Prosperity Heights Software

Code Generation 2009

Envisioning an Environment for Domain-specific Engineering

Grady Campbell

17 June 2009

< gradycampbell@domain-specific.com >

Agenda

Establishing Context

- Product lines (PL) and Domain-specific Engineering (DsE)
- Initial participant views on PL tool and environment options and needs

Notions on Environments for Domain-specific Engineering

- Origins and concepts
- Aspects of an experimental DsE environment

Discussion

- Insights from preceding comments
- Notional outline of DsE environment capabilities

Agenda

Establishing Context

- Product lines (PL) and Domain-specific Engineering (DsE)
- Initial participant views on PL tool and environment options and needs

Notions on Environments for Domain-specific Engineering

- Origins and concepts
- Aspects of an experimental DsE environment

Discussion

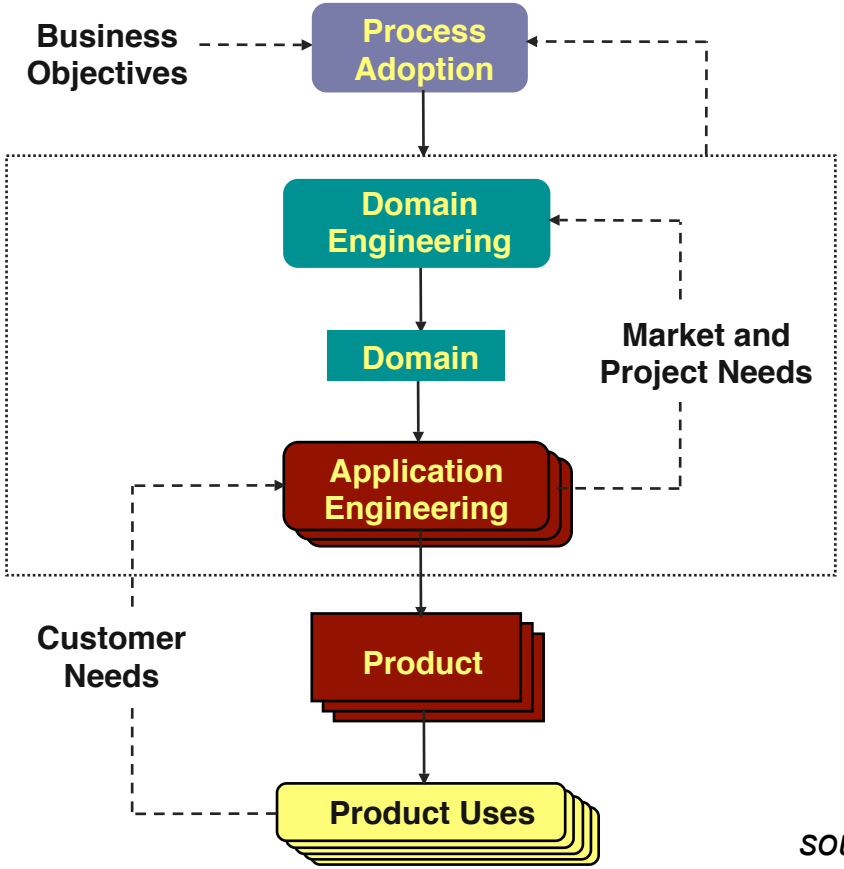
- Insights from preceding comments
- Notional outline of DsE environment capabilities

The Product Line Vision

When a market exists for a set of similar products, build and use a product family manufacturing capability that eliminates redundant effort:

- Create a capability for rapidly building similar customized products
- Limit diversity and effort by focusing on similarity of customer needs in the targeted market
- Rationalize and focus the customer-producer dialog to rapidly converge on a describable problem-solution
- Predictably build complete products with known properties
- Provide a controlled environment for evaluating alternative solutions and tradeoffs
- Continuously improve the product family as market needs and technology evolve

Domain-specific Engineering (DsE): A Product Line Process



Institute & improve a product line business

Develop and evolve a capability for building similar products

Build customized products for customers

source: **www.domain-specific.com** *(1999)*

Definitions

Domain - The knowledge (product family) and expertise (process) required to build a particular type of product

Model - A representation of a [product] that is sufficient to provide approximate answers to a designated set of questions about the represented [product]

Abstraction - A concept that denotes criteria for membership in a set (i.e., the characteristic function for a subset)

Family - (1) A set consisting of all instances that satisfy an associated defining abstraction (2) In mathematics, a set of functions that can be generated by varying the parameters of a general form

Product Family - A set of products that provide similar solutions to an envisioned set of similar problems

Product Line - A set of products having similar capabilities to address differing needs of customers in an organization's targeted market

Why Domain-specific

Domain experience and expertise is the single most important discriminator of software competence

All software businesses have an inherently limited domain-specific competence - even if they don't know what it is

Businesses with a market focus have an advantage over unfocused businesses

Developers are more effective if they are experts in how domain products work and why

A domain focus enables use of narrower, more powerful techniques

A domain focus bounds the software capabilities that need to be built

What is a Domain-specific Description?

There is an intuitively-bounded domain of discourse represented by a standardized terminology that encapsulates great complexity.

Don't think software - think building/buying a house

Buyer says: "I want a Tudor with space for 2 adults and 1 child"

No need to say: "I'd like it to have doors, windows, electricity, and indoor plumbing" - these are assumed (commonalities) but may have associated variabilities (constrained by other choices)

Builder says: "Here are 3 plans and material samples, which do you like?" You pick and he says: "This will take 3 months to build and cost $500K."

There are further details to decide and tradeoffs to make among those. ("Too expensive? You chose very expensive applicances; how about switching to one of these less expensive brands?")
etc. etc.

Agenda

Establishing Context

- Product lines (PL) and Domain-specific Engineering (DsE)
- Initial participant views on PL tool and environment options and needs

Notions on Environments for Domain-specific Engineering

- Origins and concepts
- Aspects of an experimental DsE environment

Discussion

- Insights from preceding comments
- Notional outline of DsE environment capabilities

Agenda

Establishing Context

- Product lines (PL) and Domain-specific Engineering (DsE)
- Initial participant views on PL tool and environment options and needs

Notions on Environments for Domain-specific Engineering

- Origins and concepts
- Aspects of an experimental DsE environment

Discussion

- Insights from preceding comments
- Notional outline of DsE environment capabilities

Origins of DsE

NRL SCR: disciplined engineering methods [1981]

- Semi-formal requirements notation
- Architecture-based design and module interface specifications

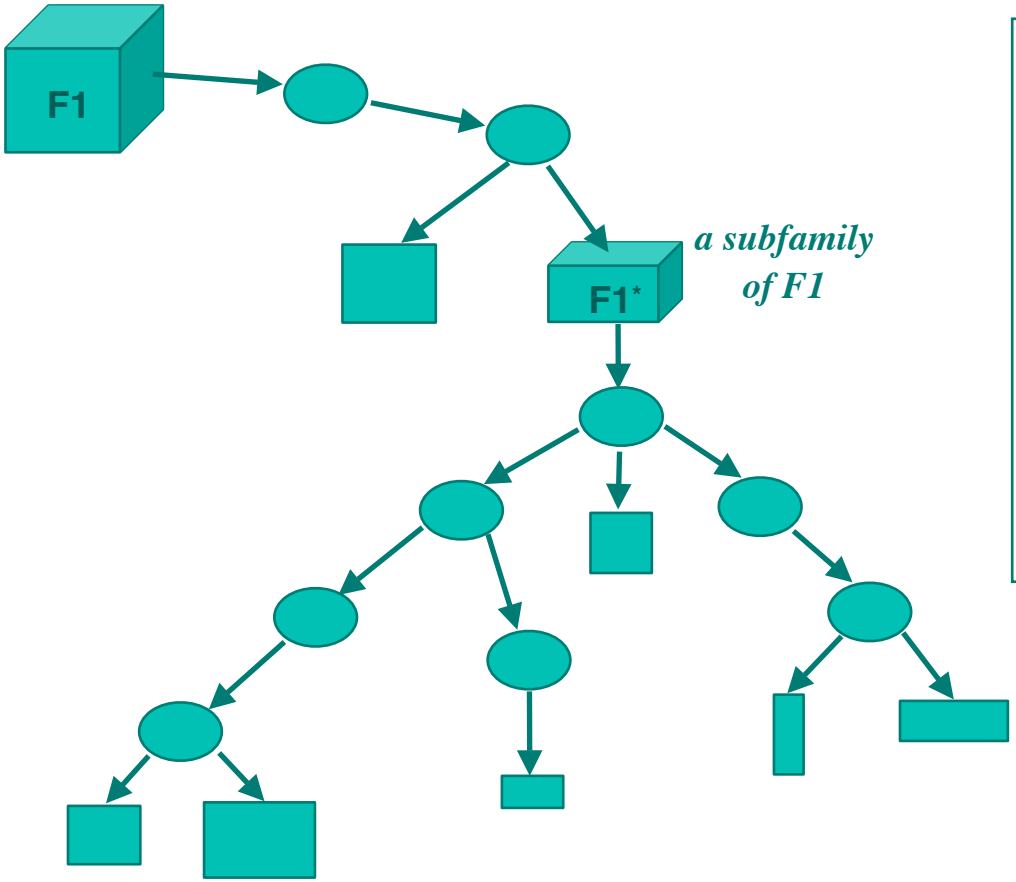
Spectrum: decision-based application generation [1984]

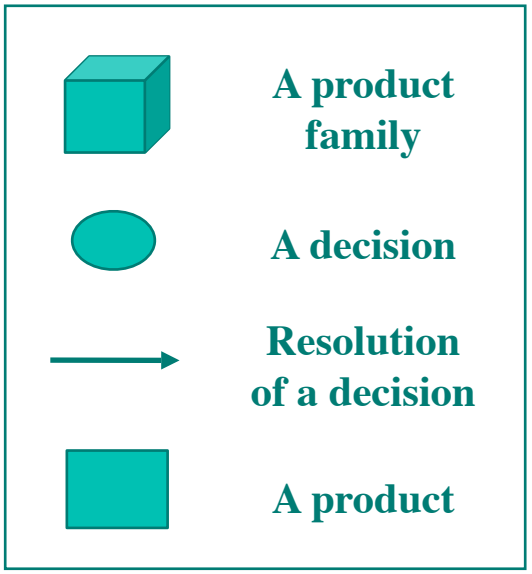
- Metaprogramming method and tool => text-based adaptable components
- Decision-based application model specification
- Application-model-based product generation with adaptable components

Synthesis / RSP: a systematic product line approach [1990]

- Domain-specific focus on a product line market
- Dual product family and process engineering
- A method for methodology tailoring/adoption/improvement

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

The Role of Decisions

Engineering is a decision-making process (different decisions result in a different product).

A product family shows how different ways of resolving a set of decisions lead to different products.

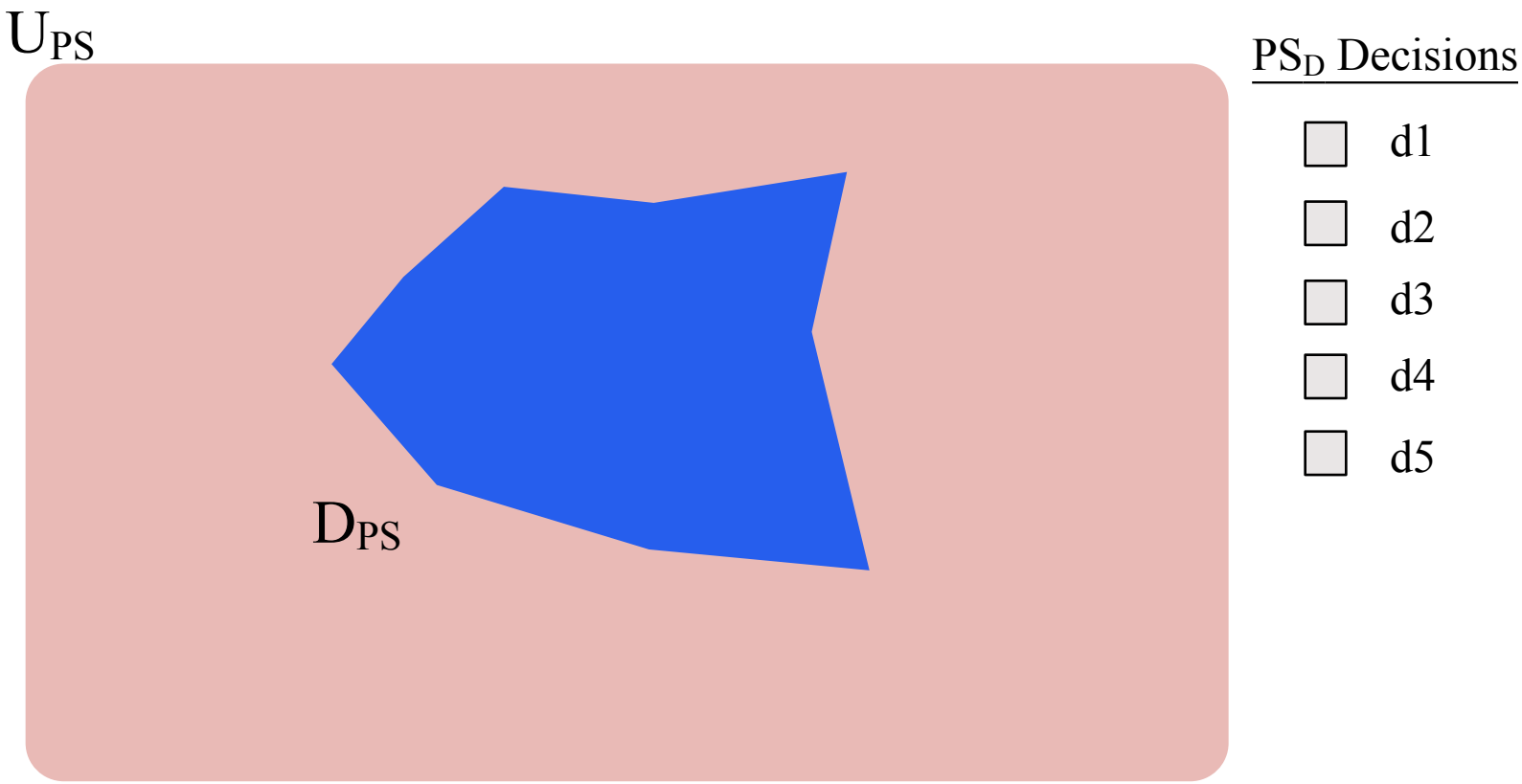
Decisions represent:

- The differences in what customers ask for (needs and constraints)
- Unresolved engineering tradeoffs

A focus on similar problems (a family) enables standardization, reducing number, variety, and complexity of decisions.

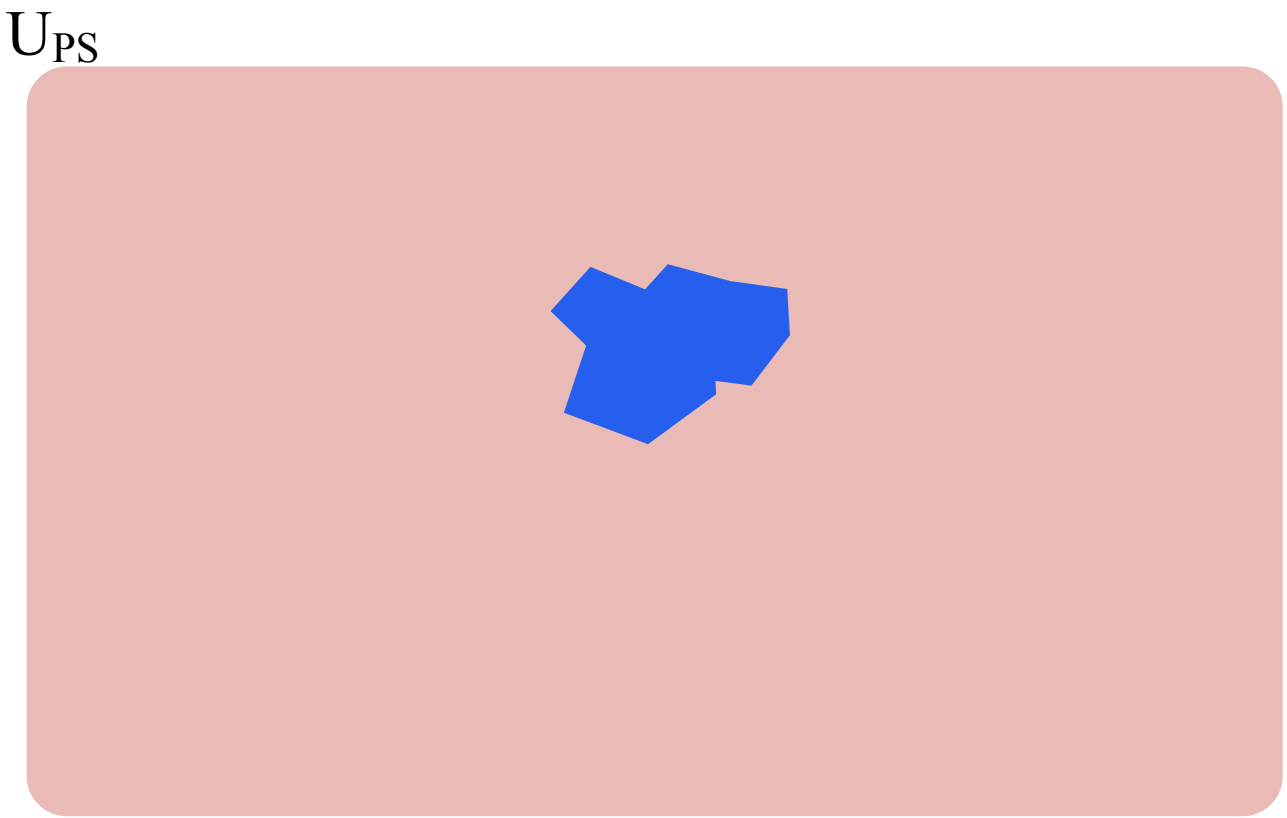
A “decision model” enables condensing the customer-developer dialog to its essentials, only what is needed to distinguish a particular product from others in a domain

Effect of Decisions on a Problem-Solution Space



(U_{PS} : universal problem-solution space | D_{PS} : domain-specific problem-solution space)

Effect of Decisions on a Problem-Solution Space



Resolving decisions reduces the applicable problem-solution space.

Spectrum Application Specification Concepts

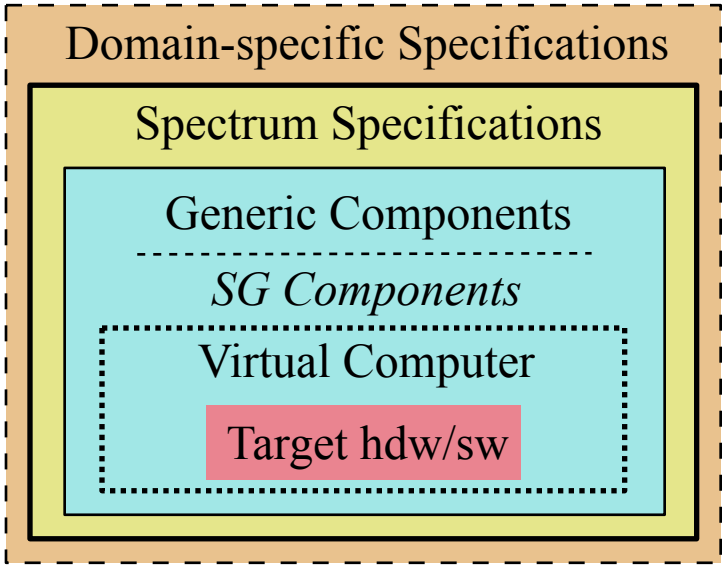
World Model

- Data model (persistent object classes with weak multiple inheritance and typed data and relationship attributes)
- Packages (procedural computation)
- Strategies (active values with associated knowledge sources and dependencies: Class, Attribute, Autonomous)

External Interface

- Devices (Standard Hardware/Users, External Software, Non-standard Hardware)
- Output Manager
  - Output Generator (nestable object-oriented, relationship-oriented, text/ document, and iconic/graphic information displays)
  - Output Control (Sequencing, Selection, Concurrent, Subordinate)

Spectrum Levels of Definition



Generic (Adaptable) Components

```
Virtual Devices: PRT CRT STR DBD NET
Data Abstractions:  OBJ TYP RDB
Logic Abstractions:  KRC STC
User Interface:  FRM EDF SEF CTL IAC

SG Components: TRF CMP CFG
```

Distinguishing Features of Synthesis/RSP and DsE

A framework for solving the problem of software productivity and product quality

Domain-specific: A focus on a domain represented as a family of products, all being similar but differing in well-defined ways

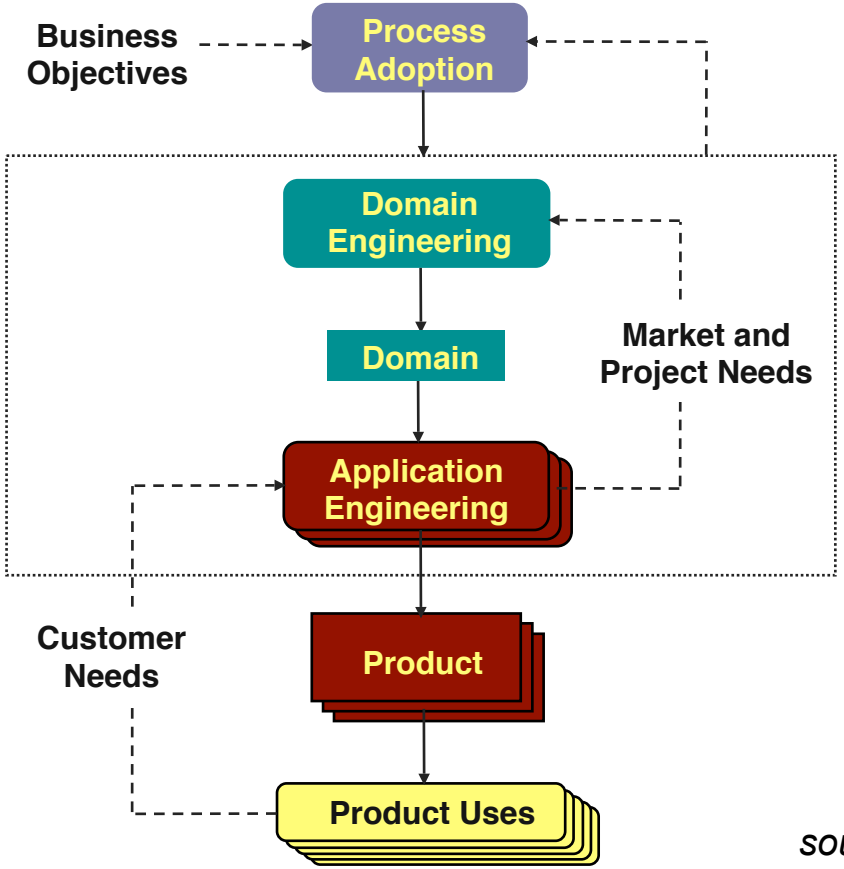
Streamlined production: Product building reduced to the resolution of decisions that correspond entirely to the ways in which customers' needs can differ

Adaptable assets: The mechanical derivation of customized work products based on domain-specific decisions

Model-based analyses: The use of model-based analyses for help in understanding which decision choices provide the best product

***source*: Reuse-driven Software Processes Guidebook (1993)**
< www.domain-specific.com/RSPgb >

Domain-specific Engineering (DsE): A Product Line Process



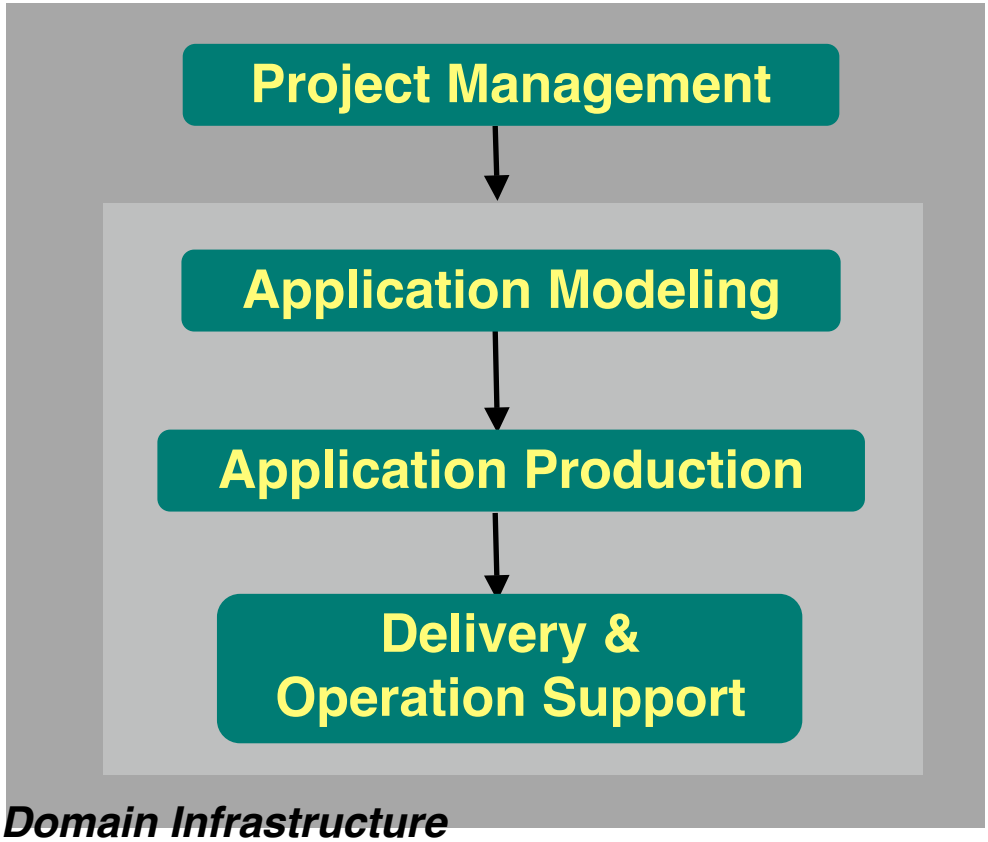
Institute & improve a product line business

Develop and evolve a capability for building similar products

Build customized products for customers

source: **www.domain-specific.com** *(1999)*

A DsE Application Engineering Process

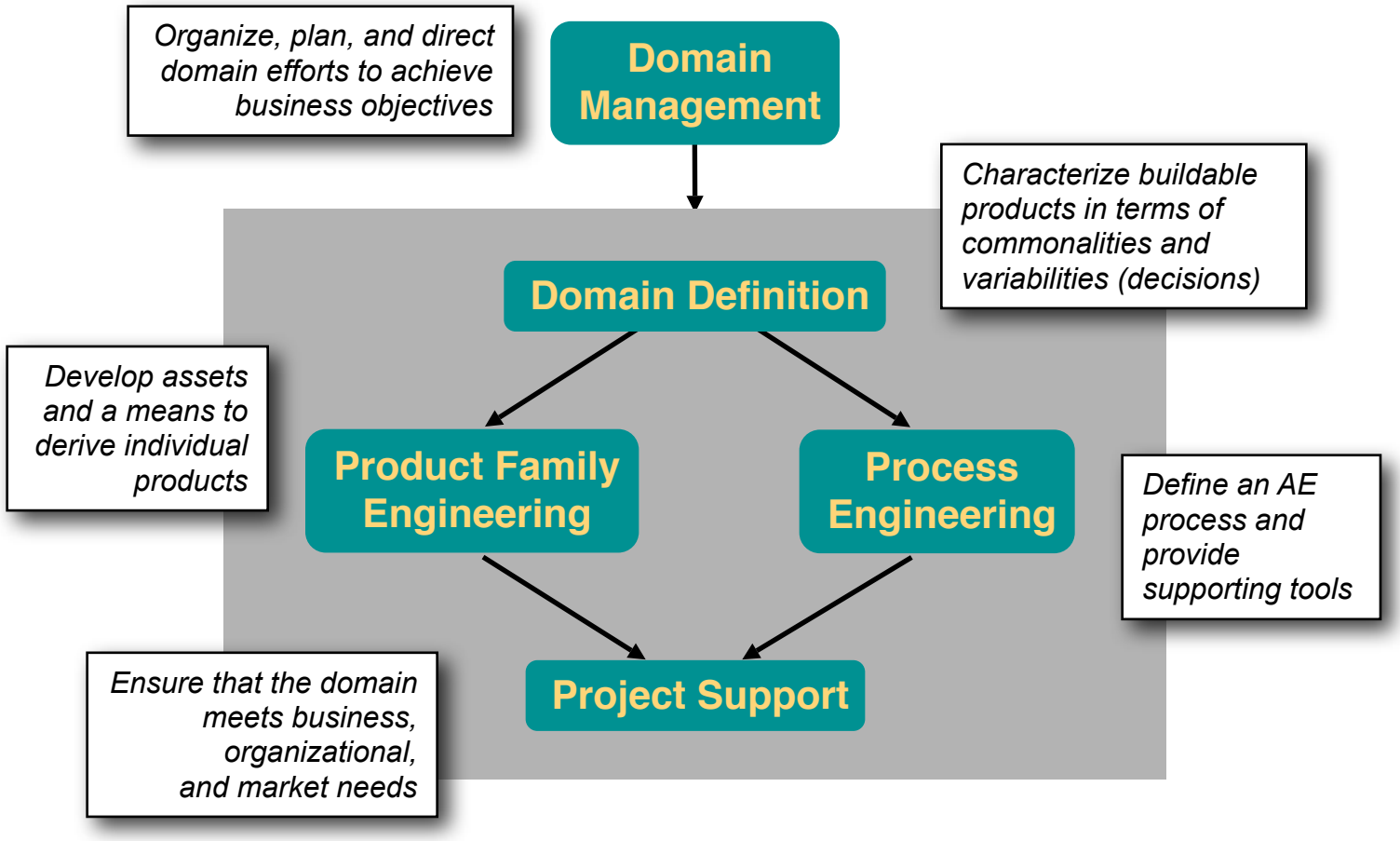


Product Specification & Evaluation

Product Generation & Evaluation

Product Distribution

A DsE Domain Engineering Process



Agenda

Establishing Context

- Product lines (PL) and Domain-specific Engineering (DsE)
- Initial participant views on PL tool and environment options and needs

Notions on Environments for Domain-specific Engineering

- Origins and concepts
- Aspects of an experimental DsE environment

Discussion

- Insights from preceding comments
- Notional outline of DsE environment capabilities

DsE Environment Capabilities

Build instances of a targeted family of products (and, being domain-specific, only those):

- Iteratively specify a problem/solution and operational environment/ scenarios
- Derive static and dynamic analyses of product behavior and properties
- Generate a product as a whole (all work products) at any time
  - Specifications and documentation
  - Source/object code and installation mechanisms
  - Testing materials and infrastructure

Options with incomplete problem/solution specifications (open decisions):

- Generate a product subfamily (alternate solutions) for empirical evaluation
- Generate code that lets users resolve decisions as needed at runtime

Reasons for an Experimental DsE Environment

Conceive and explore representations and mechanisms for constructing a product family and deriving instance products

- Develop environment implementation capabilities to reduce DE development effort for a customized AE process
- Enable representing multiple product variants within a tool
- Apply text-based adaptability to existing textual work products (e.g., code components)
- Conceive and experiment with adaptability notations for non-text constructs
- Apply adaptability uniformly across a whole product (a collection of logically consistent work products)
- Propagate adaptability through a hierarchically structured representation (e.g., composite documents)

Current Emphasis

Domain engineering

- Define a domain (synopsis, glossary, commonality-variability assumptions and associated decisions)
- Construct a graphical formulation of a domain product family, associating decisions for adaptability at all levels
- Develop representations of interrelated, composite work products (graphical and textual forms, specifications, code, and documentation, etc.)

Application engineering

- Resolve domain-associated decisions to derive and present alternative customized whole-products

(Illustrate Potential Capabilities with the Experimental Tool)

Agenda

Establishing Context

- Product lines (PL) and Domain-specific Engineering (DsE)
- Initial participant views on PL tool and environment options and needs

Notions on Environments for Domain-specific Engineering

- Origins and concepts
- Aspects of an experimental DsE environment

Discussion

- Insights from preceding comments
- Notional outline of DsE environment capabilities

Agenda

Establishing Context

- Product lines (PL) and Domain-specific Engineering (DsE)
- Initial participant views on PL tool and environment options and needs

Notions on Environments for Domain-specific Engineering

- Origins and concepts
- Aspects of an experimental DsE environment

Discussion

- Insights from preceding comments
- Notional outline of DsE environment capabilities

Capabilities Needed in a DsE Environment

Perform all activities of the domain's application engineering process (including all conventional aspects of an environment)

Formulate a problem-solution as a resolution of associated decisions

Mechanically derive whole-products (all needed types of work-product)

Manage and comparatively evaluate alternative problem-solution variants

Integrate and use tools (models) with specialized representational/ analytic capabilities, adapted to accommodate multiple product variants

Maintain complex interdependencies among work products' content

Provide a platform for modeling operational environments and dynamically evaluating problem-solution variants within those contexts

Support project/process/product management/administrative activities