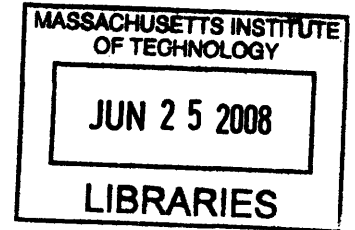


Restoring Product Focus Across the Value Stream
Through Organizational Restructuring

by

Jeffrey M. Pasqual

Bachelor of Science in Engineering, Mechanical and Aerospace Engineering
Princeton University, 2003



ARCHIVES


Submitted to the Engineering Systems Division and the Sloan School of Management
in Partial Fulfillment of the Requirements for the Degrees of

Master of Science in Engineering Systems
and
Master of Business Administration

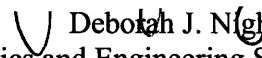
In conjunction with the Leaders for Manufacturing Program at the
Massachusetts Institute of Technology
June 2008

© 2008 Massachusetts Institute of Technology. All rights reserved.


Signature of Author

 Engineering Systems Division
Sloan School of Management
May 9, 2008


Certified by

 Deborah J. Nightingale
Professor of the Practice of Aeronautics and Astronautics and Engineering Systems
Thesis Supervisor


Certified by

 Roy E. Welsch
Professor of Statistics and Management Science
Thesis Supervisor

Accepted by

 Richard C. Larson
Professor of Engineering Systems
Chair, Engineering Systems Division Education Committee

Accepted by

 Debbie H. Berechman
Executive Director, MBA Program
Sloan School of Management

This page has been intentionally left blank.

**Restoring Product Focus Across the Value Stream
Through Organizational Restructuring**

by

Jeffrey M. Pasqual

Submitted to the Engineering Systems Division and the Sloan School of Management
at the Massachusetts Institute of Technology on May 9, 2008
in Partial Fulfillment of the Requirements for the Degrees of

Master of Science in Engineering Systems

and

Master of Business Administration

ABSTRACT

Businesses take deliberate action to change their internal context when managers believe that better performance lies beyond the capabilities of assets in their present configuration. A typical course of action is reorganization. A key consideration for organizational design is how the relationship between an organization's structure, the structure of its products, and the structure of its processes influence the value delivered to customers. In some sense, products, processes, and the organization should "fit" each other. This thesis presents a framework for thinking about product architecture, enterprise architecture, and the value stream of processes that binds them together. Critical to any enterprise architecture are process owners that control and improve organizational processes and product owners that manage the end-to-end development of products. When a product is significantly complex, independent tiers of product ownership might be established to ensure that different levels of products – systems, subsystems, or components – are managed with appropriate developmental objectives in mind. For example, some components must be distinct to a single product; other components can be common among several products. The proposed framework shows how product and enterprise architectures can be integrated to support the development of complex systems.

The thesis also presents a case study to which the proposed framework is applied. The study focuses on a business that has recently restructured its organization to achieve better alignment with the complex products it develops. Using the proposed framework, the new organizational structure is evaluated to determine if the new enterprise architecture positions the business to increase customer value and accomplish its long-term goals.

Thesis Supervisor: Deborah J. Nightingale

Title: Professor of the Practice of Aeronautics and Astronautics and Engineering Systems

Thesis Supervisor: Roy E. Welsch

Title: Professor of Statistics and Management Science

This page has been intentionally left blank.

Perfection is like infinity. Trying to envision it (and to get there) is actually impossible, but the effort to do so provides inspiration and direction essential to making progress along the path.

Womack and Jones
in *Lean Thinking*

This page has been intentionally left blank.

Acknowledgements

In the business world, every person tends to receive a label – co-workers and supervisors; suppliers and customers; competitors and partners; and the all-encompassing one – stakeholders. I find most of these to be rather impersonal and much prefer to recognize the people in my life as family, friends, colleagues, and mentors. So with all business formality aside, it is with deep gratitude and appreciation for their support and guidance that I acknowledge the following individuals.

Out in the field, I thank the team at Northrop Grumman Space Technology for welcoming me into the Space Park campus and embracing the opportunity to sponsor an LFM internship for the first time. I thank the vice president for hosting me in the Mission Excellence organization. I also thank the director of university alliances for having the persistence to engage the LFM program all these years (you finally got an intern!). I thank the entire Mission Excellence team for making me feel at home on the 5th floor of E2. I thank those who arranged for personalized tours of facilities across the campus. I thank the administrative staff for their help and patience, keeping (and getting) me out of trouble around the office – always with a smile and a sense of humor. I thank those who gave of their precious time to review this thesis and other material for public release. I especially thank those individuals who provided direction to my project. Finally and most of all, I thank my supervisor for setting me up for success, accommodating my various needs without a second thought, and most importantly, sharing his valuable time and wisdom along the way. He is certainly more a mentor than a supervisor in my book. I look forward to when our paths may cross again.

Back at the office, I thank my faculty advisors, Deborah Nightingale and Roy Welsch – two veterans of the LFM internship and thesis model, whose steady counsel kept me focused on the project amid endless distractions in southern California. I thank the entire LFM staff – Jon Griffith, Jan Klein, Jeff Shao, Nancy Young-Wearly, Davicia Smith, and many others – for their internship support and commitment to the LFM program. I especially thank Don Rosenfield for his tireless devotion to the program, taking a personal interest in the education of each and every LFM student. Lastly, I thank my 46 classmates in the LFM Class of 2008. I am humbled and inspired by their intellect and ambition, and grateful for their friendship. It has been a privilege to share this two-year journey with them, and I wish unbounded success to all in the future.

On the home front, I thank my brothers, Michael and Anthony, for faithfully supporting their older brother – even when out of touch, I know they have my well-being at heart. I especially thank Michael for enduring a graduate student as a roommate – one who's been mentally distracted and physically relocated at times. Finally and most of all, I thank my mother, Elaine, and father, John, for their unwavering love and guidance. I would never have come this far without their selfless commitment to my brothers and me. I hope that I have made them proud.

This page has been intentionally left blank.

Biographical Note

Jeffrey Michael Pasqual was born in Lubbock, Texas. As the oldest son of an Air Force pilot, he experienced the thrills and trials of growing up in a military family. His father and mother took great care to provide the most nurturing of environments for him and his two younger brothers, a relentless undertaking for which he and his brothers are eternally grateful.

Jeff attended Princeton University in Princeton, New Jersey, where he majored in Mechanical and Aerospace Engineering. He also pursued a certificate from the Woodrow Wilson School of Public and International Affairs. While at Princeton, Jeff completed the Air Force Reserve Officer Training Corps program, serving as cadet corps wing commander for a semester during his senior year. Upon graduation, he was commissioned as a second lieutenant in the United States Air Force. His first duty station was located at Hanscom Air Force Base, Massachusetts, where he served as a developmental engineer at the Air Force Research Laboratory, Space Vehicles Directorate. At the lab, his responsibilities included the development of modeling and simulation software tools in support of Air Force missile programs.

As a member of the Leaders for Manufacturing Class of 2008 at MIT, Jeff served as co-chair of the LFM ambassador program, which coordinates campus visits for prospective students. Jeff also served as chair of the student internship committee, which is responsible for developing and facilitating the assignment process for six-month internships which LFM students complete at an industry partner company. Jeff completed his own internship at Northrop Grumman Space Technology in Redondo Beach, California.

Jeff has long been a personal fitness junkie, who rarely misses an opportunity to go running along the Charles River or Minuteman Bikeway. He also has a growing interest in hiking and mountain climbing, fueled by visits to Yosemite and Death Valley National Parks, among others, during his LFM internship on the west coast.

This page has been intentionally left blank.

Table of Contents

Acknowledgements.....	7
Biographical Note	9
Table of Contents	11
List of Figures	13
1. Introduction.....	15
1.1. Thesis Structure	17
1.2. Motivation.....	17
1.3. Case Study Introduction.....	19
1.3.1. Description of Organization.....	20
1.3.2. Description of Current Situation.....	21
1.4. Research Approach.....	23
1.4.1. Problem Statement.....	23
1.4.2. Objectives	25
1.4.3. Sources and Methods	26
2. Frameworks for Complex System Management.....	31
2.1. General Viewpoints	32
2.1.1. Systems	32
2.1.2. SIPOCs.....	34
2.2. Product Systems.....	36
2.2.1. Form and Function.....	37
2.2.2. Product Architecture	39
2.3. Process Streams	40
2.3.1. Product Essence	42
2.3.2. Process Value.....	44
2.4. Integrated Product and Process Framework	46
2.5. Enterprise Systems.....	49
2.5.1. Form, Function, and Architecture.....	49
2.5.2. Product Management	51

2.6.	Integrated Product Management Framework.....	53
2.6.1.	Common Enterprise Architectures.....	53
2.6.2.	Products Relative to the Enterprise.....	55
2.6.3.	Independent Tiers of Product Management.....	57
2.6.4.	Distinctiveness and Commonality	59
2.7.	Section Summary	63
3.	Case Study for Complex System Management	65
3.1.	Satellite Architecture	66
3.2.	Satellite Life Cycle	69
3.3.	Organizational Structure	72
3.3.1.	Pre-Realignment Architecture	73
3.3.2.	Post-Realignment Architecture.....	76
3.4.	Evaluation of Architectural Fit	79
3.4.1.	Intra-Stream Value.....	79
3.4.2.	Inter-Stream Value.....	81
3.5.	Challenges Following Realignment.....	85
3.6.	Section Summary	87
4.	Conclusion	89
4.1.	Research Outcomes.....	89
4.2.	Future Research Opportunities	90
	References.....	92
	Appendix A.....	95

List of Figures

Figure 1: Conceptual model for business performance	18
Figure 2: System view	33
Figure 3: Model of a generic system.....	34
Figure 4: SIPOC view	35
Figure 5: SIPOC chain	35
Figure 6: Object-system-product hierarchy	36
Figure 7: Decompositional view of a product's form.....	38
Figure 8: View of a product's function.....	39
Figure 9: Model of a product system	40
Figure 10: Process stream (product life cycle)	41
Figure 11: Zooming in and out of the process stream	42
Figure 12: Process stream with consideration for product essence	43
Figure 13: Process stream with consideration for process value	45
Figure 14: Zigzagging principle of axiomatic design methodology.....	47
Figure 15: Integrated product and process framework for complex product systems.....	48
Figure 16: Views of an enterprise's form and function	50
Figure 17: Model of an enterprise system.....	51
Figure 18: Integration of product and enterprise architectures.....	52
Figure 19: Common enterprise architectures	54
Figure 20: Integration of complex product and enterprise architectures	56
Figure 21: Independent tiers of product management – matrix organization.....	58
Figure 22: Independent tiers of product management – center organization (Toyota).....	61
Figure 23: Spectrum of approaches to product management.....	62
Figure 24: Decompositional views of a space system	67
Figure 25: Decompositional view of a spacecraft bus	68
Figure 26: Generic satellite architecture	70
Figure 27: Process stream (product life cycle) for satellite and spacecraft subsystem.....	71

Figure 28: Organizational structure – pre-realignment.....	73
Figure 29: Integrated product and enterprise architectures – pre-realignment.....	75
Figure 30: Organizational structure – post-realignment.....	76
Figure 31: Integrated product and enterprise architectures – post-realignment	78
Figure 32: Independent tiers of product management – post-realignment	83

1. Introduction

The architectures of the product, its construction process, and the organization that manages them should “fit” each other.

(Rechtin 2000, 159)

In plain and concise language, this statement – one of Rechtin’s heuristics¹ – captures a key insight for professionals faced with the issues this thesis intends to address. For the author, it has provided a conceptual hypothesis and thematic reminder during the consolidation of his research into a coherent written work. For the reader, this heuristic represents a “bottom line” of sorts, offered here at the outset of this thesis.

In dissecting this heuristic, the fundamental things of interest are *product*, *process*, and *people*.² Each of these things has an *architecture*, which refers to the thing’s underlying structure. The structures of products, processes, and organizations are inherently interrelated and intertwined. One cannot begin to consider their architectures in isolation from each other. For example, *products* that comprise hardware and software elements arise from *processes* defined for hardware and software development, which are executed by groups of *people* responsible for hardware and software. Rechtin’s heuristic presumes these interconnections and further prescribes that the architectures “fit” each other. By definition, this “fit” should, in some sense, be “good” or appropriate. An indicator or indirect measure of “fit”-ness is, generally speaking, the value provided to the stakeholders associated with the endeavor at hand. Of course, some minimum threshold of “fit”-ness must be present before *any* value can be delivered to these stakeholders. That is, an organization carries out processes and affects products *only* when possessing the capability to do so. As used here, capability implies a certain “fit” between the structures of products, processes, and people, and this capability, when exercised, yields value for stakeholders.

¹ In numerous publications, Rechtin champions the heuristic methodology for architecting systems. To use his definition, *heuristics* are “empirical rules of thumb derived from experience and judgment, useful for attacking problems too complex to be solved by analytical techniques alone” (Rechtin 1992, 66). The late Dr. Eberhardt Rechtin, among other notable government and industry roles, served as president of The Aerospace Corporation and later founded the graduate program in systems architecting at the University of Southern California.

² *People* is substituted here for *organization* to yield the three P’s often considered the building blocks of business. Yet the term architecture is more commonly associated with an *organization*, which denotes a group of *people* brought together for a particular purpose.

Quite simply, this thesis investigates the notion of architectural “fit” and its correlation with delivered value. In other words, how do the *relationships* between (1) an organization’s structure, (2) the structure of the products it delivers, and (3) the structure of the processes it executes to yield these products *influence* the *value* delivered to its customers and other stakeholders? Within this overarching research question, the key words – *relationships*, *influence*, and *value* – are difficult to treat in a quantitative manner. A host of related questions follows. How does an organization measure its “fit” with its products and processes? To what extent should the organizational structure mirror the functional or physical decomposition of its products or the ordered set of processes that transforms its products from concept to design to finished item? More generally, what organizational structure yields a threshold capability or an optimized capability, and how might these be expressed as quantified levels of capability? In turn, how does the level of capability relate to, by way of causation or correlation, the value provided to each stakeholder? And finally, how is value defined and measured relative to each stakeholder – to include customers/users, suppliers, partners, and shareholders?

This thesis does not make a rigorous attempt to address these questions using statistical analysis or other mathematical methods. In accordance with Rechtin’s approach to *architecting*, this research has more often been governed by the heuristic methodology. That is, empirical rules of thumb, lessons learned, and subjective insights, rather than formal problem-solving methods and algorithms, have guided the questions posed, data and information collected and analyzed, and conclusions drawn. Likewise, the research outcome is not a generalized solution or codified procedure for determining optimality. Rather, results emerge in the form of new or expanded heuristics that build upon or refine the existing ones, adapting for the scope and context of the problem at hand. In many cases, such insights are extremely useful to executives, managers, and analysts alike when resource limitations preclude detailed decision analysis. Some readers may question the validity of results not directly supported by quantitative analysis or dismiss the utility of qualitative decision criteria. To those readers, the author defers to another of Rechtin’s heuristics: *An insight is worth a thousand analyses* (Rechtin 2000, 96).³

³ Nonetheless, the author agrees with these words attributed to Lord Kelvin: “... when you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meagre and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely in your thoughts advanced to the state of Science...”

1.1. Thesis Structure

This document consists of four major sections, including this introductory part. In addition, an appendix contains supplemental information. The first section, *Introduction*, explains the motivation for this thesis and the approach used to conduct research. It also provides an introduction to the principal case study for this work. The second section, *Frameworks for Complex System Management*, develops the relationships between products, processes, and organizations and proposes a framework for thinking about architectural ‘fit.’ The third section, *Case Study for Complex System Management*, applies this framework to the principal case study, which served as the guiding influence for this research effort. The fourth and final section, *Conclusion*, summarizes and reiterates the key insights gained from this research effort and suggests future areas of study. Appendix A provides a description of the diagram methodology used in many of the figures.

1.2. Motivation

While questions regarding architectural “fit” are, to some degree, relevant for organizations of all types, they are arguably of most concern to for-profit businesses.⁴ Already, the opening discussion has been biased towards commercial organizations in referring to products and customers. Adapted from *Saloner*, Figure 1 illustrates a conceptual model for business performance, from the perspective of business strategists. Relating this model to the opening discussion, *value* is clearly captured in “Performance”, whether the measure is revenue (corresponding to customer value) or earnings (corresponding to shareholder value). The three architectures exist within the “Context” of the business. In particular, *people* and *process* architectures are captured in “Owned/Employed Assets” and “Structure of Owned/Employed Assets”, which contribute to the “Internal” context; and *product* architectures are accounted for in “Market Setting” and “Non-Market Setting”, which contribute to the “External” context, in the sense that products are in response to (driven by) the external environment.

⁴ Rehtin suggests characterizing organizations by their fundamental purposes – *bureaucratic* (to follow established rules), *profit-seeking* (to increase the bottom line), and *cultural* (to create a compatible team) (Rehtin 2000, 107). The author suggests adding a fourth to this list: *cause-driven* (to promote thought and action for an issue). Accordingly, for-profit businesses have an inherent incentive to take actions that increase value delivered to their stakeholders, especially customers of their products.

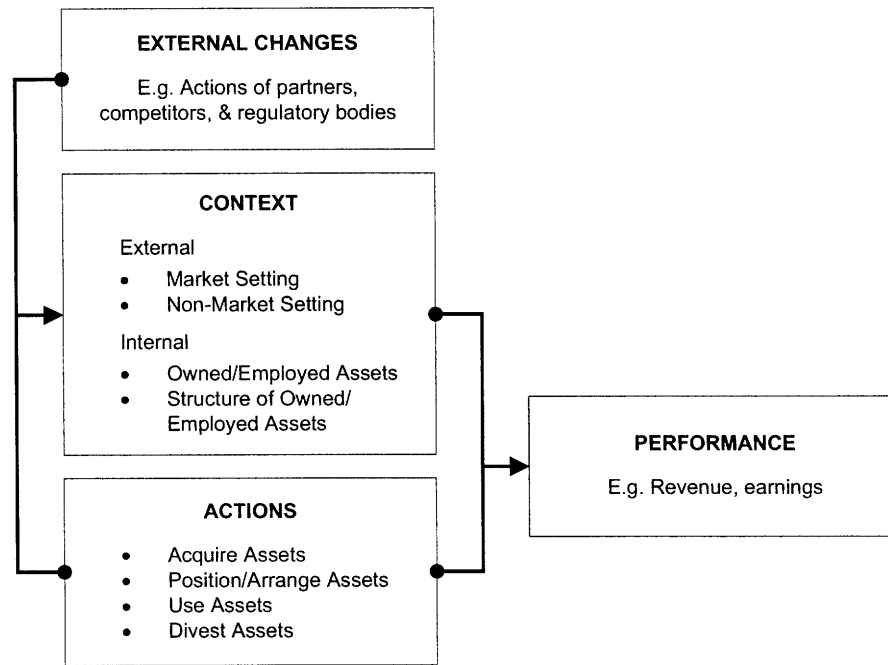


Figure 1: Conceptual model for business performance (adapted, Saloner 2001, 20)

The model in Figure 1 depicts “codetermination of performance” (Saloner 2001, 3) by the business’s actions and the context in which they are taken. In addition, the effect of certain actions, whether deliberate or not, may be to alter the context of the business. Questions about architectural “fit” are associated with this two-stage influence on business performance (i.e. actions changing context codetermining performance). The general question for the strategist might be stated as: What (strategic) actions, if any, should the business take now to alter the context in which (tactical) actions are taken later, such that performance will improve? In turn, the specific question for the architect might be expressed as: What actions, if any, should be taken to change how employees (*people* assets) are grouped and linked with respect to operations (*process* assets), and vice versa, such that *products* respond to the external environment?

In general, businesses take deliberate action to change their *internal* context when managers believe that better performance lies beyond the capabilities of existing assets in their present configuration. At a high level, “better performance” could be tagged as revenue growth, cost reduction, or both. Under certain circumstances, the course of action could involve *reorganization* – the thorough alteration of the structure of a business (American Heritage Dictionary 2004). While this term is more often identified with the restructuring of people (most

outwardly in redrawing organizational charts and renaming business units and subunits), the underlying impetus for change is always the need to redefine, regroup, redistribute, or otherwise reconfigure processes. That is, adjustments to “what” the business does, in terms of process execution, are determined *before* adjusting “how” the business does it, in terms of resource allocation – human, physical, and so on.

The present research has been motivated and steered by the example of a business that currently finds itself in this precise situation. In particular, the executive managers have set strategic goals for revenue growth and are implementing a reorganization to position the business to achieve these goals. For the author’s purposes, the distinct relevance of this case is how explicitly the change in organizational structure is being shaped by the structure of their products and processes. Moreover, the external context of this business – to include the nature of its products, interactions with customers and suppliers, and other industry attributes – provides an especially rich setting in which to explore the architecting questions raised here. As the central case study for this research, this example is described and analyzed in detail throughout the body of this work. The subsection that follows provides an introduction to the business and its current situation.

1.3. Case Study Introduction

The “business” alluded to previously is Northrop Grumman Space Technology (NGST). As of the publishing of this thesis, NGST operates as one of seven business sectors within Northrop Grumman Corporation (NGC) – “a \$32 billion global defense and technology company whose 120,000 employees provide innovative systems, products, and solutions in information and services, electronics, aerospace and shipbuilding to government and commercial customers worldwide” (Northrop Grumman Corporation 2008). For the purposes of this case study, NGST is considered apart from NGC as a whole, such that terms like *organization* and *enterprise* refer to the business sector on its own.⁵

⁵ The case study is limited to a single NGC business sector. As will be described in the next subsection, the author’s research was conducted solely at NGST and the situation of interest was largely self-contained within this business sector. This does not imply that the concepts addressed in this thesis are not equally applicable or relevant at the corporation level.

1.3.1. Description of Organization

A general description of NGST is provided here in the form of a business strategy, using the construct proposed in *Saloner*.⁶ While this portrayal of NGST is supported by company-released information and interviews with employees, it is ultimately the author's interpretation and does not necessarily represent the expressed views of NGST. This description is intended to present a brief profile of the organization rather than a comprehensive analysis.

Goals. In general, NGST strives to be the *technology leader* for all market/product segments in which it competes. By many accounts, NGST has long sustained a reputation among customers and competitors as the “go-to” business for state-of-the-art technology and the most challenging product development efforts. A closely related goal is to deliver the *highest quality products* in terms of performance and reliability. To some extent, these priorities challenge NGST's ability to compete as the lowest cost provider, yet customers are often willing to accept a price premium for the sake of reliable products that meet demanding performance requirements. As a long-term goal, market dominance – having the largest market share of revenue – is less significant to NGST in a consolidated industry of only a half dozen competitors. Still, NGST has recently identified *revenue growth* as an important goal, given projected market opportunities in the next few decades.

Scope. NGST is primarily a provider of space systems and related technologies to the United States government. Its major market segments (by customer) are civil and military organizations with mission areas that require space-based assets. Its major product segments (by purpose) are communications satellites, earth observation/remote sensing satellites for environmental purposes, earth observation/remote sensing satellites for surveillance purposes, and space telescope observatories. NGST maintains the in-house capability for end-to-end product development, to include the design, manufacture, integration, and test of entire satellite systems (i.e. spacecraft bus and payload). In addition, NGST invests heavily in technology development for the majority of satellite subsystems and components.

⁶ *Saloner* suggests that a coherent business strategy should have four elements: (1) long-term *goals* that describe “where” the business is going, (2) *scope* that defines in “what” activities the business engages, (3) *competitive advantage* that defines “how” the business achieves its goal within its scope, and (4) *logic* that explains “why” the intended competitive advantage is sustainable (Saloner 2001, 19-23).

Competitive Advantage. A principal source of competitive advantage for NGST is its system engineering capability. In particular, expertise in the system engineering discipline enables technology infusion in a manner that enhances operational performance without compromising reliability. Advancing the state-of-the-art while mitigating the inherent risks of new technology is a significant capability, especially within the context of satellite development. In many respects, this distinguishes NGST from competitors and attracts the most technically complex development projects from customers. NGST also enjoys a competitive advantage via its industry position as one of only a few domestic system integrators – having the facilities and infrastructure to integrate, test, and deliver complete satellites. This draws implicit support and protection from the United States government. In addition, NGST’s position is reinforced by a longstanding relationship with the civil and military customers that have propelled the satellite industry over the past fifty years. Some suggest that NGST has further endeared itself to customers by cooperating more as a partner than a contractor during development efforts.

Logic. Overall, NGST has achieved and sustains a position as the satellite industry’s technology leader because it effectively leverages its technology investment during the development phase of each and every operational system. In part, success depends on keeping development work for the majority of key components in-house. Without internal control of design and manufacturing processes, technology insertion might occur more slowly or with less reliability. Moreover, the system engineering effort might grow increasingly difficult when required to coordinate across a larger set of subcontractors and suppliers. The logic of the strategy summarizes as follows. To effectively capitalize on technology investment, NGST maintains end-to-end product development capabilities, which have built and internalized system engineering expertise at the spacecraft bus and payload levels, which has established NGST as a system integrator reputed for fielding reliable state-of-the-art technology.

1.3.2. Description of Current Situation

As noted above, NGST has established *revenue growth* as a primary goal during the coming years. The executive management team believes that bringing in new business (i.e. government contract awards in existing and previously untapped market segments) hinges upon two key ingredients: (1) addressing customer needs and (2) delivering competitive products. Many managers at NGST refer to these drivers as *customer focus* and *product focus*, respectively, to

imply that each is a mindset for the organization to embrace. Having identified these revenue drivers, the executive management team assessed that NGST's current organization was not configured to efficiently pursue future growth. Consequently, the team devised and initiated a realignment of the organizational structure.⁷ The realignment brings about two significant adjustments, which position NGST for ensuring *customer focus* and *product focus*, respectively: (1) splitting the business development and program management functions into separate customer divisions, each responsible for a different market segment and (2) merging the design and production functions into a single product development division. The details of these changes, and others not specified here, are presented in a later section. An immediate objective of the realignment is to clarify the roles and responsibilities of these newly formed divisions.

An important consideration surrounding the decision for NGST's latest reorganization was progress made since the previous reorganization. Specifically, the prior organizational structure consisted of functional groups with explicit ownership of processes, grouped into *core* process areas (e.g. engineering, production and supply chain) and *enabling* process areas (e.g. human resources, information technology). Under this structure, NGST embodied a *process focus*, emphasizing that all processes be clearly defined, documented, measured, and managed for compliance and improvement. A formal Six Sigma program emerged as a focal point for process improvement and process control initiatives. Moving forward with the current realignment, the executive management team stresses two points to the organization: (1) NGST's process orientation in prior years has provided a foundation on which the organization can progress and (2) their process-mindedness should not be diminished as other business concerns receive greater emphasis.

While the new organizational structure at NGST has been implemented (i.e. organizational charts redrawn, divisions renamed, and senior management reshuffled), the underlying changes in mindset and behavior believed to enable revenue growth are obviously not instantaneous. Recall that reorganization is a means of altering the internal context of a business through repositioning of assets. Yet a new organizational structure in itself does not determine better or worse performance. NGST is now faced with leveraging its new structure to provide the

⁷ The executive management team at NGST prefers the term *realignment*, rather than *reorganization*, to underscore the intent of the restructuring – *aligning* the organization with its customers and products. These two terms are used interchangeably throughout this thesis.

intended focus on customers and products. At this early stage, structural decisions continue at a greater level of detail. Each division is investigating how to configure its inherited processes, in conjunction with selecting whom to assign responsibility for them. In addition, each division is examining how to establish interfaces with other divisions and external organizations, to include customers, partners, and suppliers. These are the types of decisions for which architectural “fit” with products and processes and their expected influence on business performance are predominant considerations.

1.4. Research Approach

The author’s research originated and evolved during a six-month internship conducted at NGST’s Space Park campus in Redondo Beach, California. For the most part, the NGST case study became the starting point and guiding influence for the entire effort. The present situation at NGST developed as an unanticipated sequence of events (from the author’s point of view) during the internship period. The announcement and implementation of the realignment, which were separated by several months, both occurred as part of the author’s on-site experience. As these events unfolded, the author and his NGST supervisors identified issues of particular relevance to NGST that also showed potential for a greater research context. The workable issues that emerged centered on the concept of *product focus*, which became the tag line for the author’s research throughout the internship.

1.4.1. Problem Statement

As noted earlier, the term *product focus* is used at NGST to denote a general approach to ensuring the *competitiveness* of their products in the marketplace. While technical performance and reliability remain important to NGST’s government customers, affordability and speed of development are increasingly becoming higher priorities for (existing and potential) customers. Consequently, these considerations are key differentiators from competitors. Delivering products at lower cost and on a faster schedule requires a broader scope of management than simply satisfying technical specifications. And so, the complete realization of *product focus* entails two features: (1) mechanisms to ensure *accountability* for products and (2) methods to manage product *evolution* with respect to the full scope of customer needs – technical performance, cost, and schedule.

By all accounts, NGST implements both of these elements to the fullest extent at the satellite level. That is, program offices at NGST (i.e. groups responsible for executing a particular government contract) maintain continuous oversight of the performance, cost, and schedule objectives for their respective contracted end systems. However, oversight mechanisms at levels below the end system are less rigorous, especially with respect to subsystems and components that are *common* (by some comparison of function and performance) across multiple end systems. In other words, *product focus*, as defined above, is not fully realized at NGST when viewed from outside the boundaries of a particular program office.

The general problem facing NGST is an *incomplete* ability to enhance the competitiveness of their products. This does not imply that their products are uncompetitive (clearly, NGST's sustained position in the industry indicates the opposite), but rather that NGST could better manage the *likelihood* of delivering competitive products. A primary gap lies in accountability – the first element of product focus – for products defined at levels below the end system (e.g. subsystems, components).⁸ In turn, management of these products' evolution – the second element of product focus – occurs unsystematically through an often uncoordinated series of development decisions that serve one program office at a time. The outcome is *uncertainty* about the past, current, or future performance, cost, and cycle time of these products, when viewed irrespectively from the end systems for which they are developed. As will be discussed in a later section, drawbacks to this outcome include (1) repetition of presumably non-recurring development tasks, (2) unclear cost and schedule targets for driving process execution, and (3) inaccurate cost and schedule estimates for developing contract proposals.

An underlying difficulty is ambiguity about what constitutes a *product* at these lower levels. That is, considering the breakdown of the end system, are products defined as subsystems, components, parts, or all of these? Similarly, considering the sequence of development stages, are products defined as specification documents, build-to drawings, physical hardware (e.g. work in process or finished items), or all of these? The answers to these seemingly basic questions are less than obvious for an organization like NGST who undertakes end-to-end development and full-up integration of complex end systems.

⁸ As will be discussed in a later section, *product* is a relative term that shifts identity in the sequence of suppliers and customers. It might be said that *end* products are delivered to *end* or *external* customers. Accordingly, *lower level* products, as described here, might be considered *intermediate* or *internal* products from the supplier's viewpoint.

In short, the problem statement summarizes as follows. **Organizations often struggle to implement effective oversight mechanisms for intermediate products *in conjunction with* well-established oversight mechanisms for their end products.** This problem is especially evident in the context of contracted end products. Rigorous management of contracted end products independently benefits each customer, but often dominates similar management of intermediate products that are common across multiple end products, which has the potential to simultaneously benefit all customers and improve the competitiveness of all levels of products.

1.4.2. Objectives

The author's research suggests the following hypothesis in response to the above problem statement. **Organizations can effectively implement multiple tiers of product oversight when their product and process architectures are made *explicit* in their organizational structure. Formalized product and process ownership ensures accountability and creates interfaces at which tiers of product management (e.g. end products versus intermediate products) balance their influence on product evolution.** In support of this hypothesis, the goal of this thesis is to elaborate the concept of architectural "fit" and its importance with respect to realization of *product focus*. There are several objectives that address the general situation for organizations and several others that pertain to the NGST case study.

- Establish a general framework for the product life cycle that considers product and process architectures apart from the organization.
- Establish a general framework for product management that overlays organizational structure on the product life cycle.
- Apply the presented framework within the developmental context of space systems and the organizational context of NGST.
- Evaluate the architectural "fit" and realization of *product focus* at NGST through comparison of their organizational structure before and after the realignment.

The first two objectives are primarily addressed in the second section of this document. The last two objectives are covered in the third section of this document.

1.4.3. Sources and Methods

This subsection describes the information sources and collection methods that have supported this research effort. In general, the author used a balance of (1) published literature from prior research and (2) information obtained during the internship to develop the conceptual frameworks and case study analysis presented in this thesis.

Research Literature

In order to develop an understanding of the relevant theory and concepts related to architectural “fit”, the author surveyed several major research topics in the published literature. Each subject area and the chief works reviewed by the author are described below. In name, the first three topics appear to isolate the product, process, and people architectures. However, any reader quickly recognizes that detailed investigation of one necessarily informs the study of the other two. The fourth and final topic provides information on space systems and the greater aerospace industry in support of the NGST case study.

System architecture. As used here, this topic focuses on the technical design of products – in particular, the translation of customer/user needs into value-related functions that are mapped to physical or informational elements. The literature on system architecture is vast, and major product categories (e.g. electronics hardware, computer software, automotive systems, and aerospace systems) have detailed architecting methodologies of their own. For the purposes of this research, the basic principles of system architecture were sufficient. The collection of lecture slides and notes from Professor Edward Crawley’s *System Architecture* course at MIT provides a thorough overview of key definitions and ideas. The course material draws from several pivotal works in this area, to include *Suh* (1990) and *Rechtin & Maier* (2000). It also introduces the object-process methodology of *Dori* (2002) that utilizes rigorous diagramming conventions for system modeling, which will be featured throughout this thesis.

System development. This topic considers the structured set of processes that creates product solutions in response to customer/user needs. The literature for product design and development is also extensive. *Ulrich & Eppinger* (2004) presents a generic product development process for products that are engineered, discrete, and physical – from simple devices to complex equipment and instruments. *Adamsen* (2006) proposes a development framework for complex systems,

which encompasses aerospace systems, ship systems, and ground systems for governments and businesses that operate large-scale equipment. As an extension to system development, another body of literature studies platform development and design for commonality. A principal work is *Meyer & Lehnerd* (1997). An excellent overview of product platform planning is contained in *Robertson & Ulrich* (1998). A related body of literature, arising from the lean methodology set forth in *Womack & Jones* (2003), takes a deeper look at system development to understand how value is defined and created during design and other activities upstream of production. Graduate research in conjunction with MIT's Lean Advancement Initiative (LAI) has produced a set of master's theses addressing lean product development, including *Chase* (2001), *Millard* (2001), and *Slack* (1998).

Organizational structure. This topic considers the study of organizational design and how the grouping and linking of resources into functional groups and product teams influences business performance. A comprehensive analysis of organizational structures in the automotive industry is contained in *Cusumano* (1998), which uses statistical methods to find relationships between organizational structure and development cost and lead-time. Organizations are increasingly being studied formally as systems – so-called enterprise systems – that can be architected in a way similar to products. This approach is introduced in *Rechtin* (2000). An overview of the emerging field of enterprise architecting is given in *Nightingale & Rhodes* (2002). While this thesis equates the enterprise to a single firm, some literature considers an *extended enterprise* comprised of a central firm and its customers, suppliers, labor, and capital. For example, *Piepenbrock* (2008) expands enterprise architecting beyond a firm's internal organizational structure to include the structure of a firm's supplier, labor, and investor relations.

Space systems and aerospace industry. This topic considers both technical information about satellite architecture and design and the industry context in which space systems are developed. A principal reference for satellite development is *Larson & Wertz* (1999), which provides a comprehensive discussion of the end-to-end development process with physics-based design rules and procedures for all major spacecraft subsystems. *Murmann et al.* (2002) contains an insightful discussion of the aerospace industry – historical background and future outlook – that helps to frame the stakeholder environment which drives the evolution of modern satellites.

NGST Internship

With the internship as the primary determinant of the research setting and timetable, the author took advantage of his embedded status within the NGST organization. In order to develop an understanding of the scope and context of the case study, a diversity of information sources were drawn upon over the course of the internship. Each information source and its relevance are described in some detail below.

Product documentation. The major sources of product information were: (1) descriptive documents that capture requirements, specifications, design details (function allocated to physical elements), and analysis results; and (2) drawing packages that capture physical dimensions and configurations, part listings (type and quantity), and fabrication and assembly instructions. These information sources were sufficient to characterize the *product architecture* for a given system, subsystem, or component. In most cases, the document and drawing hierarchies in themselves mirrored the structure of the product. It should be noted that the majority of product documents were generated for a particular government contract. As a result, nearly all documentation for intermediate products (e.g. subsystems and components) was associated with a single contracted end system.

Process documentation. The major sources of process information were: (1) process flow maps and listings that depict interconnections between processes and hierarchies of processes and subprocesses; (2) descriptive documents that define and explain individual processes in terms of their purpose, owners, inputs and outputs, suppliers and customers, sequence of activities, and metrics; and (3) related procedures, work instructions, and tools (e.g. templates and checklists) that support process execution. For the most part, these information sources were sufficient to characterize the intended *process architecture*. Whether this architecture was realized in practice might be indicated by scorecards and other periodic compliance assessments, which were required of all process owners. The process documentation was comprehensive as a result of NGST's process focus since their previous reorganization in 2001.

Six Sigma project documentation. All documentation in support of completed and ongoing process improvement projects (e.g. DMAIC/DMADV projects as part of Six Sigma Green Belt and Black Belt certification) was archived and accessible through a database. For each project,

relevant information included (1) the project charter (i.e. problem statement, business case, project scope), (2) voice of the customer (VOC) data, (3) related technical and financial data, (4) analysis results, and (5) project outcomes in the form of new/revised process descriptions, tools for process execution, and metrics for process control. Several Six Sigma projects of interest addressed key issues related to managing product evolution. The documentation for these projects, especially VOC data, provided insights into the technical and organizational challenges of product management in the context of complex system development. The outcomes of these projects included new processes intended to improve product management, some of which had limited success because of the very challenges that were identified from the outset.

Executive management team communications. In addition to the standard top-down flow of information, the NGST sector president and team of vice presidents used several channels to communicate directly with the entire organization, including (1) sector-wide email and web announcements, (2) panel discussions at sector-wide events, and (3) webcasts of focus group sessions. The announcements and discussion forums regarding the organizational realignment were of particular interest because they captured the executive management team's rationale and expectations for the restructuring in a concise and direct manner, and invoked questions and concerns about strategy and implementation from individuals working in various parts of the organization.

Production facility tours. The author had the opportunity to tour a series of production facilities spread across the Space Park campus, including (1) microelectronics fabrication lines, (2) mechanical component assembly lines, (3) electrical component assembly (manual and automated) lines, (4) subsystem-level tests areas, (5) system-level integration facilities, and (6) system-level test areas. The significance of these tours was to observe first-hand the technical complexity of space systems and to appreciate the challenge of building and sustaining in-house end-to-end development expertise. In other words, reading product and process documentation is by no means a complete substitute for walking assembly lines and handling product hardware and manufacturing equipment.

Personal interviews. By far the most critical information source was interviews with NGST employees from various parts of the organization. Throughout the internship, the author had regular interaction with more than two dozen individuals, primarily Black Belts and Master Black Belts within the Six Sigma program. One-time meetings were conducted, usually on a one-on-one basis, with roughly a dozen other individuals from functional groups such as system engineering, design, and production. These interviews offered first-hand accounts and other anecdotal evidence about how the organization has historically addressed the issues of interest, including concrete examples of more or less successful approaches and outcomes. In addition, these individuals shared their personal thoughts and opinions about competing in the satellite industry and key challenges facing NGST. Given that the majority of individuals were long-time employees from a range of functional backgrounds, the collective set of interviews provided a complete perspective of both the industry and organizational context.

2. Frameworks for Complex System Management

This section presents the relevant concepts related to architectural “fit” and integrates them into a framework for thinking about how to manage product development. While numerous frameworks of this kind have been formalized in the published literature, this version has an implicit bias towards the product architecture, in the sense that the organization and its processes are primarily shaped by the goal of delivering competitive products. The objectives of this section are as follows.

- Establish a general framework for the product life cycle that considers product and process architectures apart from the organization.
- Establish a general framework for product management that overlays organizational structure on the product life cycle.

Throughout this discussion, it is important to establish *operational* or *working definitions* for all key ideas.⁹ Consistent meaning and usage of terminology are essential to fostering a common understanding of these frameworks and their applicability. For this reason, key definitions appear in boldface type and are set apart from the document text. It is the author’s intent to uphold these definitions throughout the remainder of this thesis. In addition, some words are *reserved*, in that they have a specific usage context that should not be confused with more common ones found in the English language. When introduced for the first time in this context, these reserved words appear in italic, boldface type and are in line with the document text. Here again, the author intends to maintain consistent usage of all key terminology. Finally, other key words and phrases appear in italic type to indicate specific terms whose definitions can be inferred from the surrounding discussion. Still other words appear in italic type simply as a matter of emphasis.

⁹ In the spirit of consistent language, it should be noted that the term *operational definition* in itself has a rather precise meaning in the field of philosophy. Most simply, an operational definition of something provides a measurement process for determining whether or not it *exists*. Accordingly, not all ideas and concepts may have an operational definition in the strictest sense. In these cases, the term *working definition* is used, which provides an adequate and unambiguous way to identify something.

2.1. General Viewpoints

Broadly speaking, all things can be classified as either an object or a process. It is debatable whether one category is subordinate to the other, given the tightly coupled nature of the two. While *objects* have a static, tangible (or mental) existence that humans can perceive, *processes* embody patterns of transformation that are carried out *on* objects and *with* objects. Dori suggests that processes are “*carriers* to which we mentally assign *responsibility*” for changing objects, and they “exist” only to the extent that the changed objects are perceived in their past and present forms (Dori 2002, 58). Though abstract on its own, this object-process construct gives rise to two possible viewpoints that are useful in the realm of product development. The *system viewpoint* refers to an object-centric view of the world, which emphasizes the physical objects or humans that carry out processes. The *SIPOC viewpoint* (which makes use of a Six Sigma tool that stands for Suppliers-Inputs-Process-Outputs-Customers) refers to a process-centric view of the world, which emphasizes the transformations that objects undergo.

2.1.1. Systems

To be clear, a system is an object (as used above) which exists for a particular purpose. The literature provides numerous, yet similar, definitions for the term “system.” In most examples, emphasis is on the part-whole relationship. Reichtin suggests this two-part definition (Reichtin 1991, 28):

- A system is a complex set of dissimilar elements or parts so connected or related as to form an organic whole.
- The whole is greater in some sense than the sum of the parts, that is, the system has properties beyond those of the parts. Indeed, the purpose of building systems is to gain those properties.

For the author’s purposes, the term system is used broadly to mean a collection of physical and/or human elements. A *product system* refers primarily to a physical structure that is used by human operators, whereas an *enterprise system* refers mainly to a people structure that uses physical instruments. Therefore, this general definition is adopted:

SYSTEM – a collection of physical objects and/or humans, which are configured together to produce results beyond those achievable when taken as separate elements

A key notion to consider is the *unbounded* nature of systems. That is, “each system is inherently part of a still larger [super]system” (Rechtin 1991, 1). Similarly, each system is divisible into still smaller subsystems until some indivisible, atomic element is reached. Figure 2 illustrates this using an object-process diagram (OPD).¹⁰ The boundary dotted lines are not formally part of the OPD, but depict how an observer’s relative view of a system forms internal interfaces among its subsystems and external interfaces with its supersystem. The supersystem can be viewed as part of the external environment for the system.

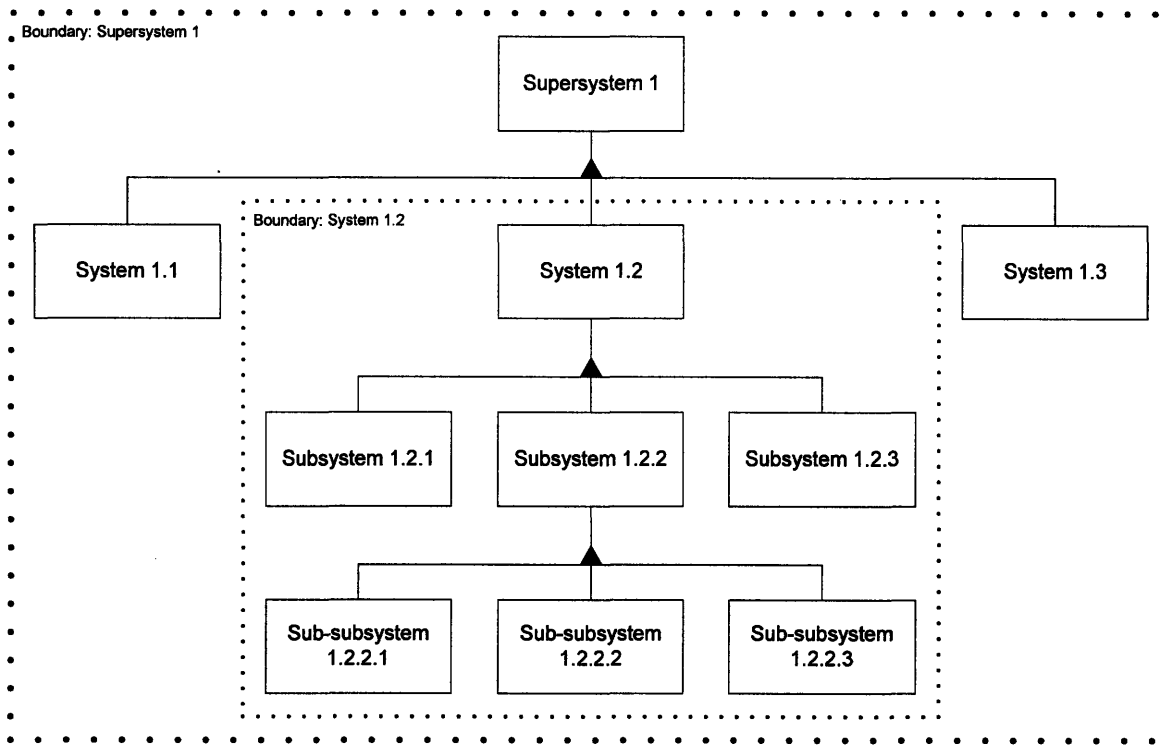


Figure 2: System view

A system exhibits a set of functions that accounts for the purpose of its existence, including the need for its creation and the goal of its behavior. These functions are closely related to the

¹⁰ Appendix A contains a description of the OPD symbology used in figures throughout this thesis, which adheres to the modeling convention set forth in *Dori* (2002).

processes that the system carries out. The *system view* is considered object-centric because a system is more easily described by its elements and the connections between them. The elements of the system are arranged in a particular configuration (the system's structure) with the intent to exhibit certain dynamics (the system's behavior). The combination of structure and intended behavior represents the architecture of the system. Figure 3 summarizes the system construct in an OPD.

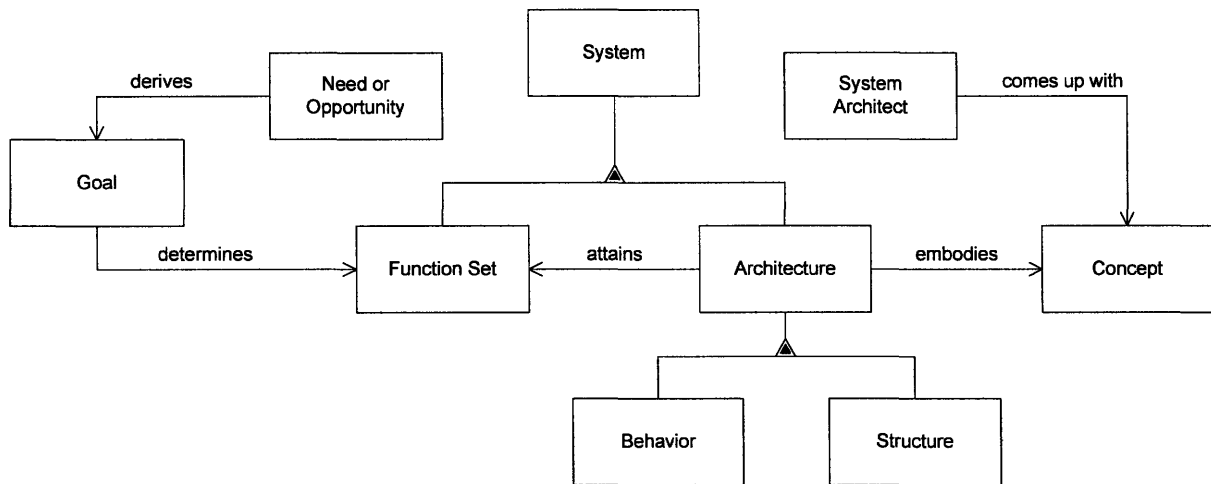


Figure 3: Model of a generic system (Dori 2002, 266)

2.1.2. SIPOCs

As noted above, a process embodies the transformation of one or more objects and does not have a tangible existence of its own. Consequently, processes are effectively inseparable from the objects that they affect. The *SIPOC view* is considered process-centric because the intended pattern of transformation receives greater emphasis than the set of objects that enter and exit the process. Among the *input objects* to a process are those that are transformed, as well as those (human operators or physical instruments) that enable the process. The *output objects* are those that result from the process. When an object is transformed – consumed, yielded, or otherwise affected, it is said to be an *operand* of the process. The SIPOC associates the input and output objects with their suppliers and customers. The SIPOC acronym also implies that the process is under some degree of human control, which is generally assigned to the process owner. In fact, a SIPOC map is constructed from the process owner's point of view. For input objects, there is a

distinction between objects provided externally by suppliers (and acted upon by the process) and objects provided internally by the process owner (and carry out the process). Figure 4 illustrates the SIPOC concept using an object-process diagram (OPD).

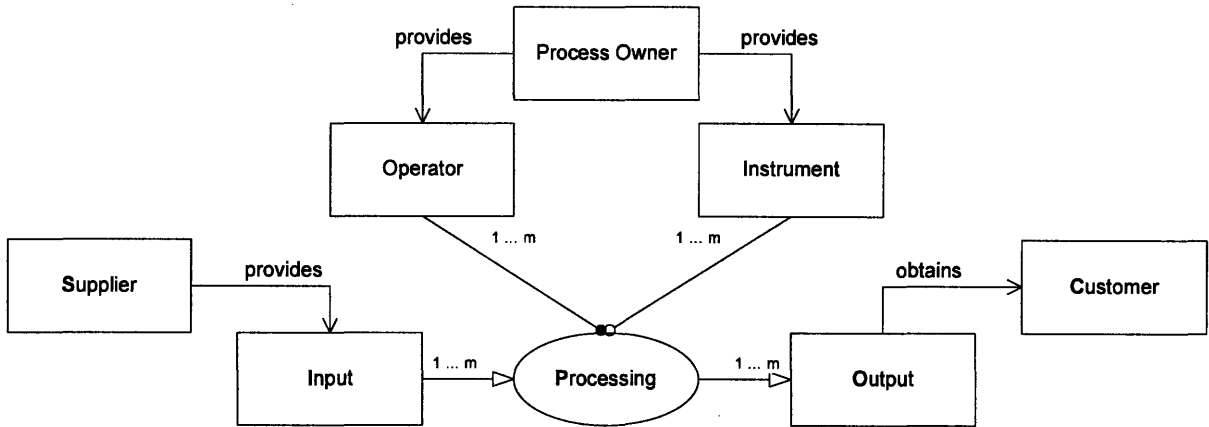


Figure 4: SIPOC view

A key notion to consider is the *infinite* nature of SIPOC chains. Depending on the observer's point of view, an individual or group is simultaneously a customer, process owner, and supplier. That is, the customer from a previous process is the process owner of the current process, and, in turn, the supplier to the next process. Similarly, the output objects from a previous process become input objects to the next process. In this way, a virtually endless string of processes is formed. Figure 5 depicts a SIPOC chain.

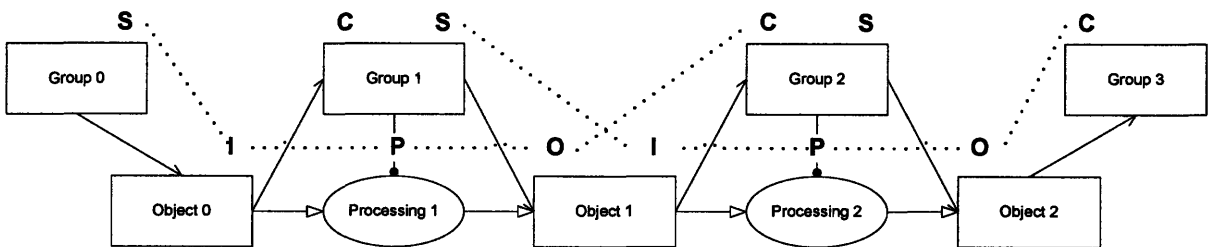


Figure 5: SIPOC chain

The SIPOC chain can grow in two ways: (1) by *extension* when new process links are appended to the beginning or end (upstream or downstream) of the existing chain and (2) by *expansion* when existing process links are spread into a sequence of two or more enclosed subprocesses.

Admittedly, the discussion thus far has been rather abstract. The following subsections build upon these basic constructs to formalize the interconnections between products, processes, and organizations.

2.2. Product Systems

Having already used the term “product” extensively throughout this document, it is long overdue to provide an operational definition for it. As mentioned, a product is a system and therefore exhibits all of the properties inherent to a system. For the author’s purposes, what distinguishes a *product system* from a generic system is that a product is created with the intent of selling it for profit. Accordingly, a product has a tangible existence (either physical or informational) and presumably provides value – benefit at cost – to one or more beneficiaries (a generalization of customers). Therefore, this definition is adopted:

PRODUCT – a physical or informational object that provides value to one or more beneficiaries, created by a business with the intent of selling it for profit to these beneficiaries

Figure 6 summarizes the object-system-product hierarchy, which is a fundamental model developed in *Dori* (2002).

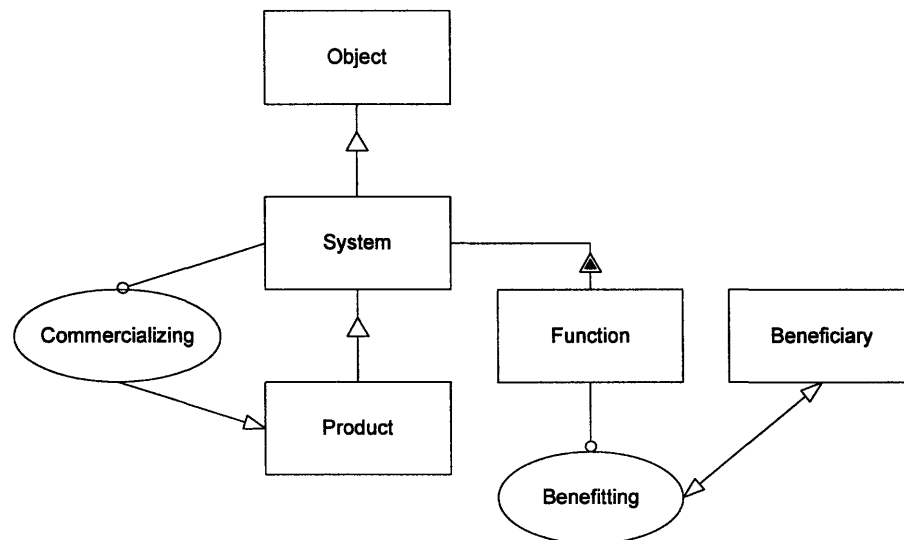


Figure 6: Object-system-product hierarchy (Dori 2002, 266)

2.2.1. Form and Function

A product system is most often described with respect to its *form* and *function*. The literature for system architecture provides many similar definitions for these ideas. Crawley offers the following pair of definitions (Crawley 2007):¹¹

- Form encompasses the *elements* (disaggregated physical and informational pieces) of a product and the *structure* (physical or virtual connections) among these elements.
- Function encompasses the *processes* exhibited by the product with explicit identification of the *operands* (input and output objects) that undergo change during these processes.

Several thoughts related to form and function are worthy of mention here. To start, note that a product's form is readily depicted using the system view as shown in Figure 2. This so-called *decompositional view*, which is shown generically in Figure 7, lists the elements of form in a hierarchy of part-whole relationships. Note that *parts* are considered the lowest level in the hierarchy because they are indivisible, atomic elements; whereas *material* is inherently a continuum of matter until shaped into a discrete part. While useful on its own, this view does not capture the connectivity between elements, which occurs at internal and external interfaces arising from the chosen system boundary. Therefore, the *structural view*, which is best represented in an engineering drawing, provides the remaining information about a product's form, such as spatial configuration and assembly instructions.

The decompositional and structural views become increasingly difficult to represent as the number of elements grows larger. One aspect of product *complexity* is the sheer number of elements and/or number of connections between elements. A first-order measure of complexity, as suggested by Crawley, is to count the number of elements. Different products might be stratified into levels of complexity by making use of the 7 ± 2 rule.¹² That is, simple products consist of one to nine ($\sim(7 \pm 2)^1$) elements, moderately complex products of nine to 81 ($\sim(7 \pm 2)^2$) elements, and highly complex products of significantly greater than 81 ($\sim(7 \pm 2)^3$) elements.

¹¹ Crawley (2007) refers to the collection of lecture slides and notes from his *System Architecture* course at MIT. All references to Crawley draw upon this source, however, specific lecture and slide numbers are not cited.

¹² This rule refers to a supposed limit, proposed by psychologist George Miller, on the number of information pieces that an average human can reliably transmit using immediate memory (Miller 1956).

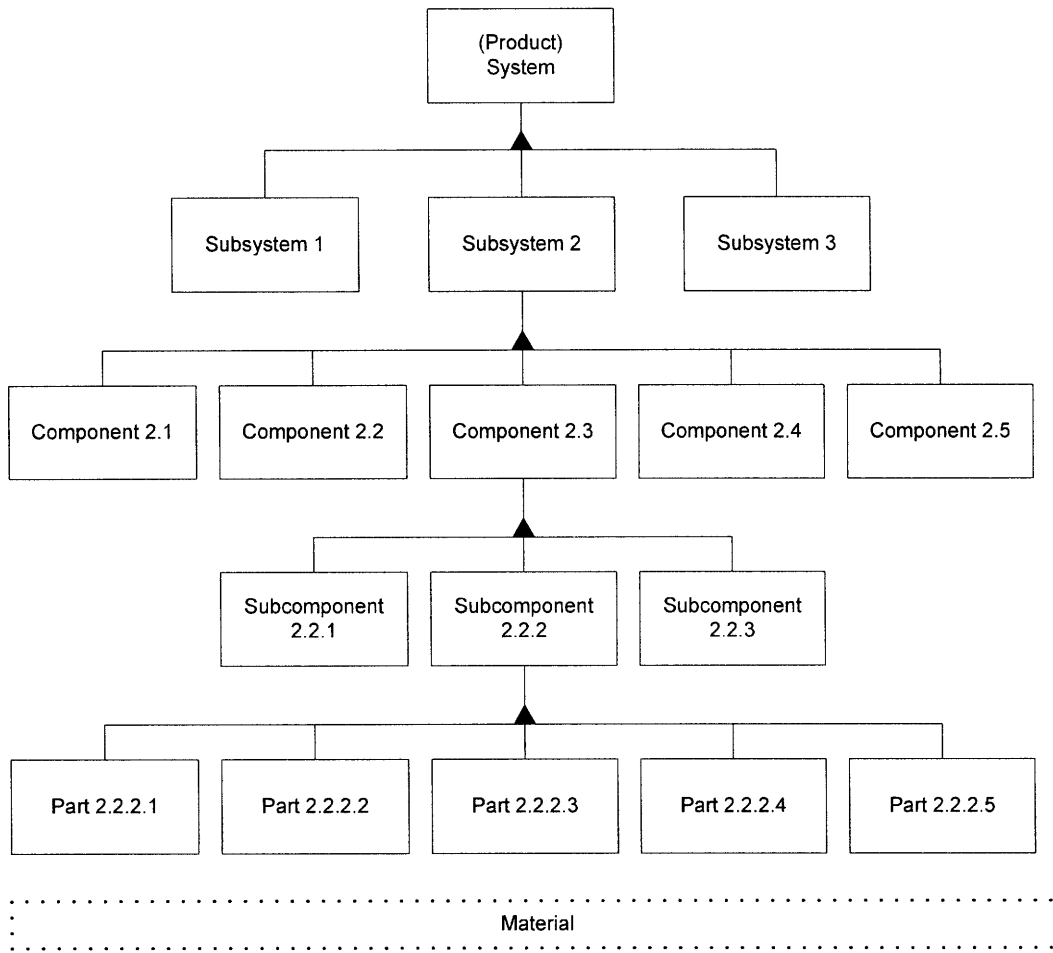


Figure 7: Decompositional view of a product's form

A product's function does not lend itself to a strict hierarchical view of processes and subprocesses. Whereas smaller elements of form simply aggregate into larger ones, Crawley suggests that lower-level functions combine in a "nonlinear" way to make higher-level functions emerge. Indeed, the notion of "a whole greater than the sum of its parts" relates to the *emergent* property of system-level functions. In contrast to the hierarchy in Figure 7, Figure 8 illustrates the rather unstructured composition of processes, where zooming in (or out) reveals (or hides) embedded subprocesses that have a "loose" part-whole relationship. It can still be said that higher-level functions *decompose* into lower-level ones, but functional decomposition and physical decomposition create hierarchies of a slightly different nature.

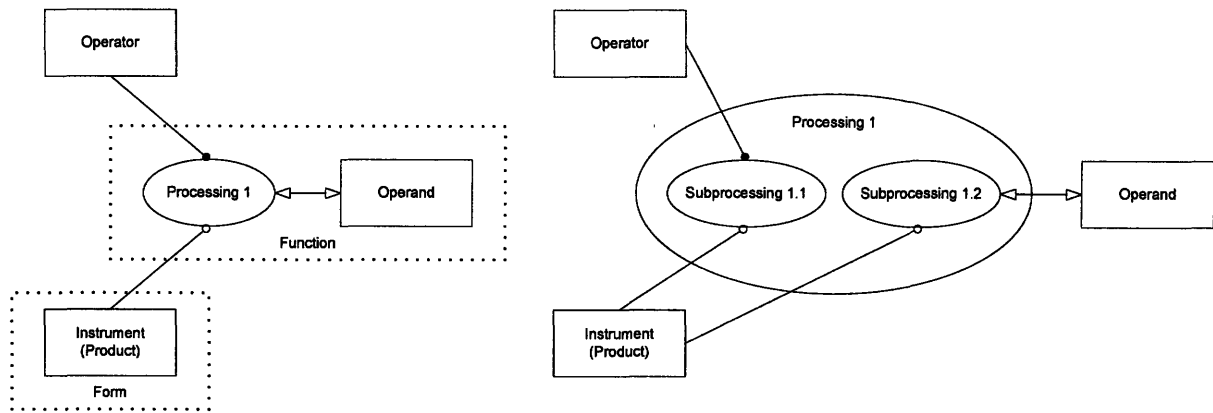


Figure 8: View of a product's function

2.2.2. Product Architecture

Another key term that remains undefined is “architecture.” Previously, the discussion has implied that architecture is synonymous with structure. This is a rather imprecise definition. More formally, architecture embodies the relationship between form and function, a mapping between objects and the processes they enable. For a product system, this definition is adopted:

PRODUCT ARCHITECTURE – the allocation of processes to one or more elements of a product's form, which, by virtue of the elements' structure, enables a product's function to emerge

The mapping of function to form is guided by a *concept*, which provides a specific solution in response to a need or opportunity. Each concept generates a different product architecture, which achieves its function through particular behavior exhibited by a particular structure of elements. In summary, Figure 9 tailors the previously shown model of a generic system for a product system.

Another aspect of product *complexity* is the nature of the function to form mapping embodied in the product architecture. Architectures that contain mostly one-to-one relationships between objects and processes are said to be *modular*, and are characterized by relatively straightforward connections between elements and a predictable combining of lower-level functions. Those that contain several one-to-many relationships between objects and processes (or vice versa) are said to be *integral*, and are characterized by intricate connections and especially emergent higher-

level functions. Accordingly, modularity implies architectural simplicity and integrality implies architectural complexity, irrespective of the number of elements that compose the product.

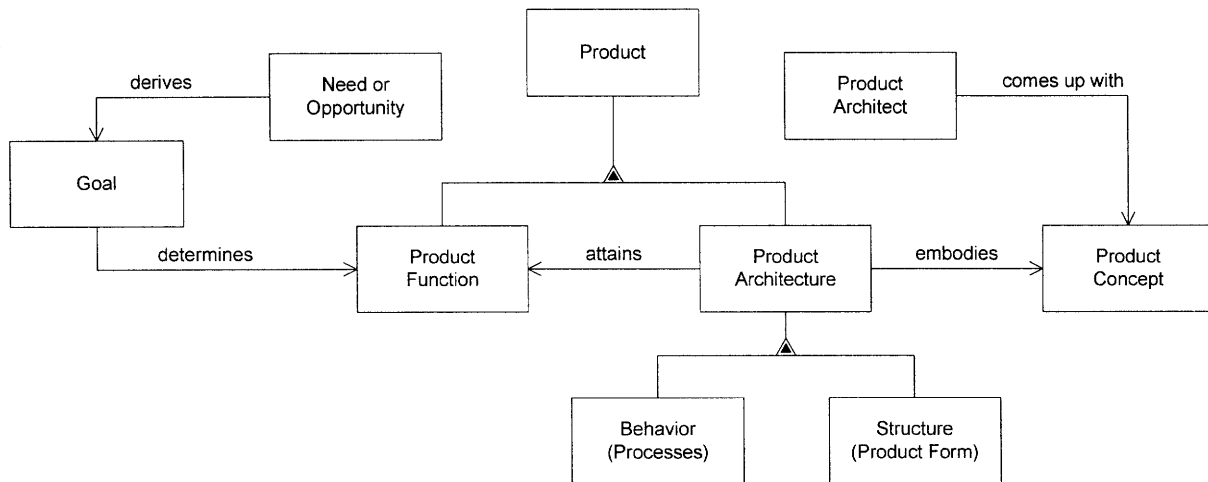


Figure 9: Model of a product system

2.3. Process Streams

As previously described, processes are inseparable from the objects they change. For any given product, one can distinguish between (1) those processes that affect the product as it is brought into existence and (2) those processes that affect the product as it provides value to its beneficiaries. Using the SIPOC view, there is a *process stream* that denotes a set of activities performed in some order over time. Equivalently, there is a *product life cycle* that denotes a set of states transitioned in some order over time. The process stream and product life cycle do not refer to the processes *enabled by* the product (i.e. product viewed as an instrument), but rather those processes *acting on* the product (i.e. product viewed as an operand). This distinction is especially subtle when the product is actually operating – simultaneously an instrument performing its intended function and an operand in the sense that human operators are changing it between states of “in use” and “out of use.”

The complete end-to-end process stream or product life cycle progresses a product from its conception into mental existence, to its production into physical existence, to its disposal into physical (and perhaps mental) oblivion. A distinct point along the process stream is that moment when the product attains complete physical existence, having the potential to perform its

intended function. This point coincides with when the product changes from output to input, possession changes from supplier to customer, and most importantly, selling occurs to the beneficiary. The stretch of the process stream prior to this point consists of *developmental processes*, and the stretch following this point consists of *operational processes*. The set of developmental processes is closely related to the notion of a *value stream*. Womack defines the product value stream as “the set of all the specific actions required to bring a specific product... through the three critical management tasks of any business” (Womack 2003, 19). He goes on to identify the three tasks as (1) problem-solving (from concept to design), (2) information management (from order to delivery), and (3) physical transformation (from raw materials to finished product). For the author’s purposes, the first and third of these are most critical to the process stream.¹³

Figure 10 depicts a generic representation of the process stream. The set of processes form a flow in the center of the diagram. Note the division (dotted line boxes) of the process stream between developmental and operational processes. Above the process flow, the product life cycle is shown as a series of states that the product transitions through. Alternatively, below the process flow, the product life cycle is shown as a series of input and output objects that equate to the product’s existence at each point along the process stream. Either representation of the product life cycle is correct, however, the bottom one is more instructive when considering that the product moves between process owners (e.g. designers, builders) along a SIPOC chain.

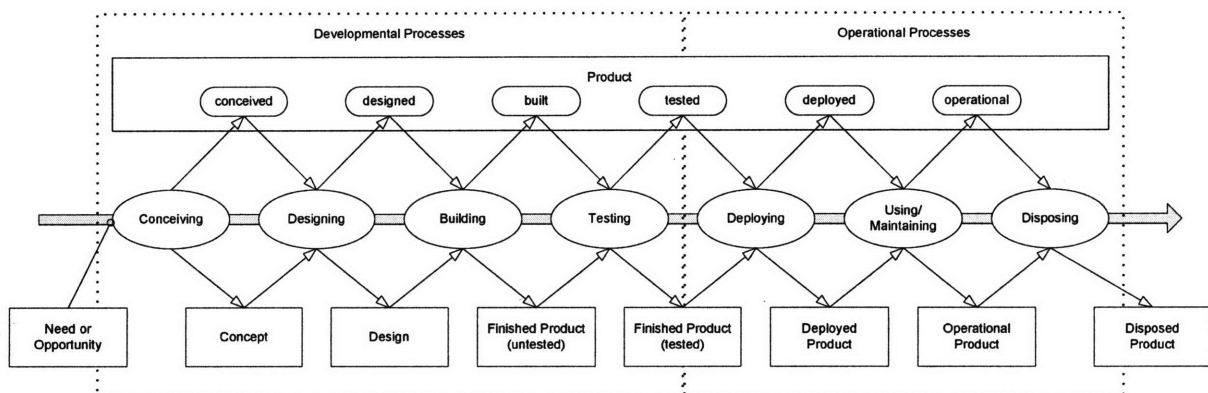


Figure 10: Process stream (product life cycle)

¹³ While important, *order to delivery* is a transitional stage between the developmental and operational stretches of the process stream, which encompasses the selling transaction and transfer of possession from supplier to customer.

Similar to the processes associated with a product’s function, the process stream can also be zoomed in (or out) to reveal (or hide) embedded subprocesses that act on the product. Figure 11 illustrates several examples of this in conjunction with the process stream from Figure 10. Zooming in or out is equivalent to expanding or contracting the SIPOC chain. For example, as shown, by zooming in the *building* process to reveal the fabricating, assembling, and integrating subprocesses, the SIPOC chain is expanded to reveal several stages of work-in-process (WIP), which are previously embedded input and output objects.

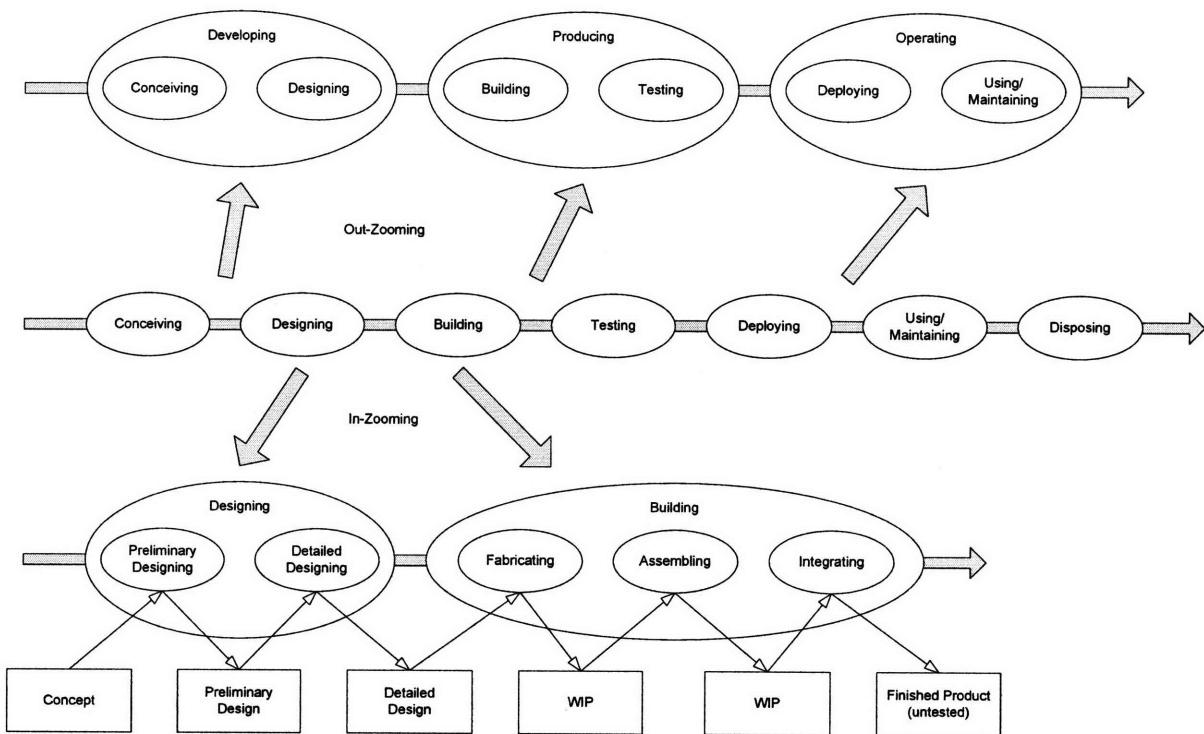


Figure 11: Zooming in and out of the process stream

2.3.1. Product Essence

The product life cycle implies that a product “exists” from the moment it is conceived, even if no tangible object is present. The notion of *essence* suggests that objects are either **physical** or **informational**. Crawley speaks of a *duality* between the physical and informational forms, in that (1) all physical objects *can* be captured in informational form (e.g. blueprint of Sears Tower) and (2) all informational objects *must* be captured in physical form (e.g. data stored on a hard disk). Throughout its life cycle, a product transitions from a series of informational forms (e.g.

concept, design) to a series of physical forms (e.g. prototype, work-in-process), culminating as a finished product – its complete physical existence. Each informational form has a corresponding physical object that “stores” the product (e.g. concept sketch or engineering drawing on paper). Once the product progresses to a finished product, an associated informational form is possible, but not necessary (e.g. designs are usually archived in some way). The intermediate stages of a product (e.g. work-in-process) effectively have a mixed essence in that some proportion exists in physical form – partially-built objects – and the rest exists in informational form – design elements yet to be fabricated. Figure 12 depicts the product life cycle with the physical and informational forms at each stage. A notional “essence bar” is drawn along the product life cycle to suggest the proportion of the product that is informational versus physical. Early on, the product is completely informational. After the building process, the product is mostly, if not all, physical – depending on whether or not information storage media is an element of its form (note that a software product might be considered completely informational in its end state).

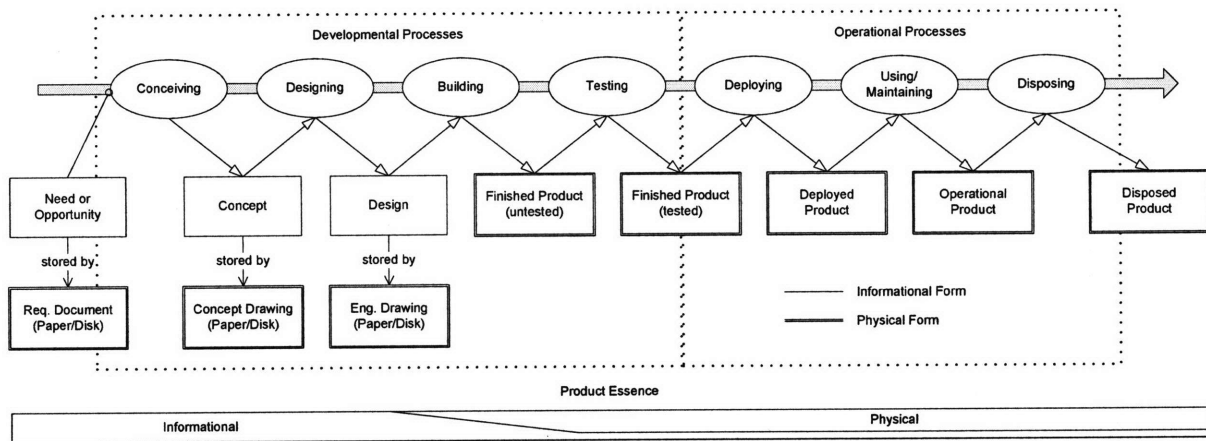


Figure 12: Process stream with consideration for product essence

During the early stages of the product life cycle, it is important to make the distinction between the product’s mental existence, for example, as a design, and the physical form that stores the product in this state. An engineering drawing in itself is *not* the product, recalling that a product is defined as an object to be sold. In most cases, a drawing is not sold and does not directly provide value to the beneficiary. A drawing simply stores the actual product at a time when the product exists in an informational form. Of course, when considering the SIPOC chain, a drawing is an output object received by a customer, say the builder. From his perspective, a

drawing might appear to be a product of his supplier, say the designer. However, in keeping a strict definition of product, a drawing is at most a *work product* that arises as the actual product is brought into complete physical existence.

2.3.2. Process Value

Another important consideration when discussing the process stream is the *value* associated with each process along the flow. A simple though useful definition of value is *benefit provided at cost*. *Benefit* rests in the product's function, in that function satisfies a need or opportunity identified by the beneficiary. *Cost* rests in the product's form, in that physical or informational elements are designed and built through the expenditure of material, energy, and time. Applying this definition along the process stream, the distinct point is again that moment when the product attains its complete physical existence. It is at this stage that the beneficiary can obtain the product. The selling transaction solidifies the notion of benefit at cost. That is, the beneficiary gains access to the product's intended function by making a payment in proportion to the material, energy, and time expended by the seller. The stretch of the process stream prior to the selling transaction consists of *value creation* activities conducted by the seller, and the stretch following this point consists of *value delivery* activities realized by the beneficiary. Womack's value stream refers to the value creation activities within the control of the business that sells the product. His lean methodology focuses on the identification of value (benefit at cost) and waste (cost at no benefit). Lean is most often associated with the elimination of waste. Using the language above, waste denotes expended material, energy, and time for which the beneficiary should not owe payment because no additional benefit is realized in return. However, waste elimination represents a rather limited view of the lean methodology. To dispel this narrow approach to implementing lean, Murmann et al. reiterate that "lean is a process of eliminating waste with the goal of creating value" (Murmann 2002, 6).

Accordingly, the process stream should be analyzed with respect to what and how value is created during each process. During the *producing* process (i.e. fabricating, assembling, integrating from Figure 11), value rests in the physical form of the product as it is brought into complete physical existence. Value creation occurs by changing (e.g. shaping, positioning) material to yield the elements and structure specified by the product's form. The traditional principles of *lean production* concentrate on how to develop and execute manufacturing

processes that minimize the material, energy, and time required to transform raw material into finished product. During the *developing* process (i.e. conceiving, designing from Figure 11), value rests in the informational form of the product as it is brought into complete mental existence. Value creation occurs by generating sufficient information about the product such that the detailed design and build-to package meets all requirements and specifications. This information includes not only the product’s concept, architecture, function, and form, but also the test and evaluation results required to increase the *certainty* of meeting all requirements and specifications. In this way, *lean product development* concentrates on how to develop and execute design processes that minimize the energy and time required to yield a final design with acceptable performance risk. Browning explains that in lean product development, “progress is made and value is added by creating useful information that reduces uncertainty and/or ambiguity... the purpose of these activities [such as measurement, analysis, review, and test] is to increase certainty about the ability of the design to meet requirements. That is, these activities decrease performance uncertainty and risk” (Browning 2002, 444). In summary, Figure 13 depicts the product life cycle with a notional “value bar” to identify the source of value along the process stream.

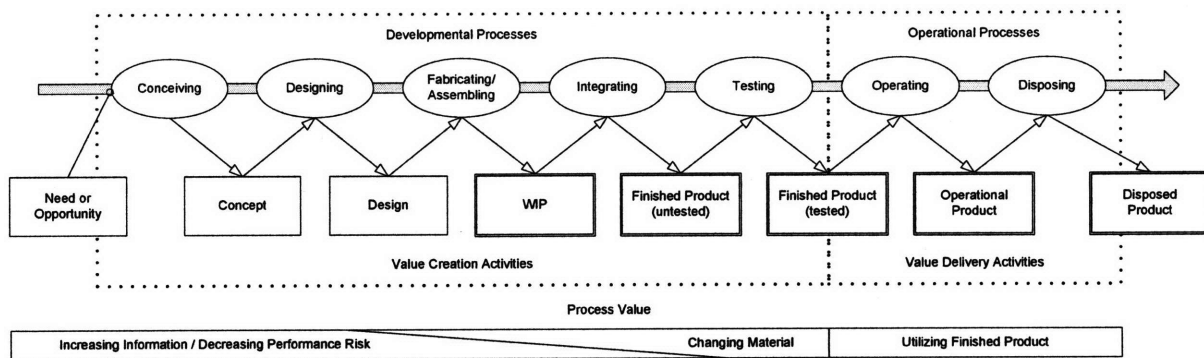


Figure 13: Process stream with consideration for process value

Though not captured in the sequential depiction of the process stream, the conceiving and designing processes are inherently *iterative*, even when, for example, designing is expanded into several subprocesses (e.g. preliminary and detailed designing). Iterative activities still create value because discovering concepts and designs that do *not* meet requirements is as much a part of decreasing performance risk as ultimately converging on a final design that does.

2.4. Integrated Product and Process Framework

To this point, the discussion has considered the relationship between product and process at a rather basic level.¹⁴ A complete framework should address how the process stream applies in the context of *complex product systems*. Recall that product complexity can be thought about with respect to form (e.g. number of elements and connections) and architecture (e.g. modular versus integral function to form mapping). To some extent, these contributions to complexity grow in opposition to each other. For example, an integral architecture (higher complexity) allocates multiple functions to a single physical element, thus minimizing the number of parts. In contrast, a modular architecture (lower complexity) may call for separate physical elements in order to maintain one-to-one object-process relationships. Rather than quantifying the transition from simple to complex, the author considers a *complex product system* to be any finished product with an indisputably large number of elements and connections (e.g. hundreds or thousands) and one or more subsystems with an integral architecture. Many classes of transportation systems, to include automobiles, aircraft, and ships, fit this description. Each consists of thousands of parts and contains integral subsystems, such as engines, electronics (avionics) boxes, and structural frames. As will be discussed in the case study, satellites also satisfy these criteria.

In the context of complex product systems, the process stream can be viewed as a repeatedly observed pattern of activity that is nested with respect to the decomposed product system. This pattern is observed at the subsystem and component levels, just as it is at the system level. Each physical or informational element, considered separately, must progress through all processes along the flow. The process stream is nested because of the hierarchical nature of product development and production. In his formulation of the axiomatic design methodology, Suh suggests that design occurs between two domains – the functional and physical spaces (Suh 1990, 26). Using his terminology, *functional requirements* (FRs) reside in the functional domain and establish *what* is to be achieved via a solution-neutral statement; *design parameters* (DPs) reside in the physical domain and determine *how* it is to be achieved via a physical solution. The design process entails decomposing system-level FRs and DPs into a sufficient level of detail to yield a build-to package. A key principle of axiomatic design is *zigzagging*, which holds that an

¹⁴ Though referred to in the first section of this thesis, the term *process architecture* is not formally defined in this framework. *Architecture* is reserved for systems of objects rather than sets of processes.

FR cannot be decomposed without first specifying the DP that satisfies it. That is, determining what should be done at lower levels requires knowing how it will be done at higher levels. In this way, there is a zigzagging between the functional and physical domains, which is depicted in Figure 14, as the product's architecture and design are elaborated in sufficient detail.

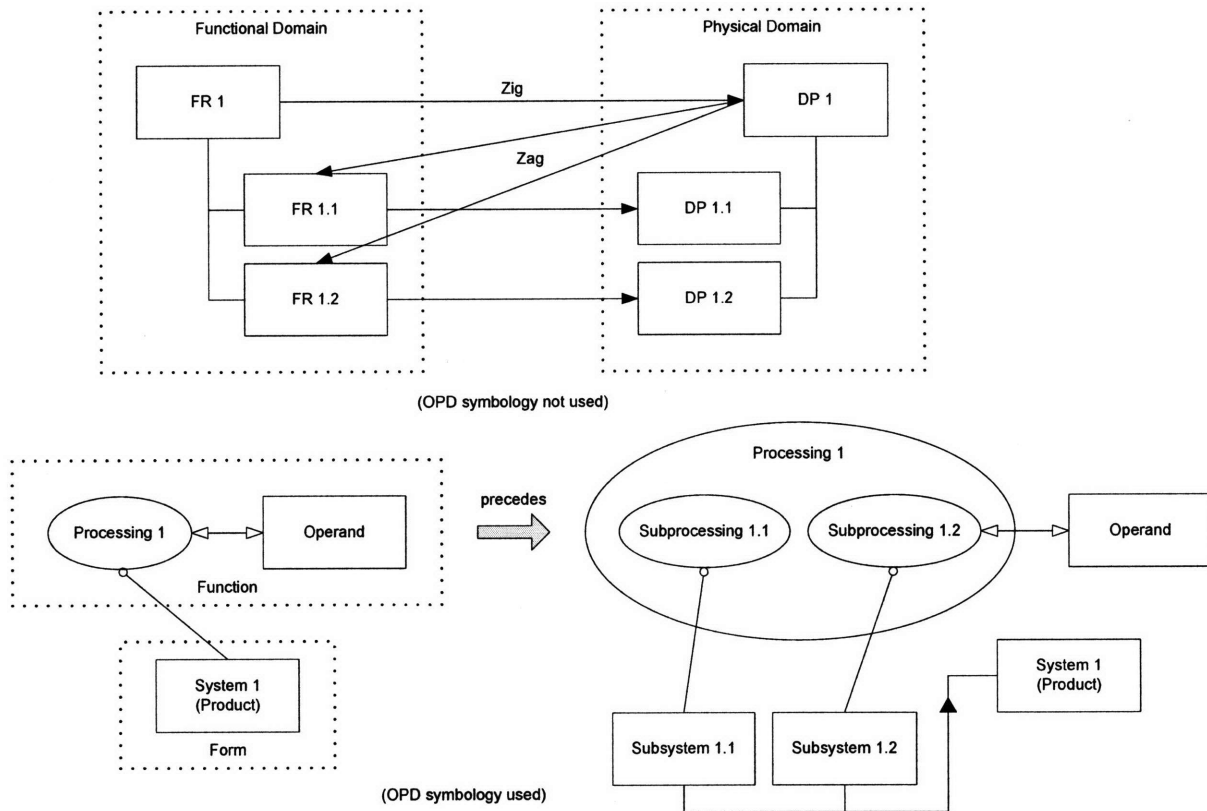


Figure 14: Zigzagging principle of axiomatic design methodology

An integrated product and process framework recognizes the nesting of the process stream as the product system is decomposed into subsystems, components, and so on. Figure 15 shows the complete framework envisioned from the above discussion. The two principal constituents are (1) the product system (left side), represented using the decompositional view and (2) the process stream (right side), abridged to include only the developmental processes. As shown, the process stream repeats at each level in a nested fashion. At the upstream end, the lower level elements cannot begin the conceiving process until the high level elements have been conceived (per the zigzagging principle). At the downstream end, the higher level elements cannot begin the integrating process until the lower level elements have been built.

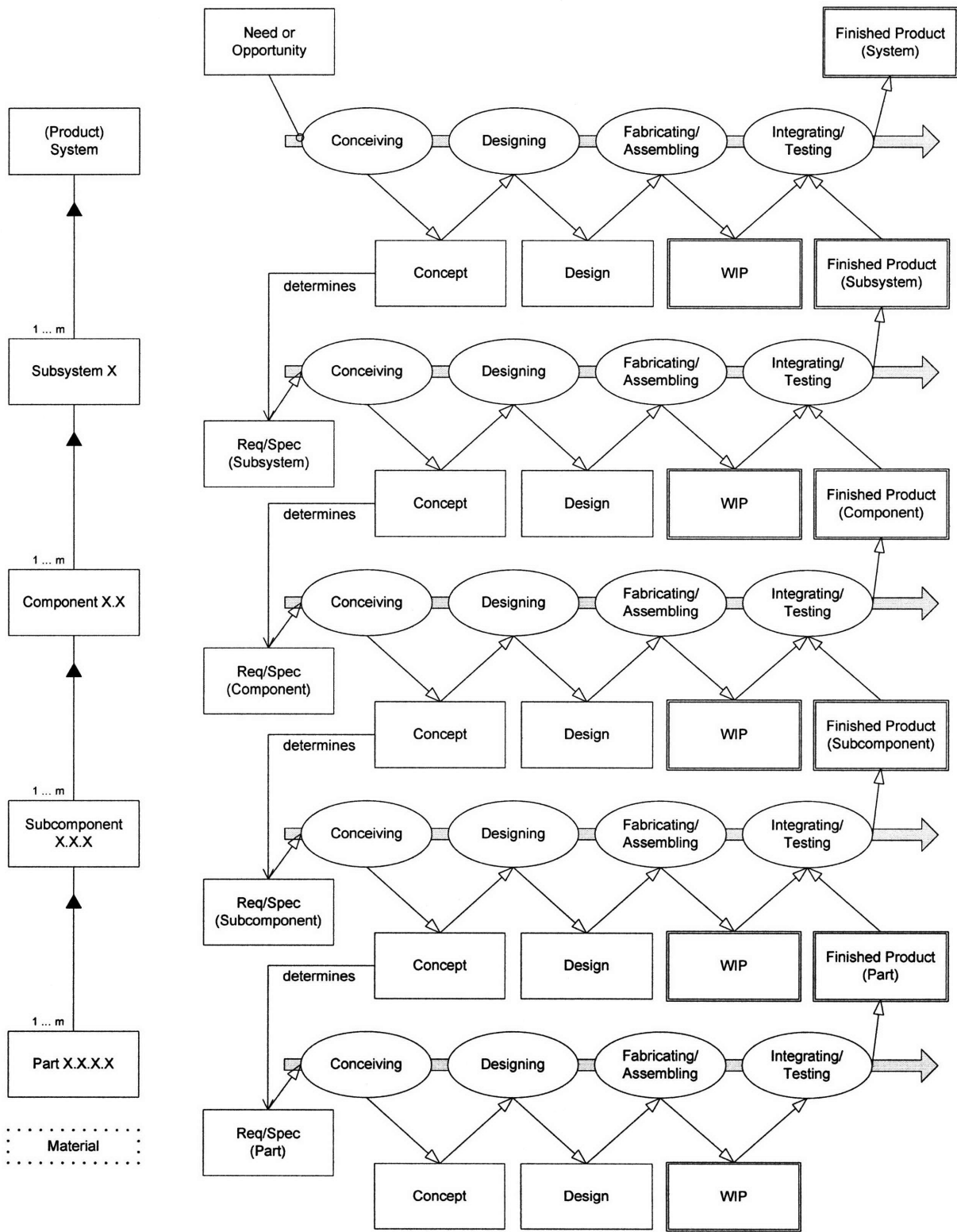


Figure 15: Integrated product and process framework for complex product systems

2.5. Enterprise Systems

For the most part, the above discussion has presented a product and process framework without describing the people who are responsible for implementing it. There have been general references made to suppliers, customers, process owners, and the business that sells the product, but no formal treatment of their purpose, structure, and behavior. Here, a discussion of enterprises introduces people into the framework. Rouse defines the term “enterprise” as a “goal-oriented organization of resources – human, information, financial, and physical – and activities, usually of significant operational scope, complication, risk, and duration” (Rouse 2005, 138). Though an enterprise encompasses more than the people who belong to it, *human* resources are clearly its focal point and lifeblood. For the author’s purposes, the discussion of enterprises is limited to businesses organizations that sell products. Such organizations form around a value proposition to address market needs with their products. Therefore, this definition is adopted:

**ENTERPRISE – an organization guided by a common purpose, namely, to
create one or more products with the intent of selling them for profit to
individuals or other organizations**

Management scientists are increasingly choosing to study businesses and other organizations as *enterprise systems*, just as engineers use a systems approach to study the products that these organizations provide. Accordingly, an enterprise can be shown to exhibit all of the properties that are inherent to any system.

2.5.1. Form, Function, and Architecture

An enterprise system can be described with respect to its form and function, where these terms have meanings similar to when applied to product systems. Indeed, an organizational chart is equivalent to the decompositional view of a product’s form. In this case, the enterprise system disaggregates into a hierarchy of elements (resource units), for example, divisions, subdivisions, and so on. Organizational processes are analogous to the processes associated with a product’s function. Nightingale suggests three categories of processes exhibited by enterprises: (1) life cycle processes, (2) enabling infrastructure processes, and (3) enterprise leadership processes (Nightingale 2004, 5). The *life cycle processes* are the very same ones that compose the process

stream discussed above, where the operands are the product systems created by the enterprise. The remaining two process categories include those activities required to support the life cycle processes and self-sustain the enterprise as a whole. In effect, the operands here are various parts of the enterprise itself. Figure 16 depicts the use of form and function to describe enterprises.

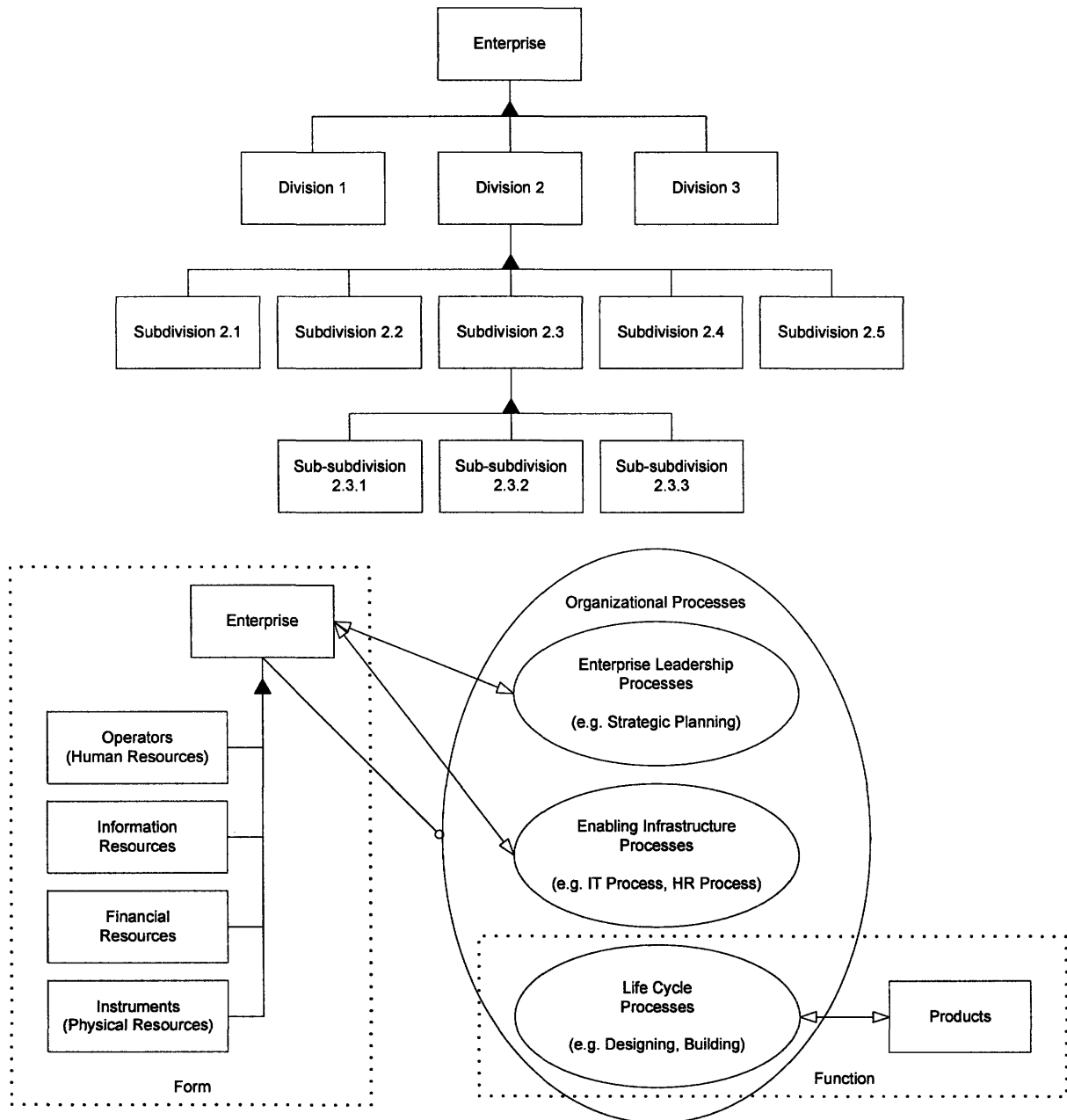


Figure 16: Views of an enterprise's form and function

For the author’s purposes, the focus lies on the life cycle processes, and so only these are considered in this document. However, make no mistake that the entire set of organizational processes is vital to the operation and sustainment of the enterprise system.

It follows that an enterprise also has an architecture associated with it – a mapping between resource units (human, as well as information, financial, physical resources) and the processes they enable. For an enterprise system, this definition is adopted:

ENTERPRISE ARCHITECTURE – the allocation of organizational processes to one or more elements of an enterprise, which, by virtue of its organizational structure, enables the enterprise to create products and realize its value proposition in a self-sustaining manner

Figure 17 summarizes the model of an enterprise system using the OPD construct seen before.

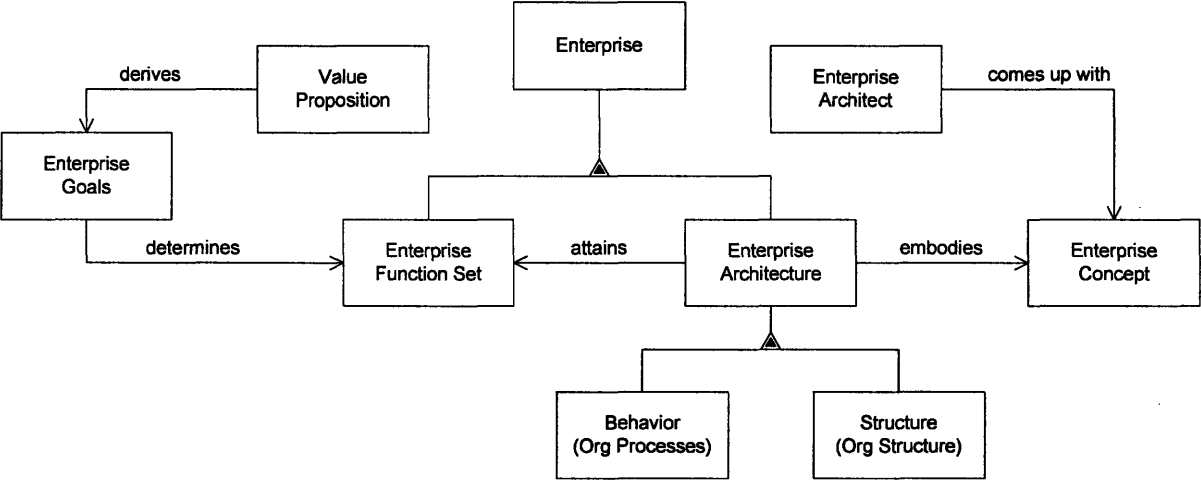


Figure 17: Model of an enterprise system

2.5.2. Product Management

The enterprise architecture assigns responsibility for each of the life cycle processes to at least one element of the organization. The elements acquire or develop the resources required to carry out their assigned processes, and the elements’ leaders (e.g. division or subdivision managers) are often designated as the process owners. In the most generic of architectures, the designing process might be assigned to the design division, the building process to the build division, and

so on. In practice, the organization often separates into traditional functional disciplines, such as the engineering division or production division. The framework presented here is not meant to prescribe a particular nomenclature for organizational structures. Rather, the intent is to illustrate, in general, the interconnections between the product and enterprise architectures. An important functional group that does not explicitly follow from the process stream is the *product management* function. Whereas other functional groups take responsibility for distinct processes along the value stream, product management serves as a controlling function for the entire product life cycle. While other functional groups are primarily concerned with the technical performance, cost, and schedule contributions of their own processes, product management oversees these parameters with respect to the entire product life cycle, to ensure that a reliable solution progresses from concept to finished product within the overall budget and time constraints. Figure 18 depicts a generic organizational structure overlaid on the product life cycle. As shown, each functional group enables an individual process along the flow. For example, the engineering division enables the designing process. The product management function enables an overarching process, which is shown to affect the product at all stages. For clarity, procedural links between processes and product states are omitted (see Figure 10).

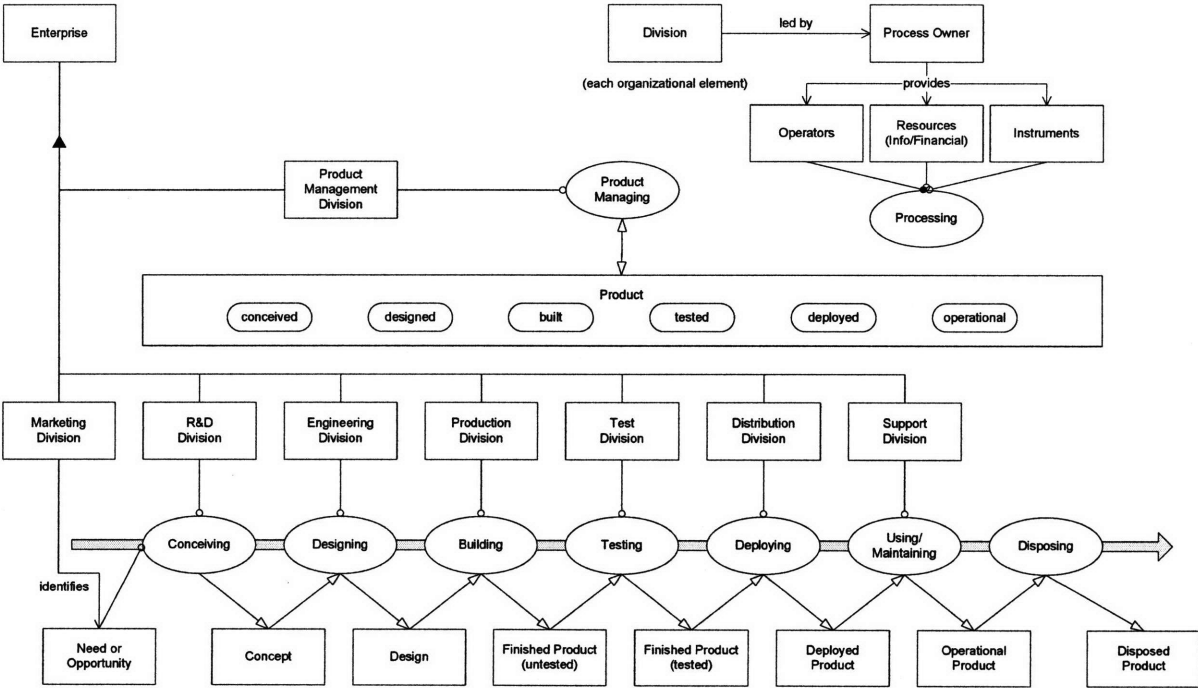


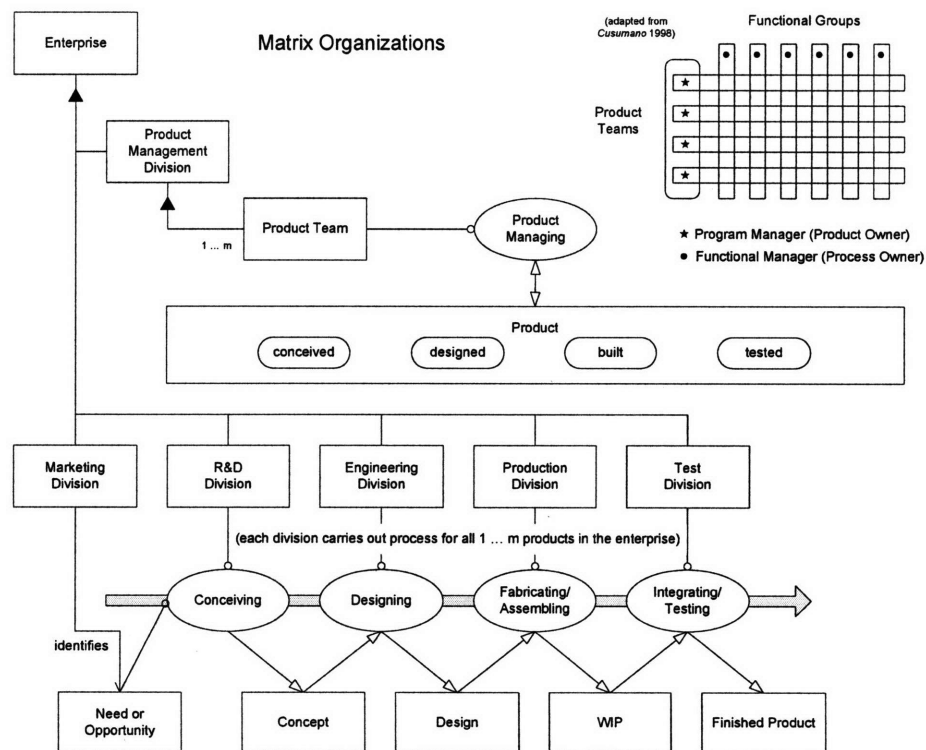
Figure 18: Integration of product and enterprise architectures

2.6. Integrated Product Management Framework

At this point in the discussion, all of the key ideas and their definitions have been introduced. The basic relationships between products, processes, and organizations have been described in words and illustrated with OPDs. Still, a more complete framework handles the case of an enterprise system that creates multiple *complex product systems*. Similar to the discussion in subsection 2.4, where the process stream was used as a repeated pattern with respect to the decomposed product system (see Figure 15), variations of the basic enterprise architecture from Figure 18 can be repeated to extend the framework for complex product systems.

2.6.1. Common Enterprise Architectures

To start, it is useful to consider several types of enterprise architectures that are commonly found in industry. Cusumano identifies four major types, which have been observed in the automotive industry (and occur similarly elsewhere): (1) matrix organizations, (2) product team organizations, (3) center organizations, and (4) semi-center organizations (Cusumano 1998, 53). Figure 19 depicts the first three using an OPD and a diagram adapted from Cusumano (1998).



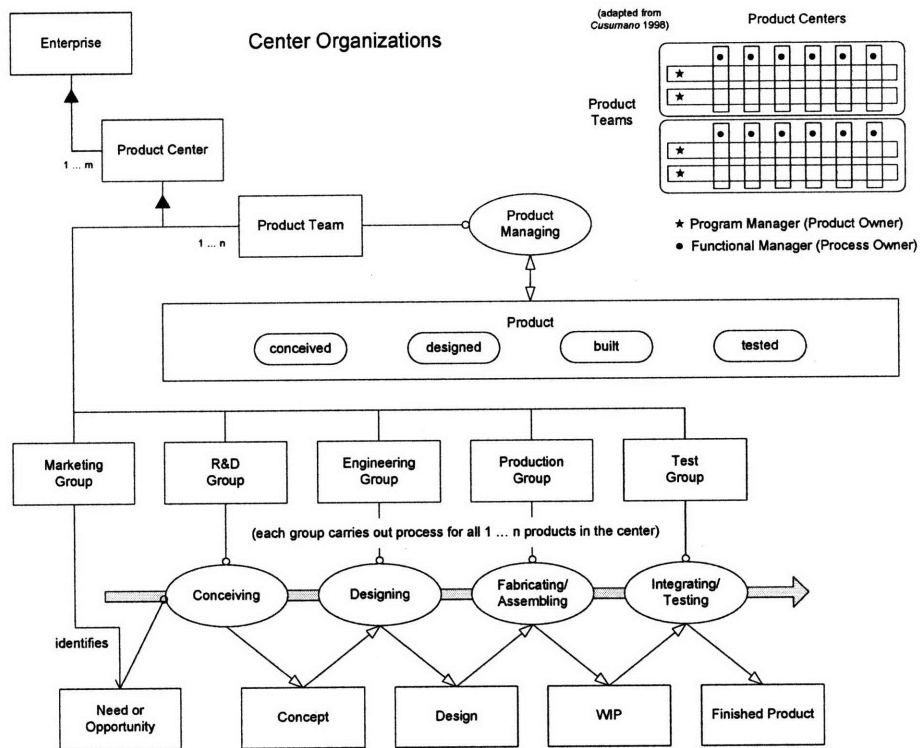
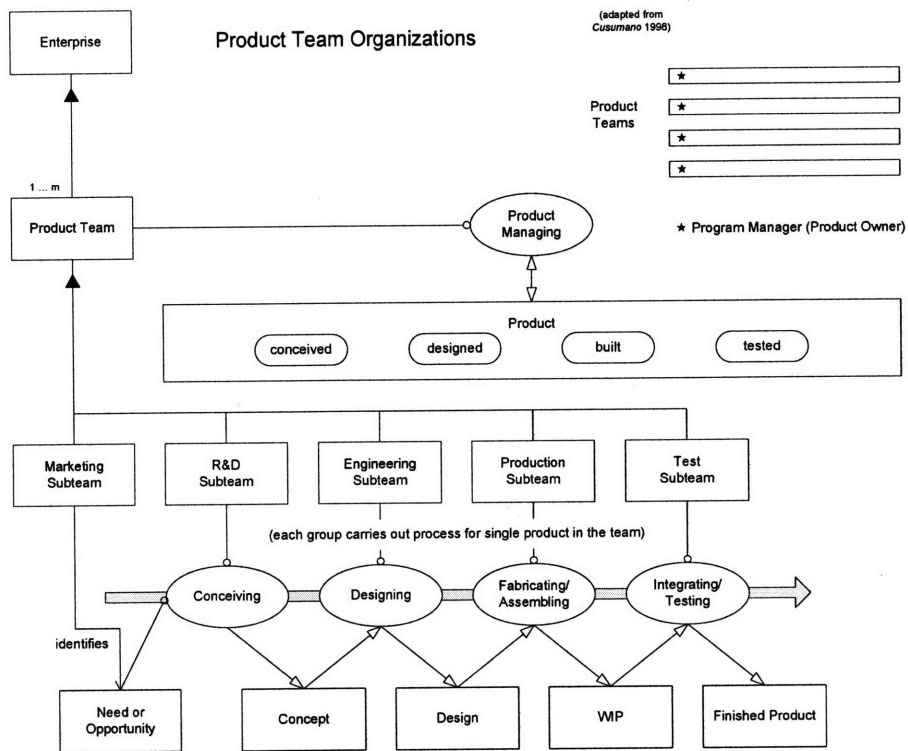


Figure 19: Common enterprise architectures – matrix, product team, and center organizations

In general, the primary difference between these architectures is if and how one or more functional disciplines are duplicated within the organizational structure. The architecting question at hand is whether functional groups should be multi-tasked with the development of all products, or cross-functional teams should be dedicated to the development of individual products. The key is striking a balance between deep functional expertise (inherent to single disciplinary groups) and integrated product solutions (arising from multi-disciplinary teams). Matrix and center organizations provide some degree of balance within their structures. In practice, one of the two modes – either functional managers (i.e. process owners) or program managers (i.e. product owners) – may tend to dominate the interaction.

2.6.2. Products Relative to the Enterprise

The enterprise architectures shown thus far have referred generally to the product in its aggregated form. More precisely, complex product systems are disaggregated into subsystems and components, and functional subgroups are assigned responsibility for life cycle processes in the creation of these lower level products. As implied in Figure 15, subsystems and components also progress through the process stream, and so it is expected that the enterprise architecture supports their hierarchical development and production. With this in mind, it is useful to distinguish hierarchical products relative to the enterprise. Here, two pairs of qualifiers are suggested. First, *end* and *intermediate* denote whether a product is fully or partially aggregated with respect to what the enterprise actually sells. Second, *external* and *internal* denote whether or not a product crosses the boundary of an enterprise system. An *end product* is necessarily an *external product* because it is fully aggregated and sold to a customer outside of the enterprise. An *internal product* is necessarily an *intermediate product* because it is partially aggregated and not sold to a customer outside of the enterprise (customers are internal, responsible for higher level products). The only remaining case is an intermediate product that is also an external product, which refers to a lower level product sourced to an outside supplier. The intermediate-external products are related to the *supply chain* function (i.e. sourcing, procurement) within enterprises. In part, this is a controlling function analogous to product management. It involves overseeing and acquiring a product created outside of the enterprise, while considering its technical performance, cost, and schedule. In a generic architecture, the supply chain managing process might be assigned to the supply chain management division.

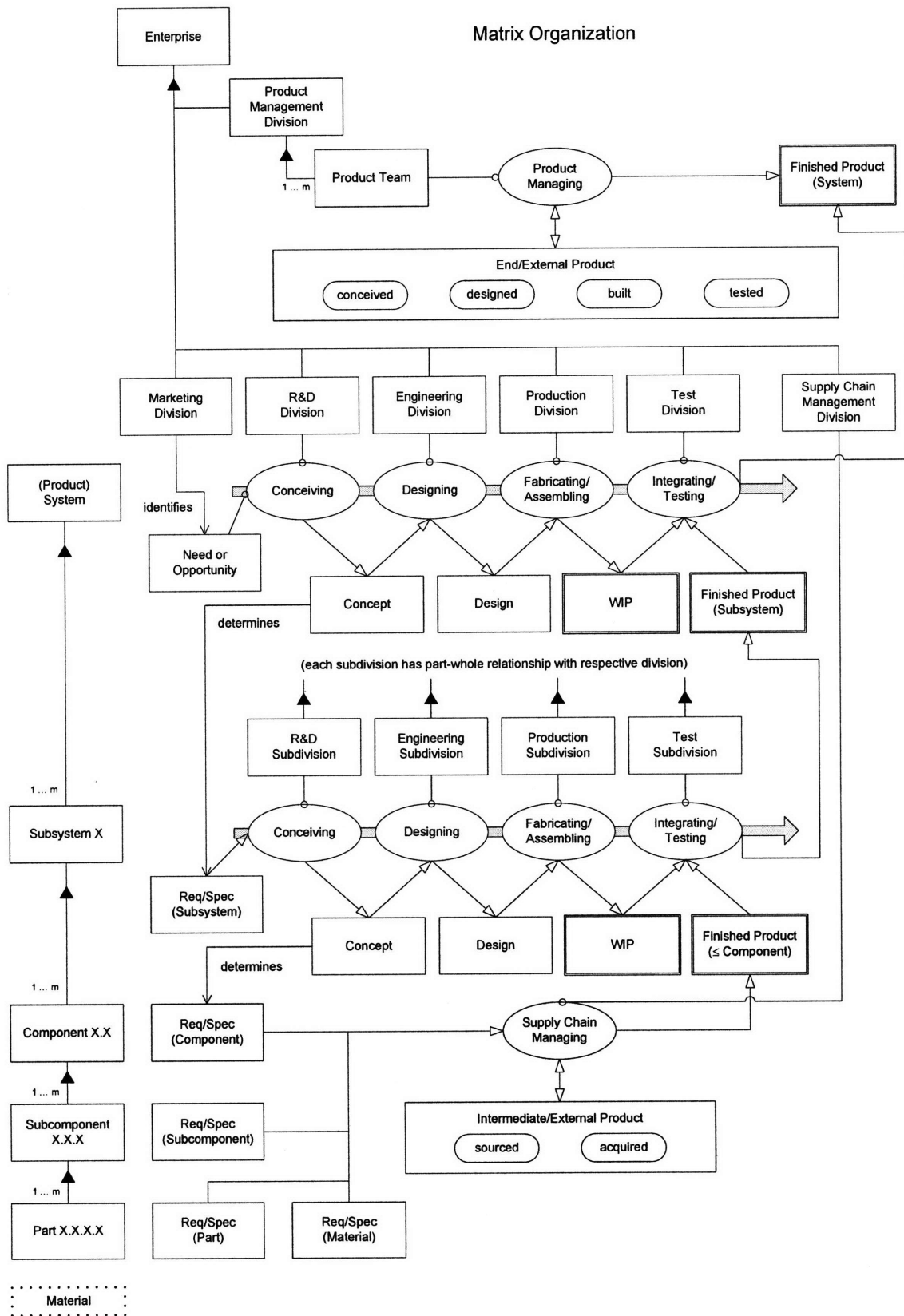


Figure 20: Integration of complex product and enterprise architectures

Figure 20 depicts a matrix organizational structure overlaid on the product life cycle for a complex product system. Note the disaggregation of functional groups into subgroups that are responsible for creating products at the subsystem level. In this example, the enterprise sources (to outside suppliers) all products below the subsystem level (e.g. components, subcomponents, parts, material). As such, the supply chain management function is inserted. Though not shown, in practice, the supply chain management division may disaggregate into sourcing teams for various classes of sourced items.

2.6.3. Independent Tiers of Product Management

The architectural matter at the heart of the thesis problem statement can now be identified with respect to the framework. Thus far, the product management function has been shown to provide oversight for the end-external product. For complex product systems, this highest tier of oversight is often referred to as *program* management – e.g. the Boeing 777 program or the Lockheed Martin F-22 program. The *program office* (labeled as Product Team in Figure 19 and Figure 20) serves as a direct liaison between the enterprise and its product’s external customers – e.g. commercial airlines or the United States Air Force. It is ultimately concerned with the system-level performance and system development cost and time. In isolation, it seems logical for the program office to disaggregate into subteams that oversee, in a similar manner, the intermediate-internal products at the subsystem-level and below. However, for an enterprise that creates multiple end-external products, there is an argument for having an *independent* lower tier of product management that oversees intermediate-internal products, where this lower tier is *not* hierarchically underneath one or more program offices. The rationale for this is discussed next. Here, Figure 21 depicts how independent tiers of product management might occur within a matrix organizational structure. As shown, the lower tier, Intermediate Product Team (**Product Office**), is joined directly to the enterprise rather than through the higher tier, End Product Team (**Program Office**). Not entirely obvious from the OPD, it is important to note that “1 ... m” tags (i.e. *participation constraint* in OPD symbology) do not imply a one-to-one pairing of higher and lower tier teams. On the contrary, the lower tier product teams are deliberately unmatched with a particular program office, so that they might satisfy the requirements passed down from multiple program offices in an optimized way.

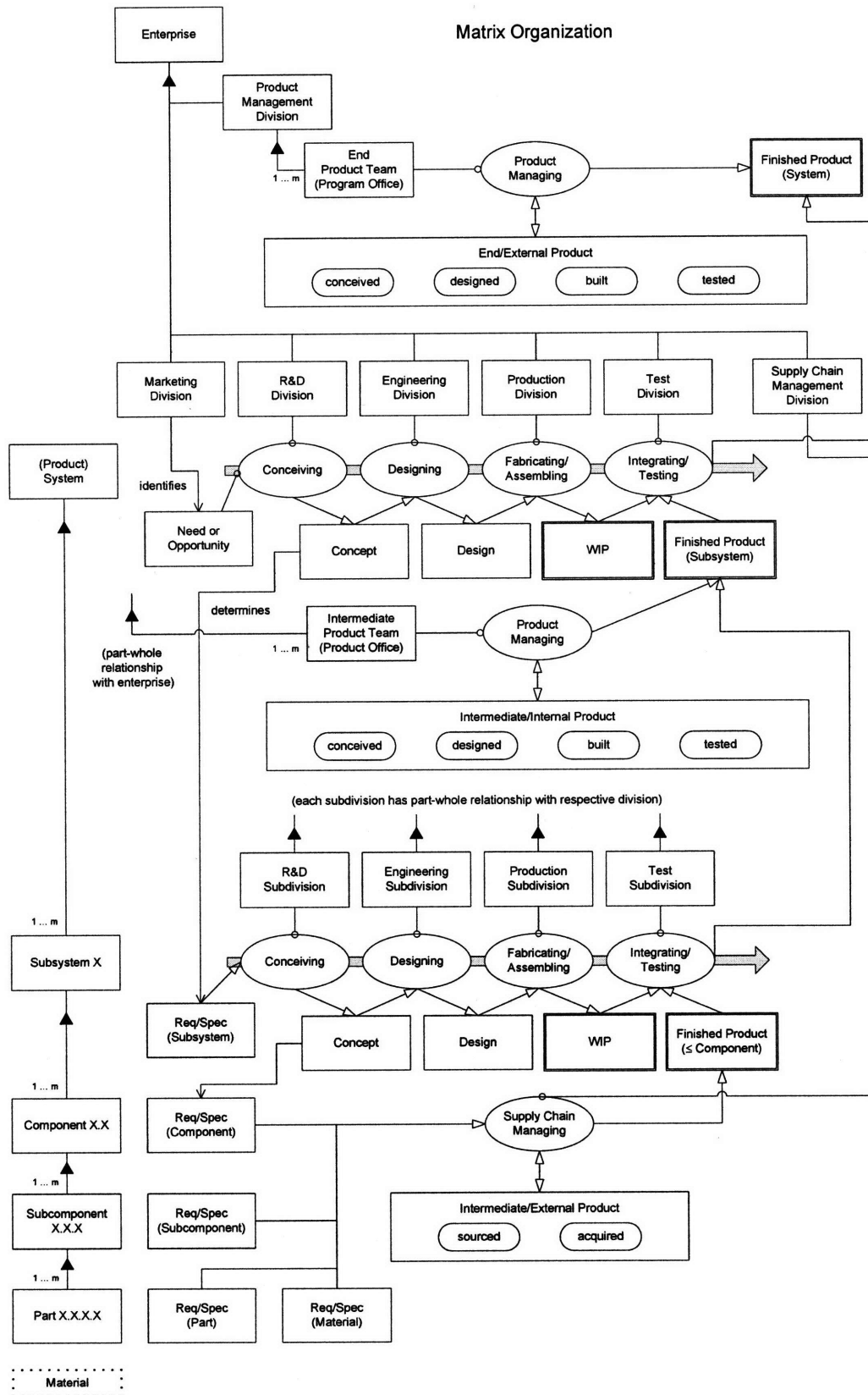


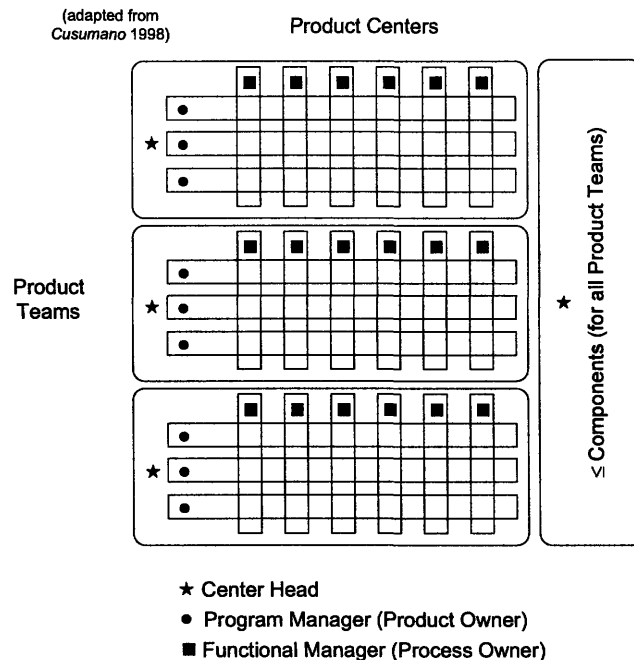
Figure 21: Independent tiers of product management – matrix organization

2.6.4. Distinctiveness and Commonality

Independent tiers of product management are driven by how products' architectures compare and contrast to each other. Some elements (subsystems, components, and so on) may only satisfy requirements specific to a *single* end product, given the unique needs of a particular external customer. Other elements may potentially satisfy requirements similar among *multiple* end products, given the common needs of several external customers. A critical product management task is identifying which elements fall into which category – distinct or common – and then overseeing the product life cycle to ensure that elements are created in an optimal way. One approach is to consider each element as distinct and then design and build it as such for a single end product. However, there is an important tradeoff between *distinctiveness* and *commonality*. Robertson explains, “customers care [that] a product closely meets their needs; they are not particularly concerned about how many parts a collection of products has in common... cost [for the enterprise] is largely driven by the level of parts held in common among a collection of products and is not directly related to how distinctive those products are” (Robertson 1998, 21). The author would suggest that customers are also concerned with commonality to the extent that it drives cost and, in turn, price. Therefore, from the perspective of both the enterprise and external customers, it would not be optimal to create a distinct element for each of several end products, when they could reasonably be created as a single common element for all end products.

How does either distinctiveness or commonality tend to arise with respect to the product system hierarchy? As a general rule, distinctiveness is more critical for higher level products because customers are more concerned with how *system-level* function and performance meet their specific needs. In contrast, commonality can prevail for lower level products because many subsystem-level functions are perceived by customers to be generic. Oftentimes, only a few subsystems explicitly contribute to the ultimate distinctiveness of the end product. The important consequence for enterprise architectures is that managing for distinctiveness differs from managing for commonality. Since program offices are more inclined to manage their end product for distinctiveness, it may be effective for an independent tier of product offices to manage intermediate products for commonality. As a benchmark, consider how Toyota Motor Corporation implements two tiers of product management within a center organizational

structure. Figure 22 depicts Toyota's product development organization using a diagram adapted from *Cusumano* (1998) and the author's translation of this diagram into an OPD.¹⁵ For clarity, the OPD omits the lowest levels of the product system. At Toyota, each product center (3) concentrates on a particular grouping of end products – e.g. one center each for rear-wheel drive and front-wheel drive vehicles. Within each center, product teams (≤ 5) each own a particular vehicle program, and functional groups (~ 6) each own a development process carried out for all vehicle programs in that center (effectively, a matrix organization within the center). At the same time, a fourth product center owns the development of lower level products for all vehicle programs in the first three centers. Toyota's rationale for this enterprise architecture involves the tradeoff between distinctiveness and commonality. Whereas a program manager oversees the entire vehicle and those subsystems distinctive to his/her vehicle (e.g. upper-body and interior design), the fourth product center oversees those subsystems and components partially or completely common to all vehicles (e.g. batteries, air conditioning units). In this way, the respective product teams can use management criteria tailored for their product's role within the architectures of one or more vehicles.



¹⁵ This figure represents Toyota's product development organization, as of 1992 (Cusumano 1998, 30). Note that Cusumano only considers Toyota's *product development* centers, without showing organizational connections between these centers and production. Still, the production and test functional groups are included in the OPD.

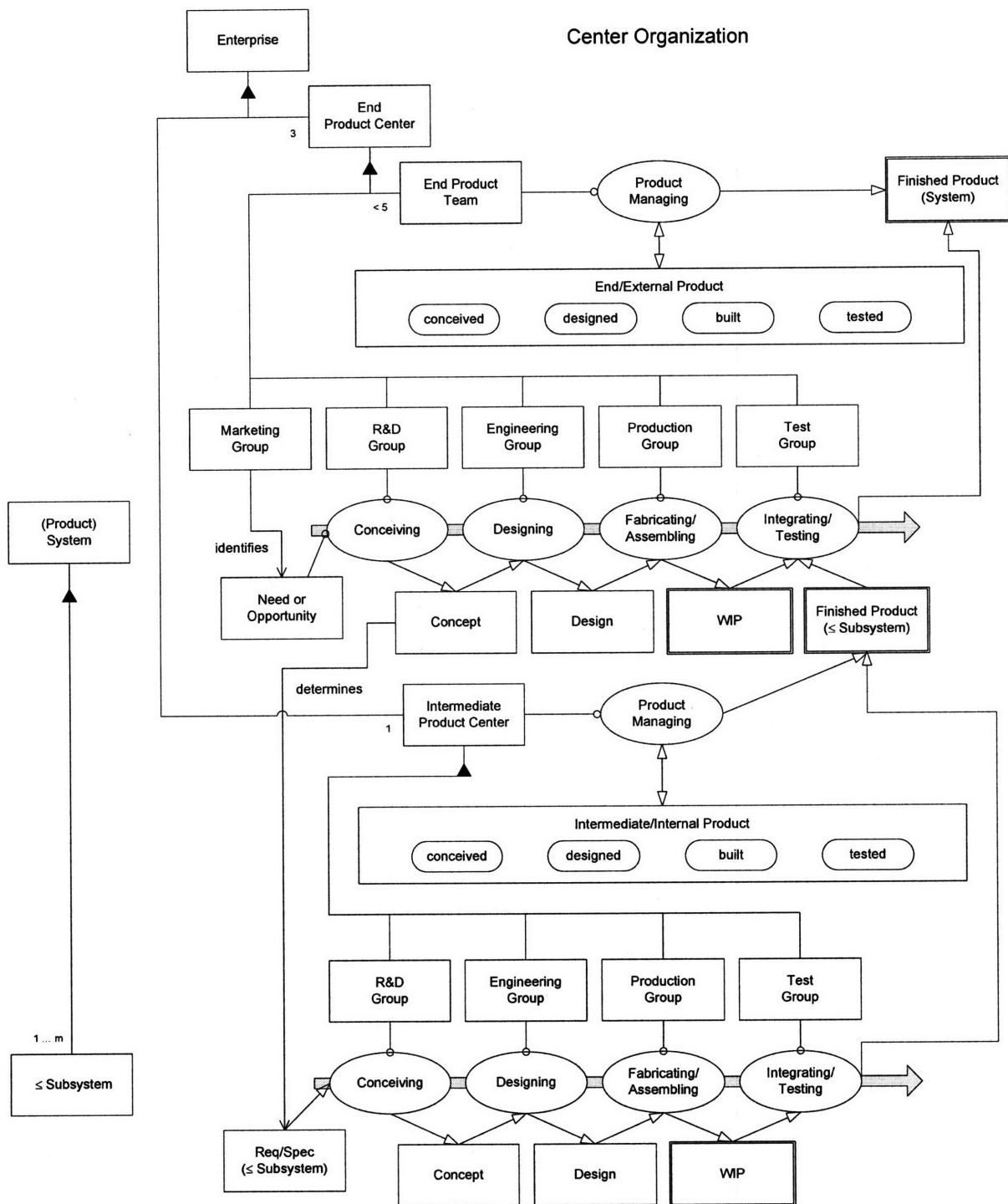


Figure 22: Independent tiers of product management – center organization (Toyota)

can be identified as more or less suitable for stovepiping, cloning, or platforming. Rigorous methodologies have been proposed for determining when platform development is appropriate. For example, *design structure matrices* can identify groupings of functional requirements that are decoupled from each other. It is beyond the scope of this thesis to examine in detail frameworks and techniques for platform development. The objective here is only to consider how organizational structure influences the enterprise's ability to pursue a range of product management approaches during the creation of complex product systems.

2.7. Section Summary

This section has presented a comprehensive framework to formalize the relationships between products, processes, and organizations. A generic model for systems, which included the notions of form, function, and architecture, was applied to products and enterprises. The process stream, which is closely related to the product life cycle, was introduced to show the progression from concept to finished product, with consideration for how product essence and process value change along the flow. These basic constructs were integrated into a framework showing the interconnections between product and enterprise architectures. Several common enterprise architectures were described with the inclusion of controlling functions, such as product management and supply chain management. The possibility of independent tiers of product management was discussed with Toyota offered as a benchmark. Finally, the tradeoff between distinctiveness and commonality among multiple products was explained and related to a spectrum of product management approaches. In the next section, this framework is applied to the NGST case study.

This page has been intentionally left blank.

3. Case Study for Complex System Management

This section presents a case study of Northrop Grumman Space Technology. As explained in the first section, NGST has established revenue growth as a primary goal and is faced with delivering competitive products as their existing and potential government customers stress affordability and speed of development. A recent organizational restructuring strives to position NGST to achieve this goal. The term *product focus* has been used to describe a general approach to ensuring the competitiveness of their products. This idea has influenced the restructuring in that certain functional groups have been reconfigured to promote greater accountability for all levels of products. For example, newly defined management positions have been created to oversee intermediate products. These types of modifications to the organizational structure are within the scope of the framework discussion in the previous section. This case study provides a rich example of how integrating the product and enterprise architectures – to achieve better architectural “fit” – influences the ability of businesses to improve their performance. The objectives of this section are as follows.

- Apply the presented framework within the developmental context of space systems and the organizational context of NGST.
- Evaluate the architectural “fit” and realization of *product focus* at NGST through comparison of their organizational structure before and after the realignment.

Following an order similar to that of the previous section, the initial two subsections of this section describe space systems, in particular satellites, and their product life cycle. The third subsection takes a detailed look at NGST’s organizational structure before and after the realignment. The fourth subsection analyzes the architectural “fit”, concentrating on the interaction between program offices and newly appointed product owners for intermediate-internal products. The remaining subsections discuss the challenges that NGST faces moving forward, as they strive to bring about the envisioned product focus.

3.1. Satellite Architecture

Space systems are among the most complex product systems ever developed by mankind. Modern satellites often contain cutting edge technology and yet demand the highest standards for reliability. Reliability is extremely critical because on-orbit servicing of satellite hardware is rarely feasible. A paradoxical outcome is that hundreds of spin-off technologies have emerged from space programs (e.g. Apollo, Space Shuttle) and yet satellites scheduled to launch in the next several years will likely contain computing hardware from the 1990s, as a result of rigorous space qualification standards. In addition to the high complexity of on-orbit elements, there is an indispensable infrastructure that deploys and supports satellites. Launch vehicle development and launch site operations constitute an entire subindustry of their own. Similarly, networks of ground control stations and their space support operations are equally critical, initiating data processing and dissemination and providing 24-7 satellite health monitoring. Accounting for the entire system, this translates into life cycle costs in the millions or billions of dollars and development periods of several years or more.

Figure 24 shows a decompositional view of a generic space system. The nomenclature for the product hierarchy is mostly consistent with that used by NGST and others in the industry. As noted above, the entire space system encompasses three so-called *segments*: (1) the on-orbit (located in space) system, (2) the launch system, and (3) the ground control system.¹⁶ In most cases, especially major programs for government customers, a single business does not (and cannot) completely provide all three segments. And so, in practice, a separate contract is often awarded for each segment. NGST mainly competes for lead development contracts for the space segment and less often seeks prime contracts for the ground segment (NGST chooses not to concentrate on the launch segment). The space segment, considered synonymous with *satellite*, is partitioned into two so-called *elements*: (1) the spacecraft bus and (2) the payload.¹⁷ Depending on the program and its contract scheme, NGST may play any of the following roles: (1) satellite prime integrator while providing both elements, (2) satellite prime integrator while providing only the spacecraft bus, or (3) major subcontractor providing only the payload. Other

¹⁶ In some cases, the system of ground-based network/user terminals (e.g. for military satellite communications) may be considered as a fourth segment.

¹⁷ *Element*, as used in the space system hierarchy, should not be confused with its usage above as an object of the product's form.

arrangements may occur, depending on how teams of businesses choose to bid for government contracts. Further decomposition of the spacecraft bus is discussed next, but consider the lower levels of the product hierarchy shown in Figure 24. Satellite subsystems, for either the spacecraft bus or payload, are more often than not a combination of *mechanical* and *electrical*

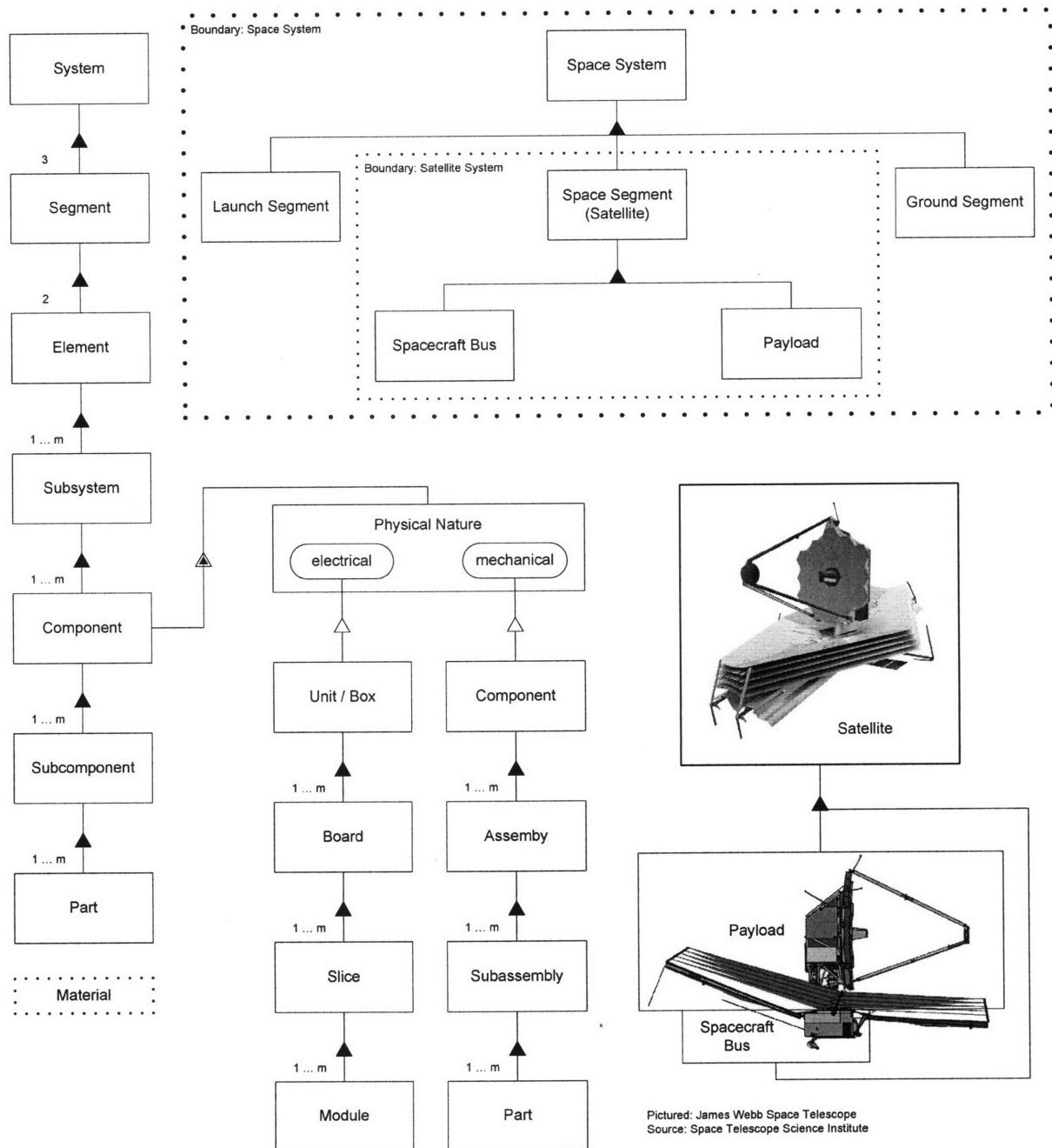


Figure 24: Decompositional views of a space system

components. As shown, a different nomenclature pertains to each class of component, though some components may not strictly fall within one category. For the remainder of this case study, the system boundary is drawn around the satellite (space segment). Furthermore, NGST's end-external products are limited to *either* the spacecraft bus *or* the payload, and it is assumed they always perform system-level integration when providing the spacecraft bus (i.e. a satellite with NGST's spacecraft bus and a subcontractor's payload).

The reason for partitioning the satellite into the spacecraft bus and payload relates to the overall product architecture. Generally speaking, the payload encompasses the elements whose allocated functions accomplish the satellite's *mission*, directly providing value to beneficiaries. The operand of a payload function is external to the satellite, for example, communication signals from earth or X-rays from outer space. In Larson's words, the payload "consists of the hardware and software that sense or interact with the [external] subject" (Larson 1999, 12). The spacecraft bus, or simply the *spacecraft*, encompasses the remaining elements that exist solely to support the payload and self-sustain the satellite as a whole. Figure 25 shows a decompositional view of the spacecraft bus.¹⁸ There are seven major spacecraft subsystems found in nearly all satellites.

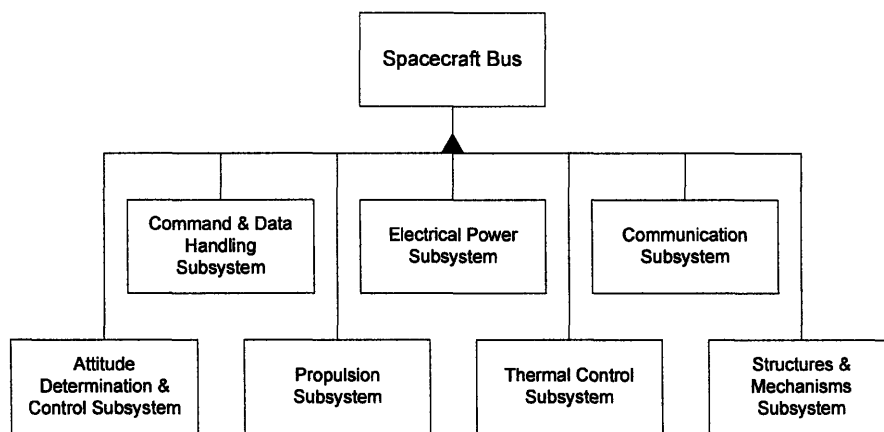


Figure 25: Decompositional view of a spacecraft bus

¹⁸ Some spacecraft subsystems are defined in different ways, referred to by various names, and have overlapping purposes. For example, the *communication subsystem* may also be called the *telemetry, tracking, and command subsystem*; the *attitude control subsystem* and *guidance and navigation subsystem* both deal with a satellite's position and orientation. Those shown in the figure are ones most commonly cited and capture the majority of spacecraft functions. Note also that *flight software* is not listed separately.

Since payloads are inherently mission-specific, there is no standard set of payload subsystems, but most can be classified broadly as (1) passive sensors intended to detect a given part of the electromagnetic spectrum (e.g. infrared, X-ray, gamma ray) or (2) communication systems intended to receive and transmit user data, serving as a relay in a network of ground terminals and other satellites. In the latter case, an all-encompassing communication system provides the data reception and transmission functions required by both the payload and spacecraft communication subsystem.

It is beyond the scope of this thesis to describe in greater detail the form and function of individual spacecraft subsystems. For the author's purposes, a basic understanding of the overall satellite architecture is sufficient. Figure 26 presents the generic architecture of a satellite, showing the high-level processes associated with each spacecraft subsystem. Despite the apparent simplicity of this OPD, modern satellites are characterized by a highly integral architecture and complex structural form with thousands of elements in a tight spatial configuration. This is driven by strict considerations for size, weight, and power (SWaP). Since launch cost scales directly with satellite mass and launch vehicles have maximum capacities in terms of mass and volume, minimizing weight and power is a prevailing design objective. Each kilogram eliminated from the spacecraft bus translates into lower launch cost or a greater allocation of mass to the payload.

3.2. Satellite Life Cycle

In general, the product life cycle for satellites contains the same set of processes observed for all product systems. Some additional processes – system engineering processes and technology development processes – are introduced here, given their particular importance for NGST. Figure 27 depicts process streams for a satellite and spacecraft subsystem. Two process flows are shown in parallel (though not explicitly connected) in order to highlight slight variations in terminology. In the top flow, the satellite progresses from concept to flight ready system during the developmental processes, and then becomes an operational, on-orbit satellite upon successful launch. *System engineering* is depicted as an end-to-end process that affects the satellite in an unspecified way. At the system-level, system engineering processes focus on the overall mission and high-level function and performance of the satellite. In the bottom flow, the spacecraft subsystem progresses from concept to finished subsystem, ready for integration and test (I&T)

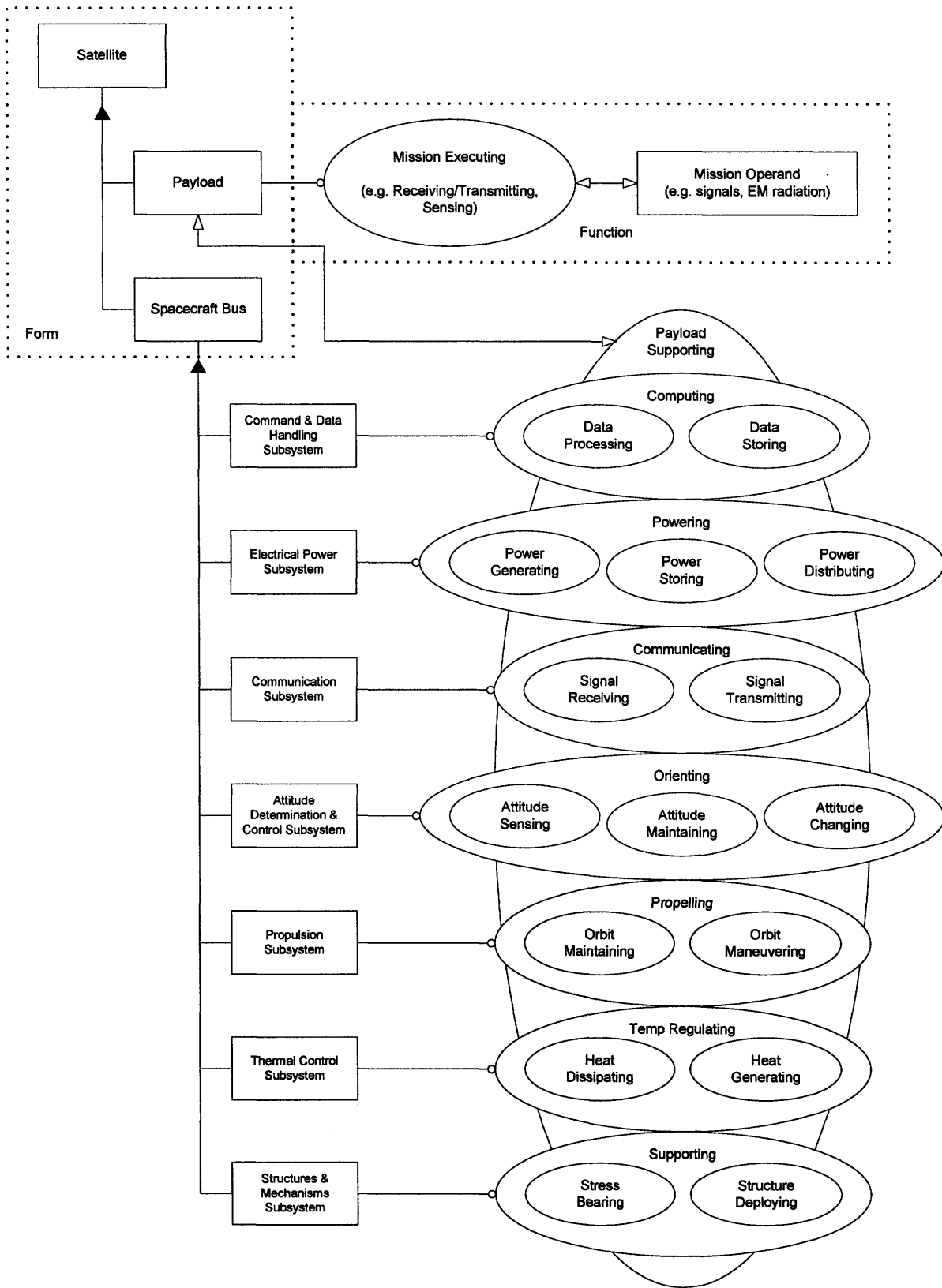


Figure 26: Generic satellite architecture

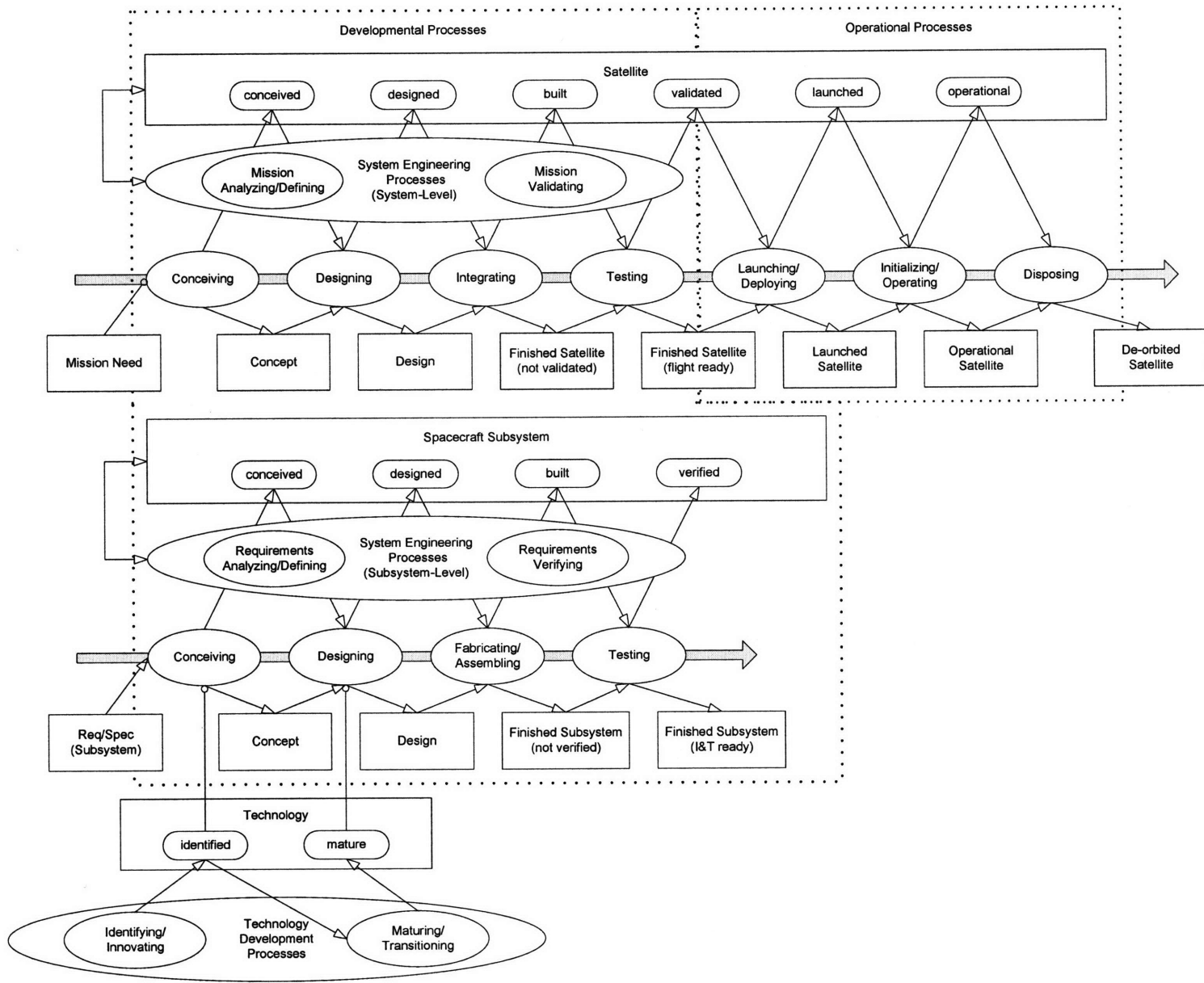


Figure 27: Process stream (product life cycle) for satellite and spacecraft subsystem

at the system level. Again, system engineering is as an end-to-end process that affects the subsystem, focusing here only on the requirements and functions allocated to the particular subsystem. *Technology development* is shown as a separate process outside of the main flow. This captures the reality that basic/applied research does not typically mesh with the sequential flow of product development. That is, new technologies are identified and matured on an often unpredictable schedule. Consequently, technology in itself is considered to enable conceiving and designing of spacecraft subsystems, and not be consumed by these processes.

There are two notes on terminology, which are illustrated in Figure 27. First, the building process is typically referred to using different terms at the satellite (or spacecraft) and subsystem levels. As shown, spacecraft subsystems are *fabricated* and *assembled* on their own, and the satellite is *integrated* as a whole. Second, the testing process has slightly different purposes for the satellite and subsystem levels. *Verification* refers to objectively proving that a subsystem or lower level product complies with requirements and specifications.¹⁹ Verification methods are often classified into four types: (1) inspection, (2) demonstration, (3) analysis, and (4) test. *Validation* refers to providing evidence that a system will meet its mission objectives under expected operating conditions at some confidence level. Consequently, as shown, spacecraft subsystems are *verified* during testing and the satellite is *validated*.

3.3. Organizational Structure

As noted above, NGST competes as a prime integrator of satellites and furthermore, maintains the in-house capabilities for end-to-end development of both the spacecraft bus and payload. Consequently, NGST's enterprise architecture encompasses the full scope of functional groups required to design and build spacecraft and payload subsystems, as well as carry out system-level integration and test of the satellite. Critical to the satellite integrator's role, system engineering is a formal end-to-end process that warrants its own functional subgroup. Technology development processes are also especially important to NGST, who, as described in the first section, has a reputation for pushing the state-of-the art of subsystem functionality and performance.

¹⁹ The system engineering discipline often distinguishes between a *requirement* and a *specification*, where the rather subtle difference relates to how they are stated in words and the nature of the test that provides verification or validation. Here, the two terms are used interchangeably.

This subsection discusses NGST’s enterprise architecture in both its pre- and post-realignment configurations. The figures presented here do not represent the exact organizational charts for NGST. Only the functional groups responsible for relevant *life cycle processes* are considered (i.e. functional groups such as human resources and information technology are omitted). Even for these, not all subgroups and sub-subgroups are shown, and the names given for groups and subgroups may or may not match the formal titles used by NGST. This is in the interest of clarity and is not meant to oversimplify their organization. The author has made every attempt not to misrepresent the actual or intended design of NGST’s organizational structure.

3.3.1. Pre-Realignment Architecture

Prior to the realignment, the governing concept for the NGST enterprise architecture was *process focus*. As mentioned, functional groups were assigned explicit ownership of processes, which were grouped into *core* and *enabling process areas*. Formal process ownership was meant to ensure that all processes were clearly defined, documented, measured, and managed for compliance and improvement. Figure 28 depicts NGST’s pre-realignment organizational structure with the hierarchical nomenclature on the left. As shown, there are six (core) process areas. In practice, NGST operated as a matrix organization (similar to that shown in Figure 19), where program offices are equivalent to product teams and major functional groups (e.g. Engineering, Production & Supply Chain, and Payload & Sensors) simultaneously provided resources to support each program office. For the most part, Business Development operated

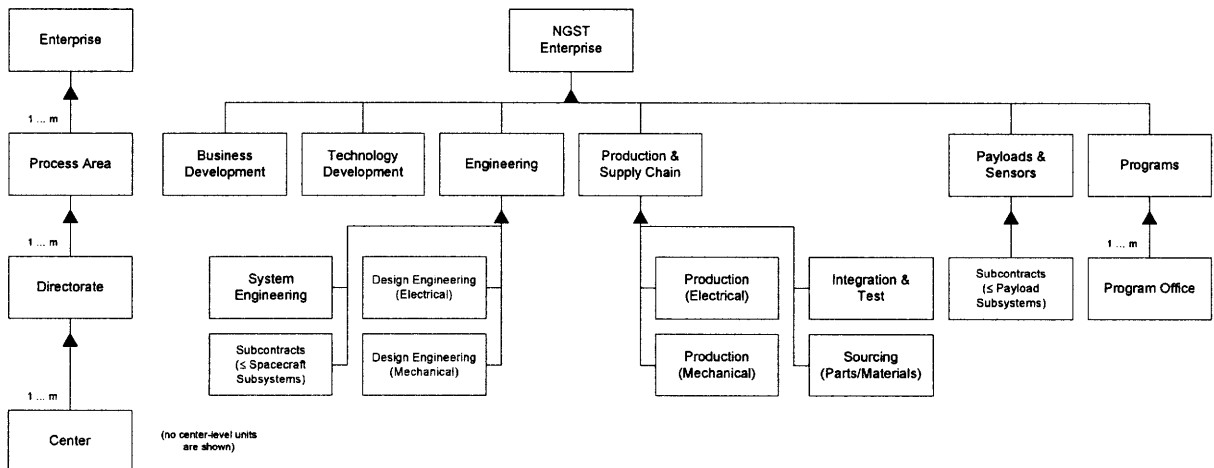


Figure 28: Organizational structure – pre-realignment

independently to identify market opportunities and develop bids and proposals for government contracts. A formal program office was established under Programs upon contract award and authorization to proceed. Similarly, Technology Development mostly operated on its own to run independent and cooperative research and development projects, transitioning new technology to Engineering and Production & Supply Chain when appropriate. Looking at the directorate level, there are several items to note. First, both the design engineering and production processes are partitioned into subgroups for electrical and mechanical spacecraft components. Second, the system engineering process, though end-to-end in nature, is housed in Engineering where the front-end mission and requirements analysis occurs. Third, processes related to the supply chain management function – sourcing commodity parts and material, managing smaller subcontracts for components or below, managing larger subcontracts for payload subsystems – are dispersed among multiple process areas.

Using the complete framework from the second section, Figure 29 illustrates NGST's pre-realignment enterprise architecture with respect to the satellite and spacecraft subsystem process streams given above. This version assumes that NGST is the satellite prime integrator while providing the spacecraft bus and subcontracting the payload. For clarity, only the six process areas (rather than the directorates) are used in the OPD. Overall, the enterprise architecture reflects the traditional matrix organizational structure with a single tier of product management (for end-external products) provided by the program office. Separation of functional groups is maintained at both the satellite and spacecraft subsystem levels (note that redundant instances are shown for Engineering and Production & Supply Chain in order to keep the OPD uncluttered). As noted above, there are multiple functional groups that manage intermediate-external products. For example, Payloads & Sensors is dedicated to managing subcontracts for payload subsystems. Note that management of subcontracts for spacecraft components, which is owned by Engineering, is not explicitly shown.

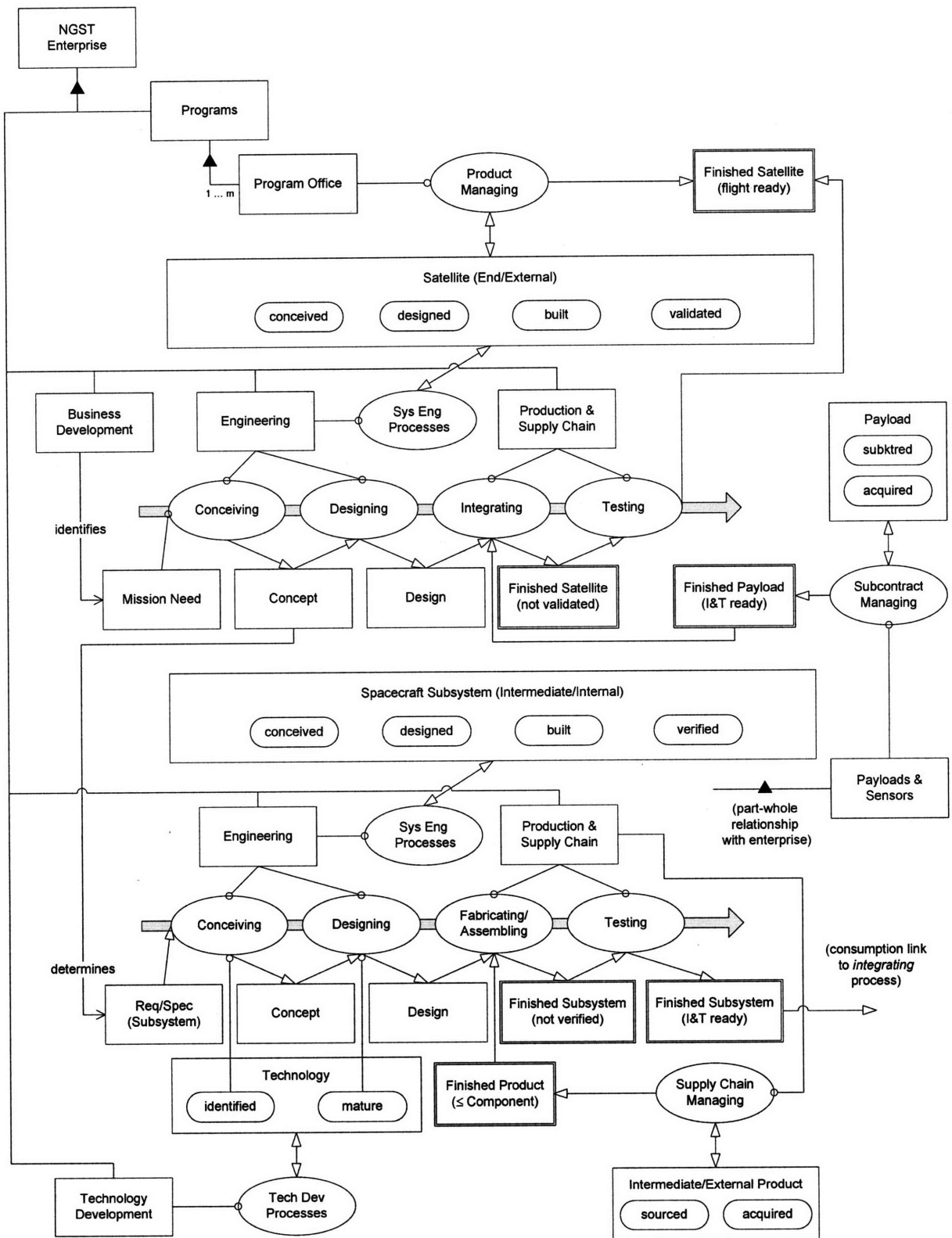


Figure 29: Integrated product and enterprise architectures – pre-realignment

3.3.2. Post-Realignment Architecture

The governing concepts under the realignment are that the NGST enterprise architecture embodies *customer focus* and *product focus*. As stated before, this does not represent a loss of process focus, but rather a change in emphasis made explicit through reorganization. Figure 30 depicts NGST's post-realignment organizational structure with the hierarchical nomenclature on the left. In order to better meet the needs of all government customers, the former Programs process area is divided into three Program Centers (at the division-level) – each for a different market segment (e.g. civil, military). Each Program Center groups the program offices for their respective market segment. Similarly, the former Business Development process area is divided into three subgroups and made internal to each Program Centers. The former Technology Development process area is also treated as a center for research and development projects, as well as some technology demonstration programs. This portion of the realignment is clearly intended to address customer needs and market opportunities in a more systematic way.

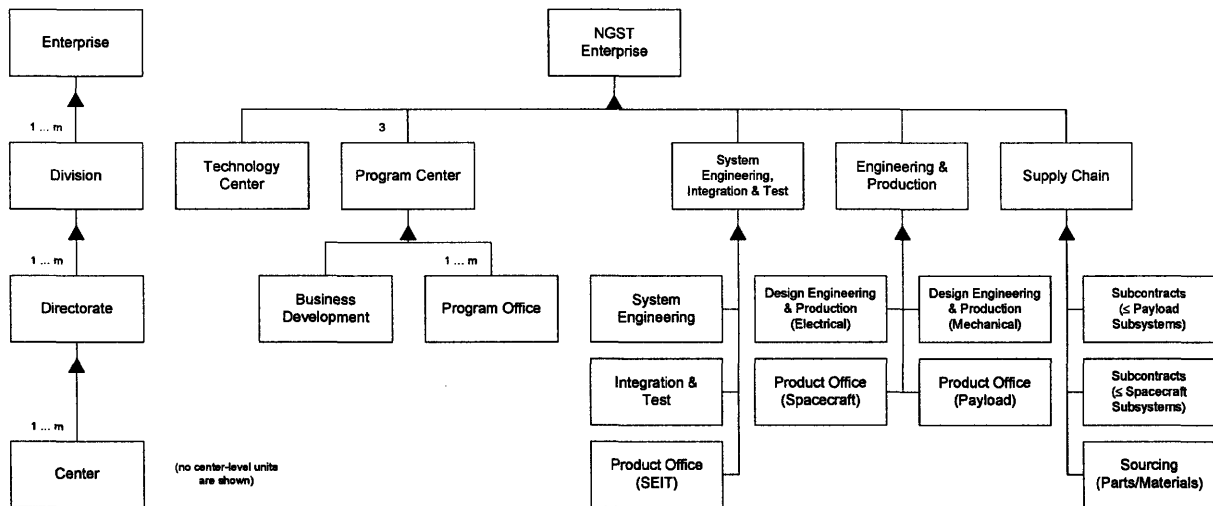


Figure 30: Organizational structure – post-realignment

In order to improve the end-to-end development of satellites, three divisions are formed by reconfiguring pieces of the former Engineering, Production & Supply Chain, and Payloads & Sensors process areas. First, the Engineering & Production division combines the design engineering and production subgroups for products at the subsystem-level and below (both electrical and mechanical components). Second, the System Engineering, Integration & Test

(SEIT) division marries the formal system engineering subgroup with the integration and test subgroup. In sum, this represents the front-end mission and requirements analysis and the back-end system-level validation. Third, the Supply Chain division consolidates management of all subcontracts (both spacecraft and payload subsystems) and sourcing of commodity parts and material. Note also the subgroups labeled Product Offices under the Engineering & Production and SEIT divisions. These are brand new directorate-level units that represent a second tier of product management responsible for lower-level products. The significance of these product owners are discussed in detail in the next subsection. Overall, this portion of the realignment is clearly intended to refocus the enterprise on delivering competitive products at all levels.

Under the realignment, NGST's three customer divisions and three product divisions still operate as a matrix organization, where the latter provides resources to support the former. More precisely, it might be said that NGST has close to a semi-center organization because some functional disciplines are duplicated to serve individual program centers (e.g. business development) and others are not (e.g. engineering, production). Figure 31 illustrates NGST's post-realignment enterprise architecture with respect to the relevant process streams. Again, this version assumes that NGST is the satellite prime integrator while providing the spacecraft bus and subcontracting the payload. For clarity, only divisions are used in the OPD. At first glance, the "tiered" nature of this enterprise architecture stands out. For example, the Engineering & Production division carries out end-to-end product development of spacecraft subsystems and the SEIT division effectively does the same at the satellite level. Note that two instances of system engineering processes are shown and enabled by two different divisions. This is meant to draw a distinction between the system-level activities (owned by SEIT) that are most critical for the satellite integrator and the subsystem-level activities (owned by Engineering & Production) that are inherent in the architecting and detailed design of all products. Overall, this enterprise architecture clearly positions NGST to achieve a greater product focus because accountability for each of the three classes of products rests with a *single* division. That is, the Supply Chain division owns all *intermediate-external products* (from material to payload), the Engineering & Production division owns the process stream for all *intermediate-internal products*, and the SEIT division owns the process stream for all *end-external products*. The key question is how independent tiers of product management – Program Offices and Product Offices – will coexist to balance the development of intermediate-internal products and end-external products.

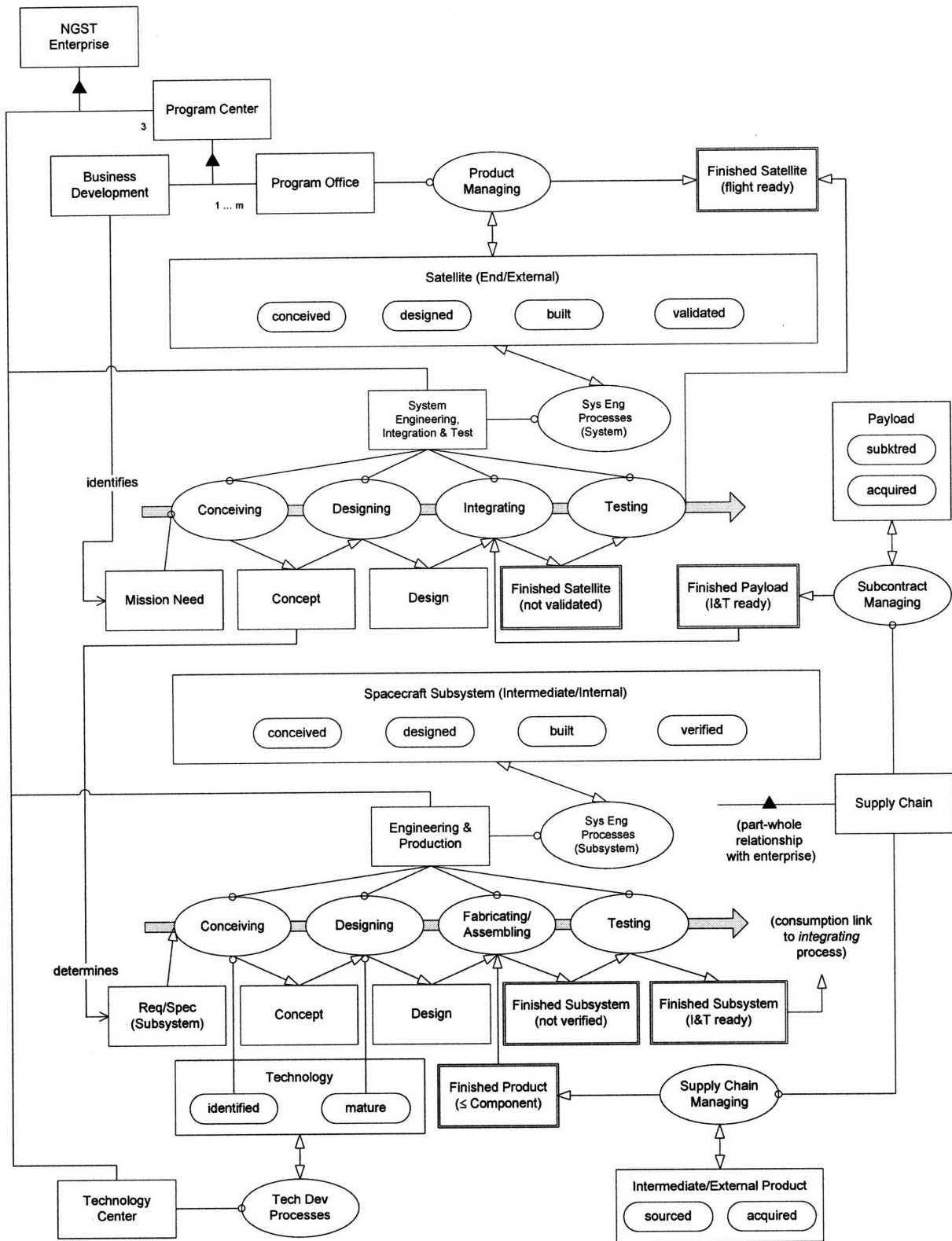


Figure 31: Integrated product and enterprise architectures – post-realignment

3.4. Evaluation of Architectural Fit

As suggested at the outset of this thesis, architectural “fit” is not something that can be measured directly, but rather correlates with the value provided to customers. This subsection evaluates NGST’s realignment by examining how their new organizational structure positions them to enhance the value of their products. In general, customer value increases as the product provides a better technical solution to customer needs, costs less, and takes less time to develop. Overall, the NGST realignment has the potential to increase customer value with respect to all of these. Since affordability and speed of development are key differentiators, the focus of this discussion will be on how their organizational structure enables reductions in development cost and time. It should be noted that a large majority of satellite development programs, especially the most technologically advanced systems that NGST often provides, are awarded through cost-plus contracts with the government.²⁰ As such, cost and time are effectively one in the same because government compensation is equal to the contractor’s expenses, to include all labor hours. And, in fact, direct labor (and related indirect charges) typically constitutes the largest proportion of program costs. Therefore, reducing development time (i.e. cumulative man-hours) necessarily reduces cost. Here, the potential for increased customer value is identified in two places within the integrated product and process framework.

3.4.1. Intra-Stream Value

The term *intra-stream value* refers to how well products at the *same level* (e.g. system, subsystem) progress along their own process stream. In the case of NGST’s products, the process streams for the satellite (or spacecraft) and spacecraft subsystems can each be isolated and assessed for their cost and time efficiency. Prior to NGST’s realignment, design and production were separated into distinct functional divisions. This created the oft-observed rift between engineers and manufacturers, where a design might be “thrown over the wall” to production. Because the Engineering and Production & Supply Chain process areas were division-level units, this separation was present at all product levels. Even as the process focus took root throughout NGST, inter-division coordination was not always optimal. Despite the matrix organization, which created cross-functional teams to serve each program office, there

²⁰ Government contracts are categorized broadly as either *fixed-price* or *cost-plus*. The contractor bears most of the development risk under fixed-price contracts. The government bears most of this risk under cost-plus contracts. Cost-plus contracts typically include an incentive or award fee paid to the contractor based on their performances.

was still a tendency for *functional stovepipes*, or so-called “cylinders of excellence,” to exist. Process areas strived to control and improve the life cycle processes that they owned, but this action in itself did not necessarily guarantee efficient coordination among adjacent process areas. Consequently, the handoff between design and production was not always smooth, resulting in various forms of inter-division rework. For example, engineering orders (EOs), which amend build-to drawings (e.g. type and quantity of parts, assembly instructions) and are passed from design to production, could range from typographical errors to substantial redesign. If designs were found to pose problems for production, or engineering found reason to change an already delivered design, this created delays and disruptions in the form of EOs. To be fair, many EOs propagate from changing requirements and specifications on the part of external customers. However, separate functional divisions for design and production built into the organizational structure a higher likelihood for this critical handoff in the process stream to be a source of delay.

The realignment positions NGST to improve coordination along the process streams for their products. At the subsystem level (spacecraft bus and payload), design and production are now combined within a single division. Furthermore, the Engineering & Production division contains two directorates (see Figure 30), each responsible for design and production of either electrical or mechanical subsystems. This configuration has the potential to significantly improve the handoff between functional disciplines. Most notably, there are greater opportunities to implement *design for manufacture and assembly* (DFMA) guidelines, which should reduce EOs and other forms of design rework discovered during production. This was not completely absent within prior cross-functional product teams, but now the functional reporting chain is much shorter (i.e. a single director owns both design engineering and production processes), which should enable better coordination. Overall, the Engineering & Production division has tighter end-to-end ownership of subsystem-level processes, which has the potential to reduce development time and cost.

At the system level, coordination is similarly improved between system engineering and I&T. Though not explicitly shown in Figure 30, an important functional subgroup associated with I&T is one responsible for the development of ground test equipment, often referred to as *test sets*. While some tests are common to all satellites (e.g. shock and vibrational testing, thermal vacuum testing), many are specific to the spacecraft and payload and are designed to test operational

command sequences particular to the mission. For example, test sets contain customized hardware and software that simulate or emulate signals expected in the on-orbit operational environment. Since system engineering conducts the mission-level requirements analysis, articulates system-level specifications, and constructs the matrix of verification/validation methods, coordination with test set development and I&T is critical. Here again, these functional disciplines are now combined into a single division. The SEIT division has the potential to significantly improve the handoff between system engineering and I&T. Similar to DFMA between engineering and production, *design for testability* should reduce any rework in the development of test sets and avoid delays during the validation testing of the satellite. Overall, the SEIT division has end-to-end ownership of system-level processes, which should again position NGST to reduce development time and cost.

3.4.2. Inter-Stream Value

The term *inter-stream value* refers to how well products at *different levels* are managed in parallel and integrated upward in the product hierarchy (see the nested pattern of process streams in Figure 15). Prior to NGST's realignment, the product management function existed primarily within the bounds of a program office. For example, the Program Office managed the satellite as a whole, and a group of integrated product teams (IPTs) were assigned product ownership of the program's spacecraft and payload subsystems, including so-called *configured items* of critical importance to the system. In practice, this product management structure formed *program stovepipes*. Given the tendency of Program Offices and their IPTs to manage most levels of products for distinctiveness, a stovepiping approach (see Figure 23) treated both the end-external product and its intermediate-internal products as one-off designs. This is not to say engineers started with blank sheet designs (clearly, cumulative learning and tacit knowledge improved the design cycle from program to program). But design reuse and other forms of commonality were less prevalent because product ownership rested within the bounds of the Program Office. The outcome with respect to customer value was highly customized product solutions that incurred full development cost and time. For some subsystems and configured items, this was necessary, given unique mission requirements. For others, what tended not to be fully explored was how to leverage previous or concurrent designs to shorten the development cycle and boost the production volume of some components.

Another consequence of this product management structure was difficulty in identifying the *average* cost and cycle time of intermediate-internal products. Because ownership of lower level products did not cross-cut the program offices, there was limited accountability for their end-to-end to development from a *program-neutral perspective*. Cost and schedule management was tailored for each program and documented as such at all product levels. The outcome was that subsequent programs found it difficult to benchmark previous and concurrent ones. For example, the Business Development process area had to reconcile different cost accounting structures among programs (e.g. how to allocate labor and material costs to subsystems or so-called box-level products), in order to generate bottom-up bids on new contracts. Overall, NGST's tendency towards program stovepipes impeded greater accountability for lower level products and deterred a platforming approach for many spacecraft subsystems and components.

The realignment positions NGST to break down these program stovepipes by implementing independent tiers of product management. As noted, a key feature of the new organizational structure is the Product Office (at the directorate-level) within the Engineering & Production division. Though the exact nature of its responsibilities is still being determined, the general idea is to introduce a product management function for intermediate-internal products that is completely independent from Program Offices. Figure 32 depicts the two-tiered product ownership structure envisioned by the realignment. This figure provides greater detail at the subsystem-level than Figure 31 (and omits the supply chain and technology development functions to avoid a cluttered OPD). Note the distinction between *process ownership*, assigned to Design Engineering & Production directorates, and *product ownership*, assigned to the Product Offices. The former takes a program-neutral view of process control; the latter takes a program-neutral view of product management. A tandem of process and product owners also exists at the system-level, where the directorates within the SEIT division control the satellite process stream and the Program Offices manage the end-external products.

At a minimum, the Product Office provides greater accountability for intermediate-internal products and a formal oversight mechanism for managing their technical performance, cost, and schedule. The Product Office is effectively a supplier of finished products to a set of internal customers – the Program Offices. Just as NGST's government customer contract for a satellite within prescribed budget and time constraints, the Program Offices can internally “subcontract”

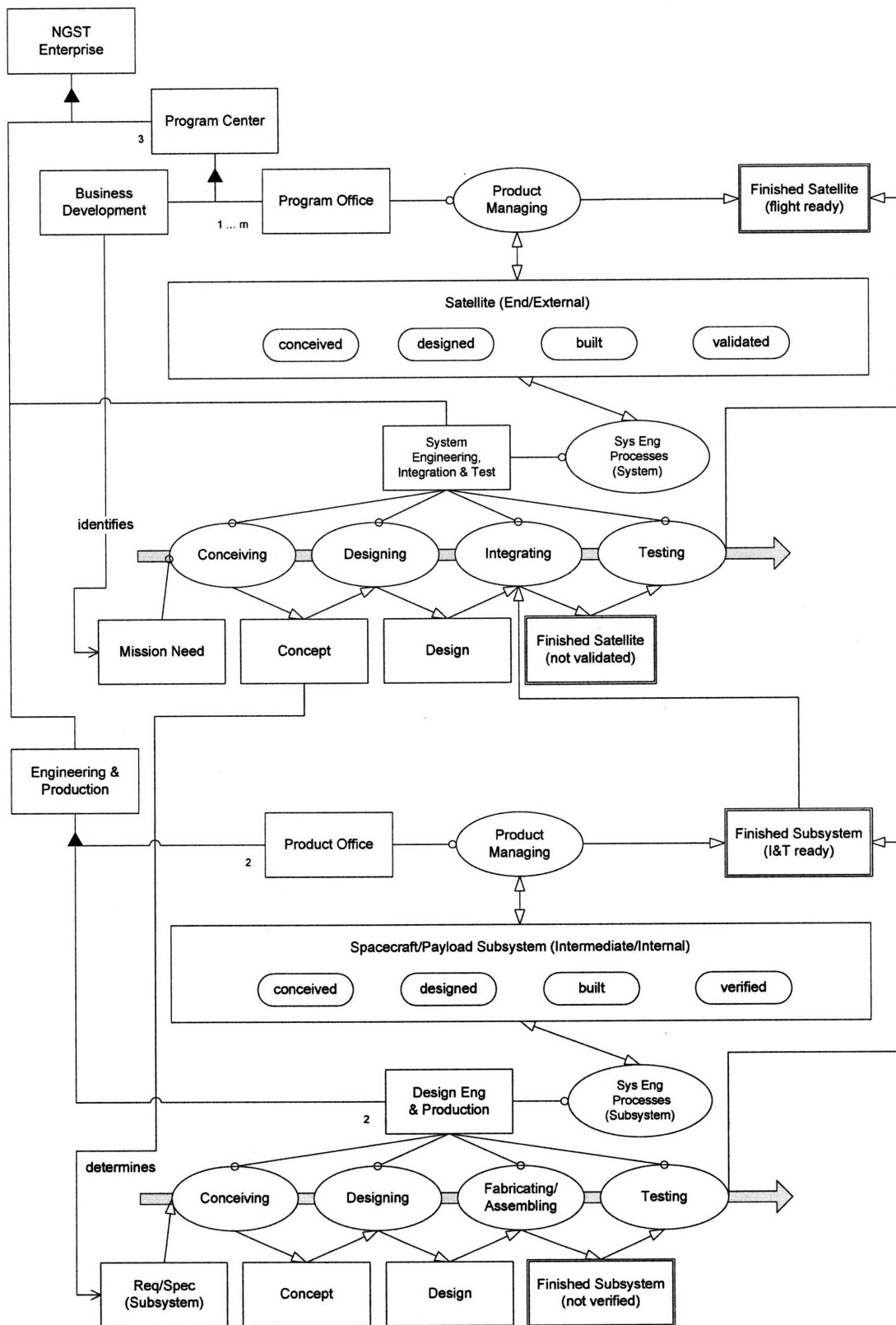


Figure 32: Independent tiers of product management – post-realignment

the Product Office for spacecraft and payload subsystems in the same way. This two-tiered model for product creation is now built into NGST's new organizational structure.

More importantly, the Product Office has the potential to significantly increase customer value by developing "product lines" of spacecraft subsystems and components. For these lower level products, there is an opportunity to introduce greater commonality across programs, moving from a stovepiping to a platforming approach, wherever possible. Commonality drives reductions in development cost and time in two main ways: (1) minimizing the time to complete upstream development processes, generally referred to as *non-recurring engineering* (NRE) and (2) generating economies of scale in downstream production processes.

At NGST, NRE is often pointed to as a primary target for cost reduction. Eliminating design rework, discussed above as part of intra-stream value, is one means to decrease NRE. However, commonality provides a more sweeping method to reduce NRE. In theory, NRE should approach zero as product managers approach the cloning end of the spectrum (see Figure 23). Though 100% design reuse of spacecraft subsystems is unlikely, platforming has the potential to share subsystem NRE across concurrent programs and evolve a family of similar designs for use in follow-on programs. Program stovepipes have typically prevented systematic sharing of NRE, though at least one government contract for a pair of satellites successfully pursued a common spacecraft bus. While a so-called *common bus* may or may not be feasible over NGST's full range of market segments, the Product Office can determine which spacecraft subsystems and components have widely common designs, and develop product lines accordingly.

The other consequence of commonality is increased repetition of recurring processes for a given product, such as fabrication, assembly, and test. This translates into cost reduction through economies of scale and learning effects from cumulative output. Though satellite production is inherently a low-volume, high-mix industry, some appreciable scale might be reached at the subsystem and component levels. Product line development can be an effective means to realizing this scale, especially now that the Product Offices can work in conjunction with the Design Engineering & Production directorates to evolve product lines and manufacturing processes in a coordinated manner. Overall, the implementation of independent tiers of product management has the potential to greatly reduce development cost and time across all programs.

3.5. Challenges Following Realignment

The above discussion suggests that the realignment has, in fact, positioned NGST to realize the *product focus* envisioned by their executive management team. Formal ownership of intermediate-internal products in the form of Product Offices provides both features identified in the first section of this thesis – (1) mechanisms to ensure *accountability* and (2) methods to manage product *evolution*. Still, a new organizational structure in itself does not guarantee better business performance. This subsection outlines several challenges that NGST faces as they look to realize the expected advantages of this new structure.

NGST has implemented numerous reorganizations throughout its history in the satellite industry and each one has brought with it general challenges inherent to any organizational change, as well as some particular to the immediate change. Described here are those challenges that might be encountered as part of the latest realignment.

Program-centric execution. For the most part, programs have always been “king” on the Space Park campus, which comes as no surprise because program offices take their lead directly from their external customers. As government customers for cost-plus contracts have come to expect build-to-order, customized products from NGST, program offices have learned to develop them within the traditional matrix organization. As discussed, the outcome has been a tendency to form program stovepipes, where IPT members narrowly focus on the objectives of their assigned program. The two-tiered product management structure has the potential to weaken customized development of lower level products in return for the benefits of greater commonality. Will a Product Office’s development decision take priority over that of a Program Office, if the global optimum for all programs requires one Program Office to suboptimize? In order for the Product Office to be truly effective, Program Offices (and their external customers) must be willing to negotiate or compromise under certain circumstances.

Engineering culture. NGST’s organizational culture has long been dominated by engineers, who take great pride in the innovativeness and technological superiority that characterize the satellites they design. Given this, maximizing design reuse and other forms of commonality may not be an attractive objective for many engineers. It would not be out of the question for an engineer to pass on an opportunity to reuse an existing component design in favor of a chance to

redesign it from scratch in order to optimize performance. Functional managers must ensure that engineers comply with design reuse processes and still provide them with opportunities to tackle technically challenging problems.

Pace of technological change. Rapid changes in technology pose a significant challenge to developing and maintaining product lines of spacecraft subsystems and components. Due to the long development cycle for satellites, there are a limited number of programs underway at any point in time. Consecutive generations of the “same” satellite (i.e. mission capability) can be separated by more than five years. Meanwhile, core technologies are continuously improving or being disrupted by more advanced ones. In order for NGST to sustain product lines, the Product Offices must be engaged to some extent with the technology development processes to ensure continuity between current and follow-on programs. Otherwise, a product line arising from a group of concurrent programs may be obsolete without a development path for inserting next-generation technology.

Cost accounting regulations. Even if NGST’s external customers support a platforming approach to developing spacecraft subsystems and components, there are certain practical matters to overcome. For one, federal regulations prescribe strict cost accounting procedures for government contracts. All labor hours, material purchases, indirect charges, and so on are carefully billed to job numbers associated with particular contracts. Because federal funding is allocated for specific tasks within a contract’s statement of work, it is difficult to pool funding from multiple programs in support of product line development. In fact, engineers engaged in design work for separate programs are prohibited from meeting together and then charging labor hours to separate contracts. In order to share NRE, NGST must cooperate across government customers to identify arrangements that fall within the cost accounting regulations.

Security requirements. Due to the sensitive nature of some satellite programs, in particular those for military customers, security requirements also present a challenge for product line development. Some technologies, which might otherwise be incorporated into a common platform of subsystems or components, may not be transferable to programs without the same security restrictions. In this way, product lines might be confined to certain market segments or to programs contracted for a subset of government customers.

3.6. Section Summary

This section has presented a case study of Northrop Grumman Space Technology, focusing on their recent organizational realignment. A primary objective has been to use the framework proposed in the previous section to evaluate how NGST's new organizational structure positions them to ensure the competitiveness of their products. The product architecture for space systems was described, with particular attention given to the decomposition of the spacecraft bus. The process streams for satellites and spacecraft subsystems were elaborated. System engineering and technology development processes were introduced, which are critical to NGST's role as a system integrator and technology leader. The organizational structure and enterprise architecture of NGST were discussed in both their pre- and post-realignment configurations. A key feature of the new organizational structure was how accountability for each class of product is now resting with a single division. The architectural "fit" was evaluated by identifying opportunities for reducing development cost and time. Within the process stream, combining design and production and combining system engineering and I&T are potential sources of value due to better coordination. Between process streams, the two-tiered product management structure provides the opportunity to pursue greater commonality at the subsystem and component levels. Finally, several challenges to realizing the benefits of the new organizational structure were described. The next and final section summarizes and reiterates the key insights gained from this research effort and suggests future areas of study.

This page has been intentionally left blank.

4. Conclusion

It is worth repeating the insight offered at the opening of this document: “The architectures of the product, its construction process, and the organization that manages them should “fit” each other” (Rechtin 2000, 159). This thesis strives to provide some formality to this heuristic by systematically describing products, processes, and organizations, and modeling interconnections among all three. In doing so, a primary objective has been to identify features of the enterprise architecture that improve customer value by addressing the full scope of their needs – technical performance and reliability, affordability, and timely product delivery. It appears that the critical construct within the enterprise architecture is a tandem of formal process owners and product owners. While a process owner takes responsibility for the design and control of one or more life cycle processes within the value stream, a product owner takes responsibility for the end-to-end development of products across the entire value stream. This is relatively straightforward when the enterprise creates simple products. A traditional matrix organization of functional groups and product teams can be quite effective under these circumstances.

However, in the context of complex product systems, the development effort expands into a hierarchy of intermediate subsystems and components that compose the end system. These subsystems and components constitute products in themselves and can be assigned a product owner in the same way as their parent system. Process owners can also be designated to control a parallel value stream for these lower level products. This two-tiered structure of product owners and process owners enables the enterprise to develop each level of products according to different criteria. Distinctiveness and commonality may be balanced differently at each level. This thesis has provided examples of how the enterprise architecture can adapt to “fit” such complex product systems. Toyota’s center organization for product development has been a benchmark in the automotive industry. This thesis has presented a case study of NGST to illustrate a similar approach in the satellite industry.

4.1. Research Outcomes

As mentioned earlier, the announcement and implementation of NGST’s realignment occurred during the course of the author’s internship. While many changes to the organizational

structure were reconfigurations of existing functional groups and subgroups (e.g. combining design engineering and production, combining system engineering and I&T), the Product Offices represented formations of brand new directorates. The author had the opportunity to engage the newly appointed directors of these Product Offices, who were faced with defining their roles and responsibilities essentially from a blank sheet – a so-called “white space” job. A major topic for discussion was the selection of intermediate-internal products to serve as pilot products as the Product Office started to develop its processes. It was clear that certain spacecraft subsystems were more amenable to platforming, given their technical nature and lower susceptibility to some of the implementation challenges previously described. Another important topic was interaction between the Product Offices and Program Offices. The author used his framework to illustrate conceptually how independent tiers of product management can manage each level of products with different development criteria. Still, an overarching consideration is establishing criteria that govern the interface between the Product Offices and Program Offices. For example, under what circumstances does the Program Office have the priority to insist upon a customized design that deviates from a planned product line? Or does the Product Office’s platform planning have to regularly yield to the specific needs of each Program Office?

It will likely require several test cases to determine the answers to these and related questions. Unfortunately, the author was not in a position to observe or participate in an example test case over the course of his internship. Given the inherently long development cycle for satellites, the role of the Product Office may take a few years to mature. In the short-term, the Product Offices can certainly provide greater accountability for intermediate-internal products, which is a vital first step. The longer-term measure of effectiveness is the extent to which they can drive commonality into subsystems and components that are utilized in multiple satellite programs.

4.2. Future Research Opportunities

A logical extension to this thesis and important area for future research is to expand the definition of an enterprise when considering architectural “fit.” The first step is to include the other two categories of organizational processes in the enterprise architecture and investigate how they relate to product architecture and contribute to value delivered to customers and other stakeholders. Recall that this thesis only considered the life cycle processes and not the enabling

infrastructure and enterprise leadership processes. To be clear, NGST's realignment did not leave their former enabling process areas (e.g. human resources) unchanged.

A second step is to take a further expanded view that includes key customers, suppliers, and partners. The so-called *extended enterprise* considers the complete collection of cooperating organizations that constitutes an enterprise supersystem. In this thesis, boundaries were drawn around a central firm, creating external interfaces with suppliers and customers through which supply chain managers and program managers acted. The extended enterprise internalizes these interactions with suppliers and customers and more closely considers the processes that exchange information, financial, and physical resources between them and the central firm. This approach is highly applicable in the context of the satellite industry and the greater aerospace industry. As mentioned, several government contracts are typically awarded in the development of a complete space system. Teams of contractors bid for a particular segment, and the roles of prime integrator and major subcontractor are established within each team. It is quite often the case that a partner for one proposal may be a competitor for another one. Clearly, the question of architectural "fit" is relevant to the contractor team as a whole, which represents an extended enterprise of cooperating organizations. For NGST, the extended enterprise might be either the contractor team for a particular satellite program or the greater Northrop Grumman Corporation.

Overall, the ideas presented in this thesis are a small piece of the growing field of enterprise architecting and the greater discipline of engineering systems. Future research should continue to formalize the relationships between products, processes, and organizations by extending the boundary of the enterprise system.

This page has been intentionally left blank.

References

In the text, references are cited in the following format:
(<First Author's Last Name> <Year of Publication>, <Page Number>)

"About Us." Northrop Grumman Corporation website, May 9, 2008.
http://www.northropgrumman.com/about_us/index.html.

"reorganization." *The American Heritage Dictionary of the English Language, Fourth Edition*.
Houghton Mifflin Company, 2004.

Adamsen, Paul B., II. *A Framework for Complex System Development*. Boca Raton: CRC
Press, 2000.

Browning, Tyson R., John J. Deyst, et al. "Adding Value in Product Development by Creating
Information and Reducing Risk." *IEEE Transactions on Engineering Management*. Vol. 49,
No. 4, November 2002, 443-458.

Chase, James P. "Value Creation in the Product Development Process." SM Thesis.
Massachusetts Institute of Technology, 2001.

Cusumano, Michael A., and Kentaro Nobeoka. *Thinking Beyond Lean*. New York: The Free
Press, 1998.

Dori, Dov. *Object-Process Methodology*. New York: Springer-Verlag, 2002.

Larson, Wiley J., and James R. Wertz. *Space Mission Analysis and Design, Third Edition*. El
Segundo: Microcosm Press, 1999.

Meyer, Marc H., and Alvin P. Lehnerd. *The Power of Product Platforms: Building Value and
Cost Leadership*. New York: Free Press, 1997.

Millard, Richard L. "Value Stream Analysis and Mapping for Product Development." SM
Thesis. Massachusetts Institute of Technology, 2001.

Miller, George A. "The Magical Number Seven, Plus or Minus Two: Some Limits on our
Capacity for Processing Information." *Psychological Review*. Vol. 63, 1956, 81-97.

Murmann, Earll, et al. *Lean Enterprise Value*. New York: Palgrave, 2002.

Nightingale, Deborah J., and Donna H. Rhodes. "Enterprise Systems Architecting: Emerging
Art and Science within Engineering Systems." *MIT Engineering Systems Symposium*. March
2004.

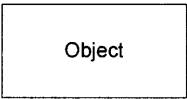
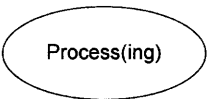
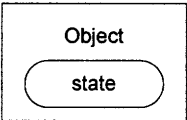
- Piepenbrock, Theodore F., Charles H. Fine, and Deborah J. Nightingale. "Toward a Theory of the Evolution of Business Ecosystems: Enterprise Architectures, Competitive Dynamics, Firm Performance and Industrial Co-Evolution." MIT Engineering Systems Division working paper, 2008.
- Rechtin, Eberhardt. *Systems Architecting: Creating and Building Complex Systems*. Englewood Cliffs: Prentice Hall, 1991.
- Rechtin, Eberhardt. "The art of systems architecting." *IEEE Spectrum*. October 1992, 66-69.
- Rechtin, Eberhardt, and Mark W. Maier. *The Art of Systems Architecting*. Boca Raton: CRC Press, 2000.
- Rechtin, Eberhardt. *Systems Architecting of Organizations: Why Eagles Can't Swim*. Boca Raton: CRC Press, 2000.
- Robertson, David, and Karl Ulrich. "Product Platform Development." *Sloan Management Review*. Vol. 39, Iss. 4, 1998, 19-31.
- Rouse, William B. "Enterprises as Systems: Essential Challenges and Approaches to Transformation." *Systems Engineering*. Vol. 8, No. 2, 2005, 138-150.
- Saloner, Garth, Andrea Shepard, and Joel Podolny. *Strategic Management*. New York: John Wiley and Sons, Inc., 2001.
- Slack, Robert A. "The Application of Lean Principles to the Military Aerospace Product Development Process." SM Thesis. Massachusetts Institute of Technology, 1998.
- Suh, Nam. P. *The Principles of Design*. New York: Oxford University Press, 1990.
- Ulrich, Karl T., and Steven D. Eppinger. *Product Design and Development, Third Edition*. New York: McGraw-Hill Inc., 2004.
- Womack, James P., and Daniel T. Jones. *Lean Thinking, 2003 Edition*. New York: The Free Press, 2003.

Appendix A

This appendix provides a short introduction to the Object-Process Methodology (OPM) that is presented in *Dori* (2002). OPM provides a set of practices, procedures, and rules to “express the function, structure, and behavior of systems in an integrated, single model” (Dori 2002, 4). It includes two modes: (1) graphical descriptions of systems called Object-Process Diagrams and (2) natural language sentence descriptions of systems called Object-Process Language (the latter has not been used in this document). The goal of OPM is to establish a common, rigorous modeling framework for understanding and developing systems of all kinds.

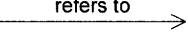
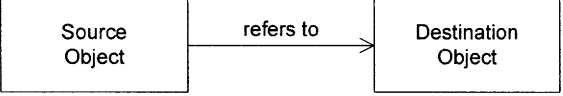
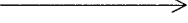


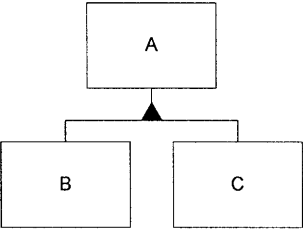

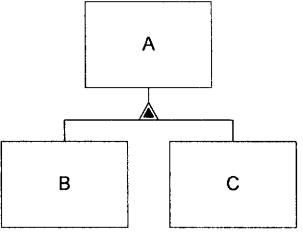

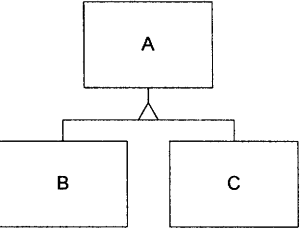
The three tables that follow name and describe the items and their symbols that are used to construct OPDs: (1) *entities* are the building blocks, (2) *structural links* show relationships between objects, and (3) *procedural links* show relationships between objects/states and processes.²¹ This is not an exhaustive list of OPD items, but rather provides a legend for those items that appear in the figures of this document. For a comprehensive catalog and explanation of the OPD symbology, please refer to Dori’s *Object-Process Methodology*.

Entities

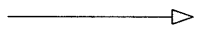
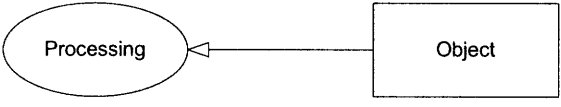
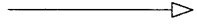


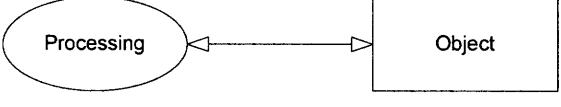
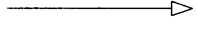
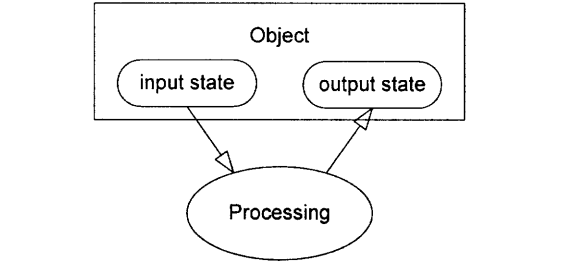

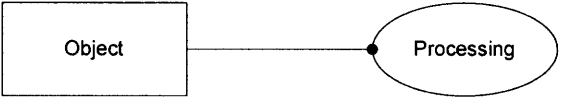

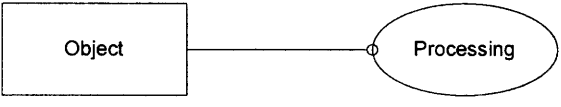
<i>Name</i>	<i>Symbol</i>	<i>Definition</i>	<i>Description</i>
Object		An object is a thing that has the potential of stable, unconditional physical or mental existence.	Static things. Can be changed only by processes.
Process		A process is a pattern of transformation that an object undergoes.	Dynamic things. Are recognizable by the changes they cause to objects.
State		A state is a situation an object can be at.	States describe objects. They are attributes of objects. Processes can change an object’s state.

²¹ These three tables are adapted from the quick-reference tables that are printed at the front and rear of the hardcover edition of *Dori* (2002).

Structural Links

<i>Name</i>	<i>Symbol / Usage</i>		<i>Meaning</i>	<i>Description</i>
Tagged			A connection or an association between things that holds irrespective of time.	Relation from source object to destination object; relation is recorded along link.
				
(Null)				Relation from source object to destination object with no tag.
				
Aggregation			Relates a whole to its parts	B and C are parts of the whole A.
Exhibition			Relates an exhibitor to its attributes	B and C are attributes of A.
Generalization			Relates a general thing to its specialization	B and C are types of A.

Procedural Links

<i>Name</i>	<i>Symbol / Usage</i>	<i>Description</i>
Transforming Links		
Consumption		Process uses object up entirely during its occurrence.
		
Result		Process creates an entirely new object during its occurrence.
		
Effect		Process changes the state of the object in an unspecified manner.
		
Input / Output		The object is at input state prior to the process occurrence, and at output state as a result of its occurrence.
		
Enabling Links		
Agent		Object is a human that is not changed by the process; process needs the agent object in order to occur.
		
Instrument		Object is a non-human that is not changed by the process; process needs the instrument object in order to occur.
		

This page has been intentionally left blank.