5. GSEP–Based Engineering for Product Families

This section discusses a GSEP–based approach to systems engineering for product families. A **product family** is a formalization of a product line, which is a collection of similar existing and potential systems that address a designated business area market. Systems engineering for a product family comprises creating and maintaining both a product family and a capability to produce and deliver specific members of the product family to customers. Section 2.4 describes the concept of product families in more detail.

The Consortium approach to reuse–driven software engineering, known as the Synthesis methodology, is to construct each system as an instance of a family of systems that have similar descriptions (Campbell, Faulk, and Weiss 1990). The *Reuse–Driven Software Processes Guidebook* (Software Productivity Consortium 1993a) describes the concepts and practice of software engineering to create and use product families. This section describes the extension of Synthesis to GSEP–based systems engineering for product families.

5.1 Rationale for Engineering with Product Families

An organization's motivation for developing a product family is that it recognizes a viable market for a series of similar systems. It must have both sufficient expertise to develop other systems of the same type, generally based on prior experience in developing or maintaining such systems, and a business commitment to do so. A customer needing a critical system that is expected to evolve to meet changing needs over many years suffices as such a market.

A conventional approach to systems engineering emphasizes one–of–a–kind handcrafting of each system. Although previous systems' components are sometimes used in developing a system, such components are seldom designed with sufficient attention to the needs of future systems and, therefore, often require ad hoc modification to meet those needs. Such an approach is inefficient for developing a series of similar systems. In comparison, a capital investment in the creation of a **domain**, comprising a product family and an associated capability to produce deliverable systems, can produce substantial long–term benefit.

A domain standardizes an organization's understanding of a set of similar problems and their corresponding system solutions. The essence of engineering for a product family is identifying and exploiting similarities in the systems that an organization builds to avoid redundant work. Each system in a domain may be large and complex, such as a space station; moderate, such as an automobile; or small, such as the engine of an automobile. Similarly, a domain may be large, encompassing substantial variety in the included systems, or small, encompassing only a small number of alternative systems having limited variety.

Using a domain to create deliverable systems can yield the following benefits:

- Reduced time and cost to develop a system (including the amortized cost of creating the domain)
- Higher quality systems, with less effort to correct and reverify flaws detected in verifying work products, through repeated flexible reuse of standardized parts
- More emphasis on understanding customer needs specific to each system, expressed in a standardized form, and a capability to identify and resolve unclear or incomplete requirements based on an explicit set of supported product variabilities
- Less dependence on direct, dedicated participation in each project by key engineers whose knowledge and expertise in problems and their solutions make a domain feasible
- An ability to rapidly produce alternative systems (or, equivalently, system models) for evaluating tradeoffs in satisfying customer needs
- Increased standardization, across a product line, of work products which are consistent and integrated by design rather than by brute force
- Improved communication and coordination among engineers based on a shared understanding of how systems to be produced are alike and how they can differ
- A sounder maintenance capability that ensures the continuing consistency and integrity of all work products of a

system

Better support for bid–and–proposal efforts based on a more accurate understanding of the organization's existing capabilities to deliver a particular type of system

Engineering for product families is most viable in an organization which emphasizes long–term business objectives. There must be a perceived market for a family of similar products, such as multiple customers with differing needs, a single customer who needs multiple versions of a system, or a long–lived system that is likely to undergo significant changes in customer needs or technology. Although creating a product family can take an investment of time and money that is greater than that required to produce a single system, it will reduce development and maintenance costs across an entire product line.

5.2 A Process to Create and Use Product Families

The process of engineering for a product family must be designed to support two independent but interrelated objectives:

To produce and deliver systems

To increase the effectiveness (profitability, productivity, product quality, manageability, and responsiveness) of system production and delivery

To address these separate concerns most effectively, the process consists of two integrated subprocesses: **application engineering** and **domain engineering**. Figure shows how these processes relate to each other. Application engineering (or Develop Application System) is the process followed to produce and deliver systems whereas domain engineering (or Develop Domain System) is the process followed to increase the effectiveness of system production and delivery. Derived from Figure 11 which depicts the independent development of several systems by an organization, this figure shows how domain engineering supports application engineering by supplying projects with a domain consisting of a product family and an associated system engineering process and environment tailored to the projects' needs. (The Above, Across, and Below relationships between any system and its containing, peer, and contained systems, subsystems, and components is omitted only to simplify this figure. The discussion concerning figure 11 in section 3.2 provides more on this.) An important aspect of the relationship which is not shown in the figure is feedback from application engineering projects to domain engineering that supports further evolution of the domain to better meet future project and customer needs.

Application engineering is a standardized process by which projects produce and deliver application systems to customers. In terms of objectives, this is the equivalent of conventional, one–of–a–kind systems engineering. Both conventional and product–family approaches start with customer requirements and produce a set of work products that are meant to satisfy those requirements. Either approach may use commercial off–the–shelf assets or reuse assets produced in developing previous systems. However, in a product family approach, through domain engineering, an organization institutes a simpler, more efficient process of automated system development, supported with standardized, reusable work products.

Application engineering creates and validates a model that is used to generate a required application. The application engineering process focuses effort on requirements and engineering decisions that are sufficient to describe a particular system, given a family of such systems. Work products, including hardware (such as instructions to a silicon compiler, a programmable device, or a circuit board layout assembler), software (including both deliverable and support programs), procedures (such as for use, for testing, or for installation of a system), and documentation (such as the GSEP's system definition, design documents, or user manuals), are derived from these decisions using adaptable forms of those work products provided by domain engineering.

Domain engineering supports application engineering in two ways. First, it creates a set of adaptable work products that correspond to the work products that application engineering projects must produce. By identifying key decisions that are deferred until a particular system is needed and parameterizing a work product to show how it varies as a result of those decisions, domain engineering creates work products that are adaptable to subsequent application engineering decisions. Second, domain engineering defines a standard application engineering process that supports the decision making and the work–product creation appropriate to projects in the business area. The process definition institutes standard procedures and practices that domain engineering augments with an appropriate automated support environment.

5.3 Using the GSEP in Engineering for Product Families

The *Reuse–driven Software Processes Guidebook* provides basic definitions of domain and application engineering processes. To support systems engineering, these definitions must be tailored based on GSEP. However, having been developed before GSEP, these processes as presented here deviate somewhat from pure GSEP–derived processes. Specifically, the scope of the domain engineering process definition is beyond that of GSEP in that it includes activities of the Manufacturing, Deployment, and Operation and Support phases of the “domain system" life cycle. Examples of this are activities which identify the steps to implement an application engineering environment or to support a domain's users during and after delivery). In addition, the application engineering process definition is only intended to be suggestive of what domain engineering should define as an application engineering process.

Figure 40 is a snapshot of the relationships between GSEP–based domain and application engineering processes. Note that a

domain may correspond to a family of similar systems, subsystems, or components. A GSEP–based domain engineering process includes the engineering of both a product family as reusable assets that are used to build applications and a tailored GSEP–based application engineering process. The product family is provided as a part of the technology base input to the application engineering process. A GSEP–based instantiation of the application engineering process includes an environment comprising tools, methods, and resources that are used to build applications (i.e., family members). Note that the product family developed in domain engineering is provided to all application engineering projects that follow the associated GSEP–based application engineering process.

5.3.1 GSEP–based Domain Engineering

The scope of domain engineering is a domain, corresponding to an organization's total `systems engineering system.' A domain includes all application systems to be created over time for customers within a targeted market, as well as a process and supporting environment for developing and delivering those systems. The leveraged Synthesis domain engineering process defined in the *Reuse–driven Software Processes Guidebook* consists of 5 activities which correspond as noted to GSEP activities:

Domain management, which is equivalent to GSEP's Manage Development Effort activity for a domain

Domain definition, which is equivalent to GSEP's Analyze Needs and Define Requirements activities for a domain

Product family engineering, which is equivalent to a GSEP process for the “domain system" subsystem corresponding to the product family component of a domain

Process engineering, which is equivalent to a GSEP process for the “domain system" subsystem corresponding to the process and environment component of a domain

Project support, which is equivalent to GSEP's Validate and Verify Solution activity for a domain but also includes installation, delivery, and support

GSEP activities remain much the same within a domain engineering process, except that each is oriented to a family of systems characterized by commonalities and variabilities rather than to a single system and therefore requires creating adaptable work products. For this reason, several GSEP process terms have special interpretations within the domain engineering context:

System. A domain, corresponding to a family of application systems, subsystems, or components and comprising the product family and the associated application engineering process and environment

Customer Requirements. Business objectives, partially based on the known and projected needs of the market (customers) for products

User Requirements. Domain (problems) knowledge, held by managers and engineers responsible for the application engineering of systems represented by a domain and by users of those systems

Technology Base. Domain (solutions) knowledge held by managers and engineers responsible for the application engineering of systems represented by a domain, including legacy products that are representative of the domain

Product family engineering creates a product family, represented by requirements, design, and implementation work products, that is sufficiently adaptable to represent any member of the product family. This includes an adaptable architecture and assets that are flexibly reused to generate product family members. Rather than being a direct source of revenue, a domain is an investment, primarily intended as a means to leverage the product development efforts of the organization for greater profitability on each product, product version, or product upgrade.

The domain engineering process can be followed to create a domain at the system, subsystem, or component level. Within a domain, each subsystem or component may be an integral part of the domain, a member of a family forming a distinct domain, or a conventionally developed system.

5.3.2 GSEP–based Application Engineering

Application engineering produces a system that is a product family member and that meets the particular needs of a customer and users. Domain engineering defines a product family and a tailored application engineering process to make this a viable capability. This section describes one form that the process could take, based on the definition of application engineering for the leveraged Synthesis process in the *Reuse–driven Software Processes Guidebook* but tailored based on GSEP. This process, as presented here, assumes that domain engineering is tasked to minimize application engineering effort by providing projects with a comprehensive and consistent set of reusable assets and maximum automated support. More modest versions of the process would leave application engineering projects responsible for more of the effort as one way to limit the needed investment in domain engineering.

Domain engineering is able to create a more effective application engineering process by standardizing the requirements, design, and implementation of systems in the domain. Standardization means that work related to similarities among encompassed systems can be completed once by domain engineering. This leads to standardized work products that can then be generated in a tailored form from domain–specific reusable assets. Because standardization provides a foundation for increased automation, the application engineering effort tends to condense into a decision making process emphasizing problem/solution modeling and analyses of alternatives. Each application engineering project focuses its efforts on resolving issues that are specific to the requirements and constraints motivating a particular system.

The leveraged Synthesis application engineering process consists of 4 activities which correspond as noted to GSEP activities:

Project management, which is equivalent to the GSEP Manage Development Effort activity

Application modeling, which is equivalent to the GSEP Define System activity

Application production, which may be fully automated but is outside the scope of GSEP in any case

Delivery and operation support, which is outside the scope of GSEP

Systems produced by application engineering are usually intended to bring in revenue to an organization. Domain engineering defines the activities and work products of the application engineering process to be used by projects so that they have the means to be more profitable. Feedback from application engineering projects to domain engineering is used to improve the domain as a `systems engineering system.'

The inputs to application engineering from domain engineering include an extended technology base that includes work products defining the product family (as a set of reusable assets), methods and tools in an environment that supports the activities of the application engineering process, and possibly resources to perform application engineering activities.

5.3.2.1 Tailoring Management Activities for Application Engineering

Domain engineering tailors two of the GSEP management activities for application engineering, in order to simplify them within the context of a domain (see Figure 41).

Understand Context Activity (1.1.1). Domain engineering provides application engineering with an ability to generate an Estimate of the Situation when supplied with specified information about context for the product; risks and assumptions that have been inherited; and organizational, political, economic, and technical issues that have been identified.

Analyze Risk Activity (1.1.2). Domain engineering simplifies the Analyze Risk activity by constraining and formalizing the potential risks that are associated with producing applications in a specific domain. Domain engineering identifies likely risks and associated aversion strategies which are supplied to application engineering.

In addition, domain engineering supplies project managers with adaptable planning documents that they tailor to direct their project through its increments for development and delivery of a product. This may be supported with process–driven tooling for monitoring progress against a tailored plan.

5.3.2.2 Tailoring Technical Activities for Application Engineering

Domain engineering tailors the technical activities of the application engineering process by providing automated support for all or parts of each activity and providing accompanying adaptable assets that are used in generating required work products for each activity (see Figure 42). The following describes how GSEP technical activities are tailored to define the application modeling portion of application engineering:

Analyze Needs Activity (1.2.1). Domain engineering tailors this activity to fit the domain which constrains the possible stakeholders, problems, environments, and informal functions that are supportable. This activity is supported by automation that assists the application engineer to work with the customer to determine these factors within the framework of the domain.

Define Requirements Activity (1.2.2). Domain engineering provides application engineering with a capability to describe (or model) a required system by making decisions that distinguish that system from other possible systems within the domain. These decisions are constrained to determine correlated, testable behavioral and performance requirements. Performance–related decisions or ones that involve a tradeoff between function and performance may be partially deferred to the Evaluate Alternatives activity. Complete requirements for a system are a composite of requirements common to all systems in a domain, requirements decisions made in this

activity, and requirements that are implied by decisions.

Define Functional Architecture Activity (1.2.3). Domain engineering establishes a standardized functional architecture that is adaptable to supported requirements decisions (equivalent to alternative architectures). This activity is the mechanical generation of one or more functionally–equivalent architectures. Each architecture partitions requirements into functions, identifies lower level functions, and defines the interfaces between functions to provide a solution structure appropriate to the requirements identified during the Define Requirements activity.

Synthesize Allocated Architecture Activity (1.2.4). Domain engineering establishes a standardized allocated architecture that is adaptable to supported requirements decisions, consistent with the standardized functional architecture. This activity is the mechanical generation of alternative allocated architectures for each functional architecture. Each allocated architecture maps a functional architecture into a possible solution, determines its interfaces and technical parameters, integrates designs of lower level components, and determines a physical architecture based on the specific requirements identified during the Define Requirements activity.

Evaluate Alternatives Activity (1.2.5). Domain engineering provides adaptable models to be used in this activity to evaluate the alternative allocated architectures and select a preferred system definition. Models provided address the performance analysis disciplines, sensitivities to variations in system parameters and external environment interactions, technical risks and problems, and trade–offs characteristic of systems in the domain. Analyses are semi–automated with user facilities for controlling models and evaluating results.

Validate and Verify Solution Activity (1.2.6). Domain engineering provides adaptable test procedures to be used in this activity to validate and verify the work products of other activities and of an implemented system solution. Domain engineering also provides adaptable procedures for executing and evaluating test results. Performance of verification and validation is semi–automated with user facilities for controlling solution analysis or execution in real or simulated target environments and for evaluating results.

Control Technical Baseline Activity (1.2.7). Domain engineering builds automatic baselining of all decisions and application work products into the application engineering environment. New versions of affected work products may be produced whenever requirement and engineering decisions are made in the Analyze Needs, the Define Requirements, and the Evaluate Alternatives activities or as a side–effect of any activity that produces derived information (such as test results recorded during the Validate and Verify Solution activity).

5.4 An Example of Engineering for a Product Family

To illustrate the concept of engineering for a product family, this section discusses an example of a product family corresponding to a hypothetical line of utility vans to be built by the passenger vehicles division of an automotive manufacturer. This discussion will focus, in overview, on the needs that motivate this family and how a product family orientation influences system engineering as coordinated domain and application engineering efforts. As noted above, management activities of domain and application engineering are much the same as management in other GSEP–based processes. Because of this, this example is simplified to discuss only technical activities in more detail, recognizing that management is a necessary and integral part of the actual process.

In this example, domain engineering is responsible for creating the capability to produce and deliver any van that is properly part of the product family described here. That capability includes provision of marketing materials, a manufacturing facility, and material supplies of component parts and raw materials. Application engineering comprises the interactions between a customer and a dealer to create an order for a particular van and between the dealer and a manufacturing representative who effect production and delivery of that van to the customer.

In terms of the automobile system example discussed in Section 2, this section discusses primarily issues related to the operational (car) subsystem. However, domain engineering is normally concerned with all life cycle subsystems, including operational, manufacturing, support, and others as appropriate, for a family of systems. Domain engineering may create a life cycle subsystem in either a static (non–varying) form for the entire product line or adaptable to the needs of each project. For example, the manufacturing subsystem of a domain, which is a major component of the application engineering environment (for producing individual vans to fill specific orders), may exist in only one form that is the same regardless of the characteristics of a particular van being produced or may be adaptable so that each project can create a more efficient facility tailored to their particular needs.

In the example, the passenger vehicles division has been producing such vans in 3 limited–option models for several years and wishes to offer greater variety to its customers in the future. To do this, it is setting up an order–driven production facility that will allow flexible optioning of features by its dealers for customers. The targeted invoice price range for these vans is $25,000 to $40,000.

The utility vans product family will have a standard frame but most other features will vary, though only in limited ways in many cases. Based on marketing surveys of customers and dealers, there is a market for vans with the following options (along with others not listed here):

Any combination of fixed and convertible seating and cargo space, allowing for up to 9 adult passengers with minimal storage, 1 passenger with maximum cargo, or anything in between

A cargo storage infrastructure designed to the customers specifications as to number, size, and positioning of compartments, within the constraints of cargo space determined above

A choice of engines that provide different performance based on their power and efficiency characteristics

An ability to emphasize fuel economy (up to at least 30 mpg), flexibility of usage (such as being able to tow a trailer), or safety (such as reinforced side panels), recognizing that conflicts among choices need to identified for the purchaser.

Some of these variability features may translate into adaptability of a single product, such as selection of the option to be able to convert between seating and cargo space; some, such as fixed seating or engine selection, are strictly production options (that is, determined when a particular vehicle is ordered and then produced; others, such as fuel economy, are derivative of other features (vehicle weight and engine selection in the case of fuel economy).

5.4.1 Domain Engineering of the Utility Van Domain

Based on the above objectives, domain engineering, in the Analyze Needs activity, defines a domain of utility vans for the division. It performs an analysis of viability that determines that the division has the technical expertise to be successful and that there is a market adequate to support distribution of 100,000 units per year by this company. This is sufficient to generate profit based on the necessary investment in the domain and projected unit cost profile for a van.

In the Define Requirements activity, domain engineering systematically elaborates the requirements for the family of vans to be developed, taking account of how variability features correspond to different needs. Selected dealers and senior engineers of all disciplines participate to ensure completeness and accuracy.

In defining and evaluating adaptable forms of the functional and allocated architectures for vans, domain engineering cannot identify adequate current technology that will enable any van to be produced with a fuel economy of 30 mpg. Available engines and the minimum obtainable weight dictates a maximum of 28 mpg. Accordingly, this is fed back to the Define Requirements activity for revision of requirements for the van family.

In defining the application engineering process and environment for the utility vans domain, domain engineering defines the driving procedures, maintenance, manufacturing, distribution, marketing, and recycling life cycle subsystems appropriate to that domain. These subsystems may be made adaptable to variabilities in the product family leading to product variations such as different maintenance schedules and procedures for vans with high–performance versus standard engines. The marketing subsystem provides dealers with a capability to help customers configure a van to suit their needs, verifying that no manufacturing constraints are violated. This includes an ability to generate realistic pictures of a van as specified by the customer and a simulation capability to evaluate trade–offs in expected operational usage.

To verify the domain, experienced engineers evaluate all work products during the domain engineering effort. To validate the domain, dealers and engineers, both ones who were involved in the requirements phase and others who were not, evaluate the van production capability as it evolves. This includes use of early versions of all subsystems, particularly the marketing subsystem. As the capability matures, prototype products are produced and evaluated, leading to refinements. Evaluations continue even after full production is active to ensure that the domain evolves to remain viable to all concerned.

5.4.2 Application Engineering in the Utility Van Domain

Application engineering, being standardized across the utility van product line and heavily automated, is a rapid and highly responsive process. The dealer and customers have almost direct control over the production and delivery of a highly tailored product. Dealers and customers are presented with a series of restricted questions that determine the options to characterize a particular, producible van. Decisions are made and changed as the customer's understanding of the alternatives for an end product improve.

The customer's decisions on options are assisted with an evaluation capability to view a simulated model of the vehicle to be produced. The model supports observation of both appearance and performance characteristics under a variety of anticipated operating conditions so that the customer can weigh the implications of alternative decisions. Although some feel overwhelmed by too many choices, dealers help them by fitting them into a few traditional categories of need and making many of the choices accordingly; most find the ability to view realistic models of the finished product helpful.

When the customer is happy with the particular van's characteristics, an order is generated for the dealer and transmitted to the

factory. A van is produced to satisfy the specified options and delivered. At the same time, an owner's manual, including tailored driving procedures and maintenance instructions matching the van's specific features, is generated for the customer.

Although this example shares much in common with conventional manufacturing practices, it differs in its explicit and systematic focus on standardizing the product line upon common and variable features in products. Much more of the process is partially or fully automated. Consequently, support for much more extensive diversity in the product line becomes feasible, with much greater emphasis on allowing the customer to obtain a personally tailored product. Nevertheless, some things do not change; getting the best price still requires the customer to get bids from competing dealers.

5.5 Tailoring the Product Family Process

The degree to which a product family process actually exhibits the features called out in Section depends on the needs and abilities of the adopting organization. A GSEP–based Synthesis process of engineering for a product family should be tailored to suit the needs of each particular organization.

One form of tailoring is based on Consortium work in process improvement, specifically in reuse adoption. The *Reuse–Driven Software Processes Guidebook* (Software Productivity Consortium 1993a) describes alternative forms of Synthesis process which derive directly from critical success factors of reuse adoption. Any of these forms of tailored processes can be built around the GSEP. The guidebook defines opportunistic and leveraged forms in detail.

An inherent option for tailoring a Synthesis process occurs in the need to choose specific development methods for management, requirements analysis, design, implementation, testing, etc. Both the Consortium's Synthesis methodology and the GSEP were conceived specifically to allow an organization the choice of its preferred methods and to orient process definitions to being tailored accordingly.

An important consideration in creating a tailored process is achieving a suitable balance between the amount of engineering done at the domain level and the amount done at the applications level. Shifting engineering from the applications level to the domain level results in more standardized architectures and greater reliance on reusable assets, reducing the effort required to produce a system. However, this shift is also accompanied by an increase in the constraints domain engineering imposes on the application engineering process and a decrease in the flexibility of application engineering to accommodate system needs that were not anticipated by domain engineering. The shift of resources from direct revenue producing systems (i.e., application systems) to the indirect revenue producing system (i.e., a domain) must be justified by greater profitability, due to lower unit cost, reduced errors, and shorter time to deployment, on each system produced.

Construction of modern, complex systems is a major undertaking that often requires the coordination of many large groups of engineers with expertise in diverse fields. Some of these groups may be organizations in separate companies, working jointly or as subcontractors, to deliver the required system. Engineers follow a discipline of systems engineering, in part, to partition the problem and apply appropriate expertise to solve each facet of the problem most effectively. This partitioning into subsystems and components is compatible with engineering for product families, particularly when engineers are able to follow a systematic approach that results in similar partitionings of similar systems. Each subsystem may correspond to a member of a family of similar subsystems (i.e., a hardware, software, or human system). The required subsystem can, in turn, be developed as a one–of–a–kind system or, based on a domain of such subsystems, as an instance of a product family.

This page intentionally left blank.