for removals? *first*

Subfamily: “queue” data structures

Decision 2.1.1.1: maximum size? *Unlimited*

Instance: an unlimited integer queue program

The Role of Decisions

- **Engineering is a decision-making process**
- **A family is the potential that different ways to resolve a set of decisions leads to different programs**
- **Decisions derive from:**
 - **Customer needs (requirements and constraints)**
 - **Engineering tradeoffs (subjective factors such as cost, quality, or esthetics)**
- **A domain makes a family tractable, reducing many possible variabilities to commonalities by precluding economically and technically non-viable alternatives**

Course Topics

Overview

(concepts and benefits)

Application Engineering

(using a domain to deliver products)

Domain Engineering

(creating a domain)

Reuse-driven Process Improvement

(adopting and instituting DsE)

Detailed Outline

Half day 1

Introduction

Overview

Domain-specific Engineering

Reuse-driven Process

Improvement

Half day 2

Application Engineering

An augmented process

A streamlined process

Domain Engineering 1

Domain Management

Domain Definition

Half day 3

Domain Engineering 2

Product Family

Engineering

Process Engineering

Project Support

Half day 4

Reuse-driven Process

Improvement

Evaluating viability

Adopting DsE

Wrap up

General Terminology (1)

- **Methodology** - An integrated body of principles, practices, and methods that prescribe the proper performance of a process
- **Method** - Guidance and criteria that prescribe a systematic, repeatable technique for performing an activity
- **Plan** - A designation of tasks and resource allocations for accomplishing a specified objective (an instance of a process)
- **Task** - A managed work assignment to accomplish a specified objective (an instance of an activity)

General Terminology (2)

- **System** - A collection of hardware, software, people, and other systems that operate together to perform a mission
- **Application** - Hardware, software, and manual procedures that an organization employs to enact a system
- **Process** - A (partially) ordered set of steps, intended to accomplish specified objectives
- **Activity** - A step of a process for producing or evaluating work products to satisfy objectives supporting that process and comprised of other steps
- **Product** - An application and all associated work products (resulting from an AE project)
- **Work product** - Any tangible artifact resulting from an activity