Business Process Improvement Using Axiomatic Design and Object-Process Methodology

by

Jason M. Casebolt

J.D., Seattle University School of Law, 2010

B.S. Business Administration, The Ohio State University, 2003

Submitted to the Faculty in partial fulfillment of the requirements for the degree of

Master of Science in Engineering and Management

at

Massachusetts Institute of Technology

June 2016

Author: .... Signature redacted .....

Jason M. Casebolt
System Design and Management Program
May 6, 2016

Certified by: .... Signature redacted ....

Dov Dori
Thesis Supervisor
Professor, Information Systems Engineering, Technion, Israel Institute of Technology
MIT Research Affiliate

Accepted by: .... Signature redacted

Patrick Hale
Chairman & Senior Lecturer, Engineering Systems Division
Director, System Design and Management Program

ACKNOWLEDGEMENTS

To my family, colleagues, mentors, and supporters who were my sources of inspiration for my studies at MIT . . .

Family: *First and foremost to my wife Leanne. Her encouragement and engagement with our son Jackson (who was born 2 weeks after beginning the program) enabled our family to function while I concentrated on MIT learning. Her patience and perseverance made earning this education truly a team-effort. To my parents, Mac and Dolores and their never-wavering support and encouragement that has enabled me to reach every goal I have ever set for myself. To Ken Magness, who like my brother, looked after my wife and son during my trips to Boston.*

Supporters: *To insightful and supportive leaders, Lindsay Anderson (LGO, '93) and Lisa Slagle, who supported my goal to learn better problem solving at MIT. To advocates Jose Garza, John Byrne, and Steve Patneaude (SDM, '06); as well as legal educators W.H. ("Joe") Knight and Thomas Antkowiak, whose enthusiastic mentoring and encouragement enabled me to combine "legal thinking" with "systems thinking."*

MIT Community: *To the educators in the MIT System Design & Management program, particularly Pat Hale, who welcomed me into MIT, and Dov Dori, who mentored me on both this thesis and Object-Process Methodology.*

Cohort Teammates: *To Chris Gardner, Kory Miller, David Parry, Eli Grun, Drew Hall, and Andrew Walz whose support and wit made the most out of long hours of learning.*

Believers: *Finally, to Tyler Staal and Justin Franke, who both believed in me enough to push me farther than I had ever thought possible.*

. . . I wouldn't have started or finished this journey without all of you.

Business Process Improvement Using Axiomatic Design and Object-Process Methodology

By

Jason M. Casebolt

Submitted to the System Design and Management Program on May 6, 2016 in Partial fulfillment of the Requirements for the Degree of Master of Science in Engineering and Management.

Abstract

This thesis introduces AD-OPM BPI, which is a new method of conducting business process improvement using both Axiomatic Design and Object-Process Methodology. The premise underlying the method is that modern process improvement techniques boast large efficiency gains, but fail to address the broader process system. Through first using Axiomatic Design to map and optimize the process system, broader-inefficiencies will be addressed before they constrain individual processes. Then Object-Process Methodology is conducted for process-specific optimization by utilizing modern system architecture layering principles to identify non-value-adding entities and improve them through deletion or simplification.

A case study at a large aerospace manufacturing company demonstrates the method in practical application. Results suggest that application is better suited to new or small-scale systems due to the challenge of applying Axiomatic Design to pre-existing large scale systems. Despite this limitation, Object-Process Methodology remains a viable option for business process improvement, whether or not it is coupled with Axiomatic Design in AD-OPM BPI.

Thesis Supervisor: Dov Dori

Title: Professor, Information Systems Engineering

Technion, Israel Institute of Technology

MIT Research Affiliate

Table of Contents

Chapter 1: Introduction

Emerging as a major focus in the 1980's and 1990's, Business Process Improvement ("BPI") has been a steady undertaking for major companies around the globe (Harrington 1991) (Harmon and Wolf 2014). The definition of BPI used here is simply "*improvement of a process [by] means [of] changing a process to make it more effective, efficient, and adaptable*" (Harrington 1991, 133). The leading drivers of this movement have been the need to save money and to improve performance. Additional motivations include increasing customer satisfaction, improving organizational responsiveness, complying with regulations, such as Sarbanes-Oxley, and major events, like a merger or an acquisition (Harmon and Wolf 2014). These efforts have made BPI a big business, with process improvement departments, consultants, and practitioners who focus a large part of their time or resources on improving business processes. They use popular products and methods, such as Lean, Six Sigma, Business Process Reengineering, Workflow, ERP software, and Business Process Management Suite software.

Despite over 20 years of focus, companies are still spending substantial sums on process improvement each year. For example, a 2013 survey of over 300 large companies revealed that 46% spent at least $500,000 that year on process improvement efforts (Ibid., 22). Nearly half of those companies (26% of the overall total) spent at least $1 million. Of all companies surveyed, 31% classified BPI as a major strategic commitment (Ibid., 12).

The purpose of this thesis is to apply "*systems thinking*" by using Model-Based Systems Engineering ("MBSE"), specifically Axiomatic Design ("AD") and Object Process Methodology ("OPM"), to perform a new method of BPI (collectively "AD-OPM BPI"). This approach treats a set of processes as a system to be engineered, and each individual process as a product to be architected with considerations for requirements, form, and function.

The rest of this thesis is organized as follows: Chapter 2 presents a critique of the current state of BPI, to which this thesis addresses. Chapter 3 introduces AD. Chapter 4 describes the first part of the AD-OPM BPI method, where AD is used to perform system analysis and optimization. Chapter 5 introduces OPM. Chapter 6 describes the second part of the AD-OPM BPI method, where OPM is used to perform individual process analysis and optimization. Chapter 7 applies the AD-OPM BPI method on a real-life case study at a large manufacturing company. Chapter 8 summarizes the results and conclusions.

Chapter 2: Critique of the Current State of BPI

Processes are a key aspect of contemporary systems engineering ("SE") theory and practice; however, SE theory and practice are currently not key aspects of process design or improvement. "*While many have tried to settle the debate by providing process modeling standards and tools, most of them have fallen short in important areas, particularly when it comes to treating processes as systems and using SE approach in their development*" (Browning, Fricke and Negele 2006, 105). Processes alone are not systems—systems comprise objects transformed by processes. Processes do not happen in vacuum—they transform objects within context of a system, and this is where value is created. Neglecting objects in a process model is a major cause of the inadequacy of the process model techniques and the reason why they fall short of providing good process improvement tools. Neglecting a systems-of-systems approach, that combines objects with the processes to be improved, is contrary to the gains that are being targeted with all of the resources being allocated to the field.

Linkage Does Not Demonstrate Conscious System Design

Consistent with foregoing a SE approach, current practices to demonstrate process compliance result in clear linkages between related processes to reveal structure (Damelio 2011). Structure is expressed by objects, which are things that exist. This is in contrast to processes, which are things that occur. While structure can be demonstrated and verified, best practices do not yield an architecture that demonstrates a conscious design, or even purpose, of the process system. Even the term "*process system*" is a misnomer as there is no system without the other objects that a process transforms, creates, consumes or changes the state of.

The difference between the two concepts of existing linkage and preferred architecture is that linkage only shows relationships "*that give systems their added value*," while system architecture reveals the "*interfaces among activities*" that reflect the conscious design of the system (Browning, Fricke and Negele 2006, 109).

Examples of current practices that prioritize linkage over architecture are process trees (types of relationship maps) and compliance matrices that are popular methods to demonstrate compliance (Damelio 2011) (Page 2000). Simple examples of each are provided in Figures 2.1 and 2.2 respectively. Though each have different formats, both methods convey the simple relationships between the processes involved. Other than connectivity, neither process trees nor compliance matrices provide a clear demonstration of the design of the system, context of the requirements of the process system, waste beyond requirements that is present within the system, or demonstration that architecture satisfies the intended function.

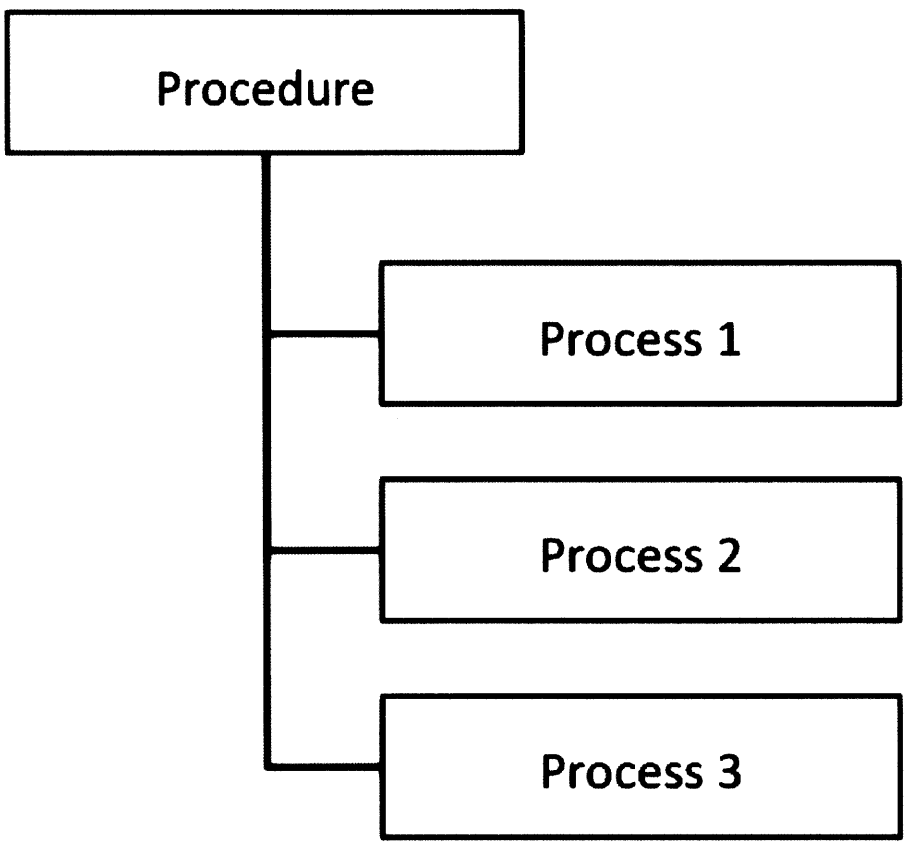

Figure 2.1: Simple Process Tree

Policy	Procedure	Process
Policy-1	Procedure-1 Procedure-3 Procedure-5	Process-2

Figure 2.2: Simple Process Tree

Focus is on the Scope of a Process Instead of the Scope of the System

Another critique of modern process improvement methods is that the focus is on single or a limited set of processes. This often occurs through Lean, Six Sigma, or other process mapping methods that select individual processes, or process groups, for which there is a concern (LeanOhio). By only reviewing individual processes and without reviewing the broader system, higher-tier design decisions that constrain lower-tier processes are also not examined (Suh 1998). This means that higher-tier system constraints will continue to constrain the lower-tiers. Therefore, while methods focused on individual processes may yield up to 50% or more efficiency improvement, the broader system remains unaddressed, incrementally improved, and sub-optimal until the higher-tier system design decisions are addressed.

Mapping Methods Are Too Vague to be Effective

Another critique of the current state of process improvement is that popular mapping approaches are not really effective for optimizing a system of processes. This seems like a contradiction given the objective results that routinely are part of such improvement efforts (LeanOhio). But there are aspects that have "*caused debates, misunderstandings, and wastes* *of time and money in industry, not to mention a failure to realize the advantages provided by a really useful process model*" (Browning, Fricke and Negele 2006, 109-110). The distinction is that models provided system and process context, where other modern methods simply do not.

Ironically, detailed process models with enough system context to be useful are disfavored by some companies. This is due, in part, to an incentive for companies "*to keep the models purposefully ambiguous so that when a process conformance auditor shows up, any actions by the workforce will in high likelihood be found to fall under the large and loose umbrella of a vague process description*" (Ibid.). Therefore, many companies consciously or unconsciously tradeoff better understanding of business processes rather than risk exposing auditors to a visualization of the complexity of their process system.

Critique Summary

Processes will continue to be a source of inefficiency for companies as long as the systems that comprise them are ignored. Without considering the conscious design and scope of the system, process improvement efforts remain too localized to be effective on a larger scale. As current practices of process improvement bifurcate efficiency increases to lower tiers, companies are incentivized to embraced ambiguity at the higher-tier levels to obscure the design rather than risk audit findings that could be associated with poorly demonstrating the system due to excessive-complexity of a process model. To address this tension, the AD-OPM BPI method is presented as a means to optimize both the process system and individual processes, while increasing audit compliance by clearly demonstrating how the process design choices clearly fulfill process system requirements.

Chapter 3: Introduction to Axiomatic Design

Axiomatic Design ("AD") is a general systems design methodology developed by Professor Nam P. Suh. This method of design embodies the nature of engineering, as it "*consists of synthesis and analysis, which mutually reinforce each other in a feedback loop*" (Suh 2001, xv). While there are many possible synthesis tools, Axiomatic Design enables the "*application of scientific principles and rigorous mathematical tools*" to demonstrate that a certain system design is "*good*" without having to rely purely on a subjective opinion (Ibid.).

This method bases the design of system architecture on hierarchical maps that comprise, amongst other inputs, Functional Requirements ("FRs") and Design Parameters ("DPs") (Suh 1998). The goal of these maps is to analyze the decisions that are made in a design and demonstrate how the FRs and DPs map to each other. This FR-to-DP mapping is then compared to design axioms, theorems, and corollaries that respectively demonstrate the "fundamental truths" of good design, inferences that are made from axioms, and concepts that can be proven through mathematical argument (Ibid., 205). To the extent that axioms, theorems, and corollaries demonstrate that the design can be improved, then a redesign using AD will be more consistent with those concepts and will reveal either a "*good design*" if the redesign satisfies the axioms, or a "*better*" or "*best*" design if the redesign remedies inefficiencies consistent with the theorems and corollaries (Ibid., 207-209).

Chapter 4: System Analysis and Optimization with Axiomatic Design

The process of performing Axiomatic Design involves four steps: (i) defining the FRs of the system, (ii) mapping between the domains, (iii) evaluating the independence of system functions, and (iv) evaluating the information content to validate "*best design*" (Ibid., 191-193). In what follows, these steps are discussed in depth.

Step 1 – Define the FRs of the System

The FRs are the "*minimum set of independent requirements that completely characterize the functional needs of the product [or other system] in the functional domain. By definition, each FR is independent of every other one at the time the FRs are established*" (Ibid., 205). In the context of traditional engineering, FRs represent the Customer Needs ("CNs") or attributes that must be satisfied by the system. Concurrently, the Constraints ("Cs") are also identified, as the fulfillment of the CNs in the functional domain will be shaped by the presence of Cs. The FRs are formed in a solution-neutral manner, meaning that they should not be written with any particular solution in mind so that designers can creatively find multiple ways of fulfilling the FRs (Ibid.). A solution-specific FR therefore becomes a C to the rest of the design, thus reinforcing the practice of resisting solution-specific concepts until absolutely necessary.

Step 2 – Mapping Between the Domains

After the FRs are identified in Step 1, the corresponding DPs must be mapped. This mapping links the requirements of the system onto the physical or otherwise functional domain that manifests it (Ibid., 191). DPs are chosen through creativity, but are bounded by the need to satisfy FRs and limited by the presence of Cs. Similarly, each DP that is created becomes a constraint to each subsequent tier of progressive design that flows from within it.

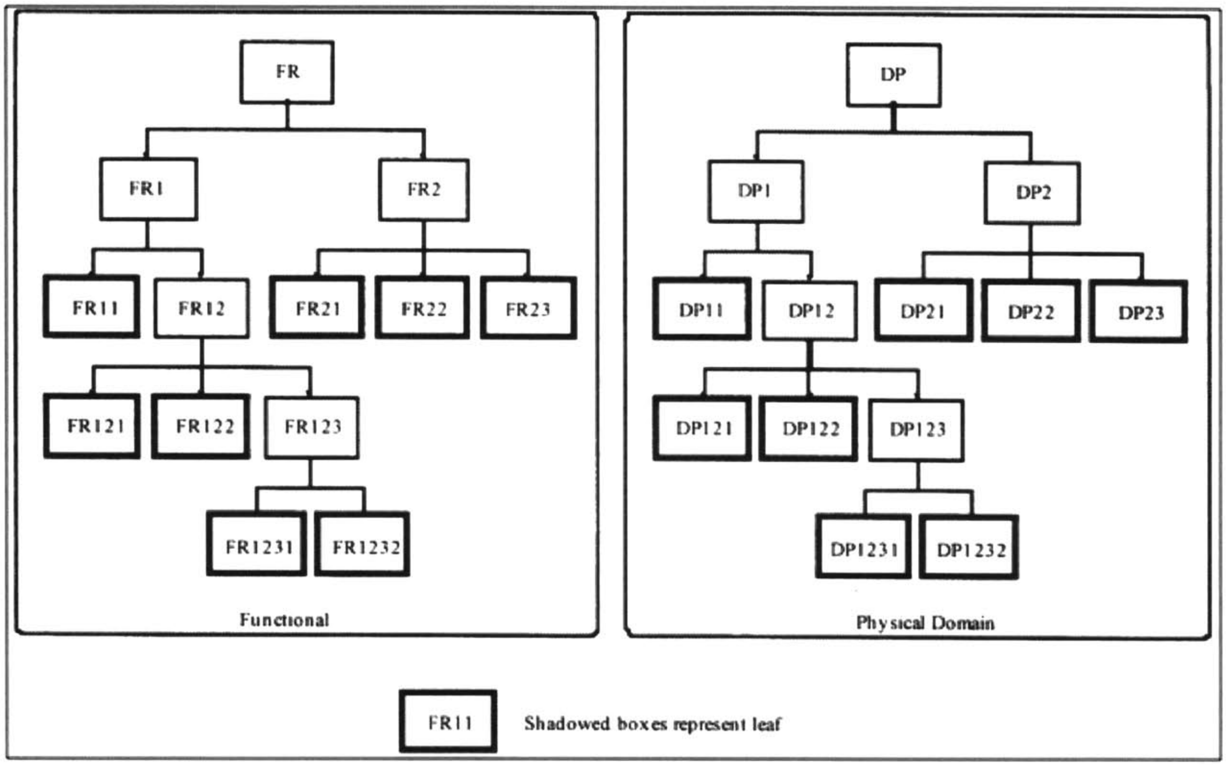

Figure 4.1: Concept of Axiomatic Design (Suh 1998, 195)

Mapping can take several forms, both graphically and mathematically, but the thought process behind it remains constant. The example in Figure 4.1 shows the relationships of a one-for-one FR-to-DP design. As the solution-neutral FR becomes more defined, so does the design embodied by the DPs. This continues to the subsequent tiers of requirements and design, until the "*leaves*" are represented. The "*leaves*" are displayed with bold outline, where each "*leaf*" indicates an "*FR that does not need further decomposition*" (Ibid., 195). DP "*leaves*" satisfy respective FRs, and a higher-tier FR or DP is satisfied by combining the respective "*leaves*" that tier off from it.

To transition from the top-tier FR-DP concept to the lower-tier FR-DP leaves requires a technique called "*zig-zagging.*" Professor Suh "*observed that when you reason about a system, you alternate reasoning in the form domain and the function domain. One tends to start in one* *domain and work as long as practical, and then switch to the other*" (Crawley, Cameron and Selva 2015, 44). Applied here in Figure 4.2, each level of DPs constrains the tiers of FRs and DPs that satisfy it. For example, "*zig-zagging*" occurs when as much of the Tier-1 DP has been defined, and to define any further requires "*zig-zagging*" to the FR domain to define the Tier-2 requirements, from which the respective Tier-2 DP's can be designed. This repeats through each tier, with the FRs and DPs starting as solution-neutral and then becoming more narrowly constrained at each interval until the design becomes solution-specific "leaves."

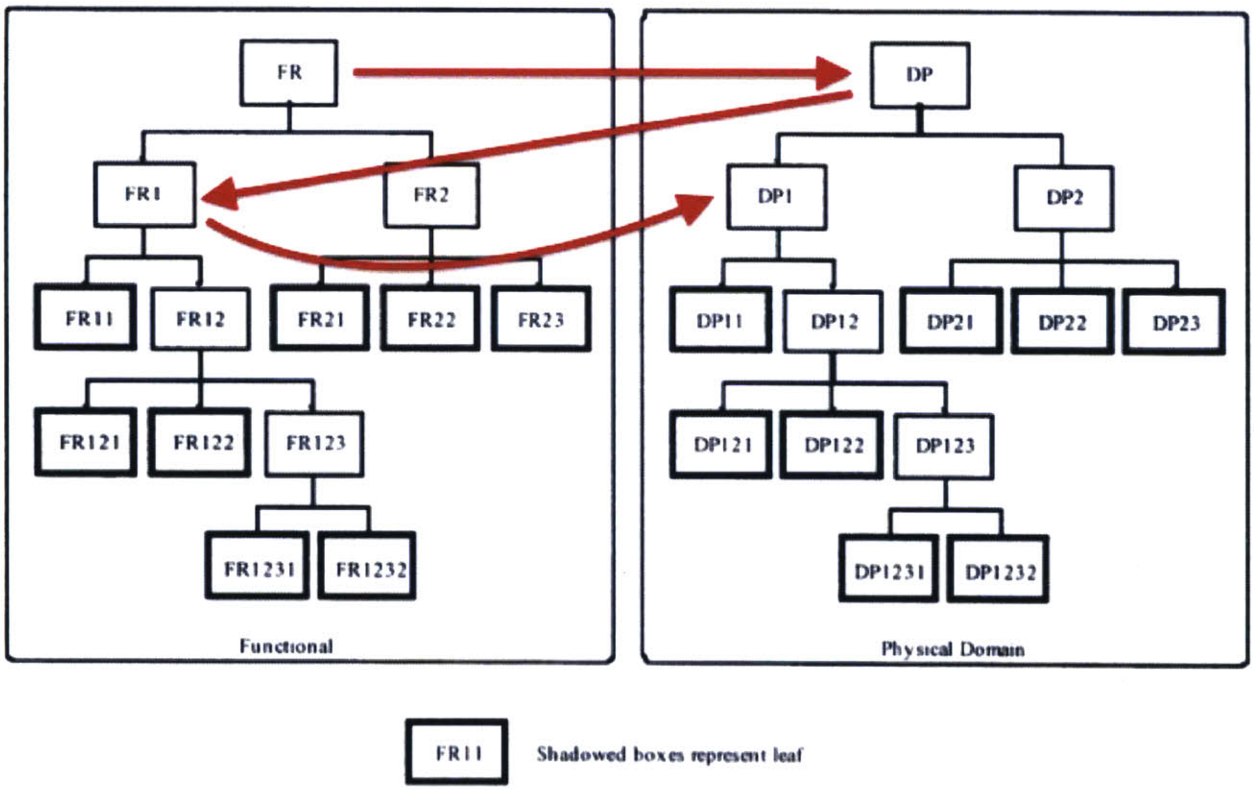

Figure 4.2: "Zig Zagging" Concept (Adapted from Suh 2001, 30)

Axiom 1: The Independence Axiom
Maintain the independence of the Functional Requirements (FRs)

The third step of applying Axiomatic Design is to evaluate the DPs and FRs against Axiom 1: The Independence Axiom. The Independence Axiom is satisfied when either (i) "*each of the FRs can be satisfied independently by means of one DP*," or (ii) FRs are not satisfied independently by means of one DP, but independence is still achieved "*if the DPs are changed in the proper sequence*" (Suh 1998, 192). The former is referred to as an uncoupled design and the latter represents a decoupled design, both of which are desired by designers. All other designs are considered coupled designs, which violate the Independence Axiom and are undesirable to designers.

An uncoupled design results in a design matrix where the relationship of FRs to DPs is one-one, and where the requirements for independent sub-function can be satisfied by any sequence in the design. Another term for this is a diagonal design matrix, where the relationships are represented only on the matrix's diagonal (Ibid.). Examples of an uncoupled design concept and matrix are provided in Figures 4.3 and 4.4.

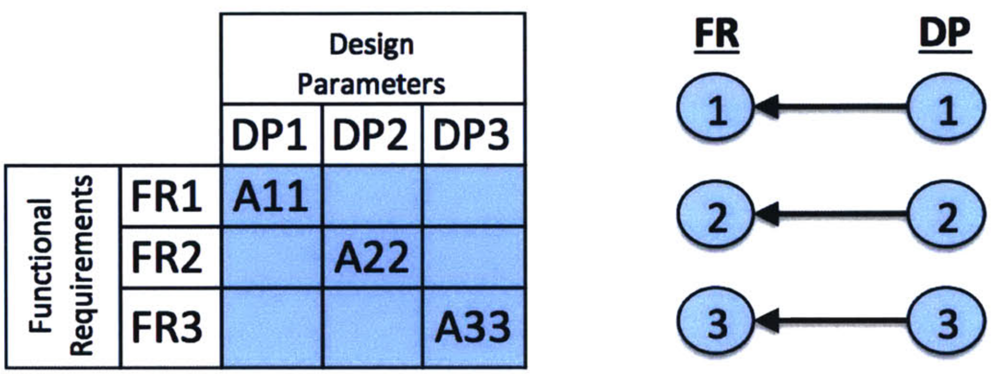

Figure 4.3: Uncoupled Design Concept (Adapted from Van Eikema Hommes 2015)

					Design Parameters	DP																		
						DP1									DP2									
						DP11	DP12	DP13							DP21	DP22	DP23	DP24			DP25			
						Description11	Description12	DP131 Description131	DP132 Description 132	DP133 Description 133	DP134 Description134	DP135 Description135	DP136 Description136	DP137 Description137	Description21	Description22	Description23	DP241 Description241	DP242 Description242	DP243 Description243	DP251 Description251	DP252 Description252	DP253 Description253	DP254 Description254
Functional Requirements						1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
FR	FR1	FR11		Description11	1	1																		
		FR12		Description12	2		2																	
		FR13	FR131	Description131	3			3																
			FR132	Description 132	4				4															
			FR133	Description 133	5					5														
			FR134	Description134	6						6													
			FR135	Description135	7							7												
			FR136	Description136	8								8											
			FR137	Description137	9									9										
	FR2	FR21		Description21	10										10									
		FR22		Description22	11											11								
		FR23		Description23	12												12							
		FR24	FR241	Description241	13													13						
			FR242	Description242	14														14					
			FR243	Description243	15															15				
		FR25	FR251	Description251	16																16			
			FR252	Description252	17																	17		
			FR253	Description253	18																		18	
			FR254	Description254	19																			19

Figure 4.4: Uncoupled Design Matrix (Adapted from Suh 2001, 282-283)

A decoupled design results in a design matrix where the relationship of FRs to DPs is more than one-for-one, but where independent sub-function still occurs when a proper sequence enables the DPs to satisfy the FRs in a manner similar to a diagonal design matrix. This can also be referred to as a lower triangular design matrix, where several FR-to-DP associations occur only on the diagonal and in the lower-left of the matrix (Suh 1998). Here, the absence of associations in the upper-right of the matrix indicates that sequencing of the remaining DPs that fall on or under the diagonal will fulfill the FRs in a manner as functionally effectively as a matrix that contains only diagonal inputs. Examples of a decoupled design concept and matrix are provided in Figures 4.5 and 4.6.

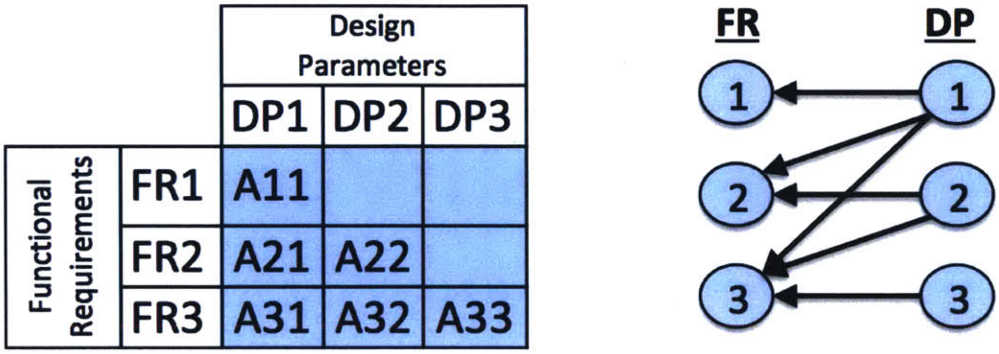

Figure 4.5: Decoupled Design Concept (Adapted from Van Eikema Hommes 2015)

				Design Parameters (DP)	DP1: DP11	DP1: DP12	DP1: DP13	DP1: DP13	DP1: DP13	DP1: DP13	DP1: DP13	DP1: DP13	DP1: DP13	DP2: DP21	DP2: DP22	DP2: DP23	DP2: DP24	DP2: DP24	DP2: DP24	DP2: DP25	DP2: DP25	DP2: DP25	DP2: DP25
							DP131	DP132	DP133	DP134	DP135	DP136	DP137				DP241	DP242	DP243	DP251	DP252	DP253	DP254
					Description11	Description12	Description131	Description 132	Description 133	Description134	Description135	Description136	Description137	Description21	Description22	Description23	Description241	Description242	Description243	Description251	Description252	Description253	Description254
Functional Requirements					1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
FR	FR1	FR11	Description11	1	1																		
FR	FR1	FR12	Description12	2		2																	
FR	FR1	FR13: FR131	Description131	3	1	1	3																
FR	FR1	FR13: FR132	Description 132	4	1	1		4															
FR	FR1	FR13: FR133	Description 133	5	1	1		1	5														
FR	FR1	FR13: FR134	Description134	6	1	1		1	1	6													
FR	FR1	FR13: FR135	Description135	7	1	1		1	1	1	7												
FR	FR1	FR13: FR136	Description136	8	1	1		1	1	1	1	8											
FR	FR1	FR13: FR137	Description137	9	1	1		1	1	1	1	1	9										
FR	FR2	FR21	Description21	10									1	10									
FR	FR2	FR22	Description22	11									1	1	11								
FR	FR2	FR23	Description23	12			1	1	1	1	1	1	1			12							
FR	FR2	FR24: FR241	Description241	13						1				1	1	1	13						
FR	FR2	FR24: FR242	Description242	14						1								14					
FR	FR2	FR24: FR243	Description243	15						1							1	1	15				
FR	FR2	FR25: FR251	Description251	16							1			1				1		16			
FR	FR2	FR25: FR252	Description252	17							1			1				1			17		
FR	FR2	FR25: FR253	Description253	18	1	1	1				1			1				1			1	18	
FR	FR2	FR25: FR254	Description254	19	1	1	1				1			1				1			1	1	19

Figure 4.6: Decoupled Design Matrix (Adapted from Suh 2001, 282-283)

Finally, a coupled design matrix results where there are several FR to DP relationships that cannot be sequenced in manner that enables independent function. This is represented by a design matrix with associations that exist on both lower triangular and upper triangular sections, regardless of the sequence they are arranged (Suh 1998). This results in sub-functions that cannot be performed independently. Examples of a coupled design concept and matrix are provided in Figures 4.7 and 4.8.

		Design Parameters		
		DP1	DP2	DP3
Functional Requirements	FR1	A11	A12	A13
	FR2	A21	A22	A14
	FR3	A31	A32	A33

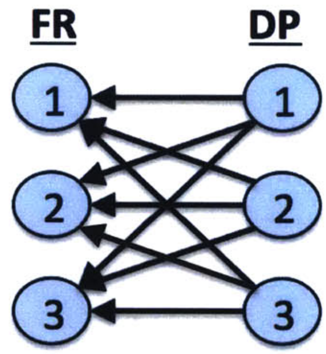

Figure 4.7: Coupled Design Concept (Adapted from Van Eikema Hommes 2015)

					Design Parameters	DP																		
						DP1									DP2									
						DP11	DP12	DP13							DP21	DP22	DP23	DP24			DP25			
								DP131	DP132	DP133	DP134	DP135	DP136	DP137				DP241	DP242	DP243	DP251	DP252	DP253	DP254
						Description11	Description12	Description131	Description 132	Description 133	Description134	Description135	Description136	Description137	Description21	Description22	Description23	Description241	Description242	Description243	Description251	Description252	Description253	Description254
Functional Requirements						1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
FR	FR1	FR11		Description11	1	1									1	1	1							
		FR12		Description12	2		2								1	1	1							
		FR13	FR131	Description131	3	1	1	3	1	1	1	1	1	1										
			FR132	Description 132	4	1	1		4					1										
			FR133	Description 133	5	1	1		1	5				1										
			FR134	Description134	6	1	1		1	1	6			1				1	1	1				
			FR135	Description135	7	1	1		1	1	1	7		1										
			FR136	Description136	8	1	1		1	1	1	1	8	1										
			FR137	Description137	9	1	1		1	1	1	1	1	9							1	1	1	1
	FR2	FR21		Description21	10	1	1							1	10	1								
		FR22		Description22	11	1	1							1	1	11								
		FR23		Description23	12	1	1	1	1	1	1	1	1	1			12	1						
		FR24	FR241	Description241	13						1				1	1	1	13						
			FR242	Description242	14						1								14					
			FR243	Description243	15						1							1	1	15				
		FR25	FR251	Description251	16							1			1				1		16	1	1	1
			FR252	Description252	17							1			1				1			17	1	1
			FR253	Description253	18	1	1	1				1			1				1			1	18	1
			FR254	Description254	19	1	1	1				1			1				1			1	1	19

Figure 4.8: Coupled Design Matrix (Adapted from Suh 2001, 282-283)

Axiom 2: The Information Axiom
Minimize the information content of the design

The second Axiom is the Information Axiom, which states that the information content of the design should be minimized. Professor Suh expands on this by noting that "*among all the designs that satisfy the Independence Axiom, the design that has the least information content is the best design*" (Suh 1998, 192). With this premise, the uncoupled or decoupled designs that satisfy Axiom 1: The Independence Axiom are considered "*good*" designs. Between them, the "*best*" design is the one that minimizes the information content.

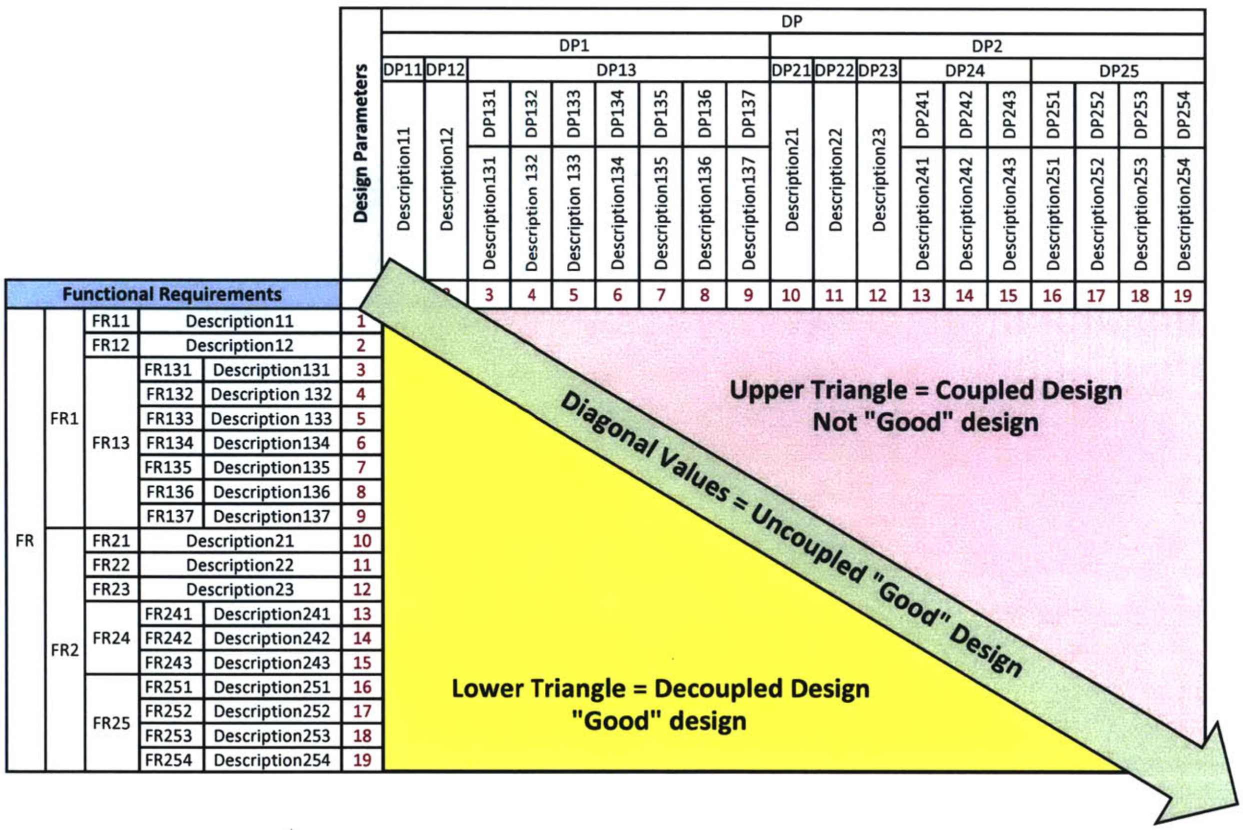

Figure 4.9: Visualization of "*Good*" Designs

The example in Figure 4.9 summarizes what Axiomatic Design characteristics are considered "*good*". Here, an undesired design is any design that has FR-DP coupling apparent in the upper triangle. Desired designs are those where FR-DP linkages appear purely on the diagonal and/or lower triangular values. It is between all designs that map to the diagonal or lower triangle values from which the information content will be evaluated to determine the best design.

The information content involved is "*simply the information needed to satisfy the highest functional requirements*" (Ibid., 192-193). This information content is presented in the context of the logarithmic probability of satisfying the functional requirements. For application of AD-OPM BPI presented in this thesis, the actual logarithmic computation is not necessary.

Summary

AD is useful for analyzing a system and demonstrating how its design fulfills its requirements. Through mapping the maturation from a solution-neutral concept through the design decisions that result in a solution-specific design, AD demonstrates more than just what a design is, but also answers the questions of how and why the design is made. Beyond more context, the combination of AD mapping and the two design axioms provides an opportunity to objectively define the characteristics of "good design", thereby highlighting a desired state to target through process system redesign efforts.

Chapter 5: Introduction to Object-Process Methodology

OPM is a leading MBSE platform due, in part, to its December 15, 2015 release by the International Organization for Standardization ("ISO") as the ISO-19450 specification for "Automation Systems and Integration – Object-Process Methodology" (Dori 2002) (International Standards Organization 2015c). Founded on the minimal ontology of stateful objects and processes that transform them as a set of necessary and sufficient building blocks, OPM is a holistic conceptual modeling language and cross-system lifecycle methodology, expressed graphically in a single kind of diagram and a complementary, auto-generated natural language text. It is different from other MBSE modeling languages in (i) the equal priority given to stateful objects and processes as the only two conceptual building blocks needed to represent systems in any domain – the minimal ontology, and (ii) the bimodal representation of the OPM model in both formal intuitive graphics and automatically generated text – simples sentences in a subset of English.

OPM is flexible in its application and has been applied in a wide array of industrial domains, from defense and avionics through electronic consumer appliances to software engineering, Web applications design, and molecular biology. OPM has been used in the evaluation of complex socio-technical system in fields such as aerospace, defense, information systems, medicine, sciences, and space exploration (Mordecai and Dori 2015). Formal yet intuitive, OPM is learned quickly and enables involving the customer as a partner, starting from the early product or system development phases all the way to deployment and maintenance, providing for the integration of risk and interoperability into the architecture and design of complex systems and systems-of-systems.

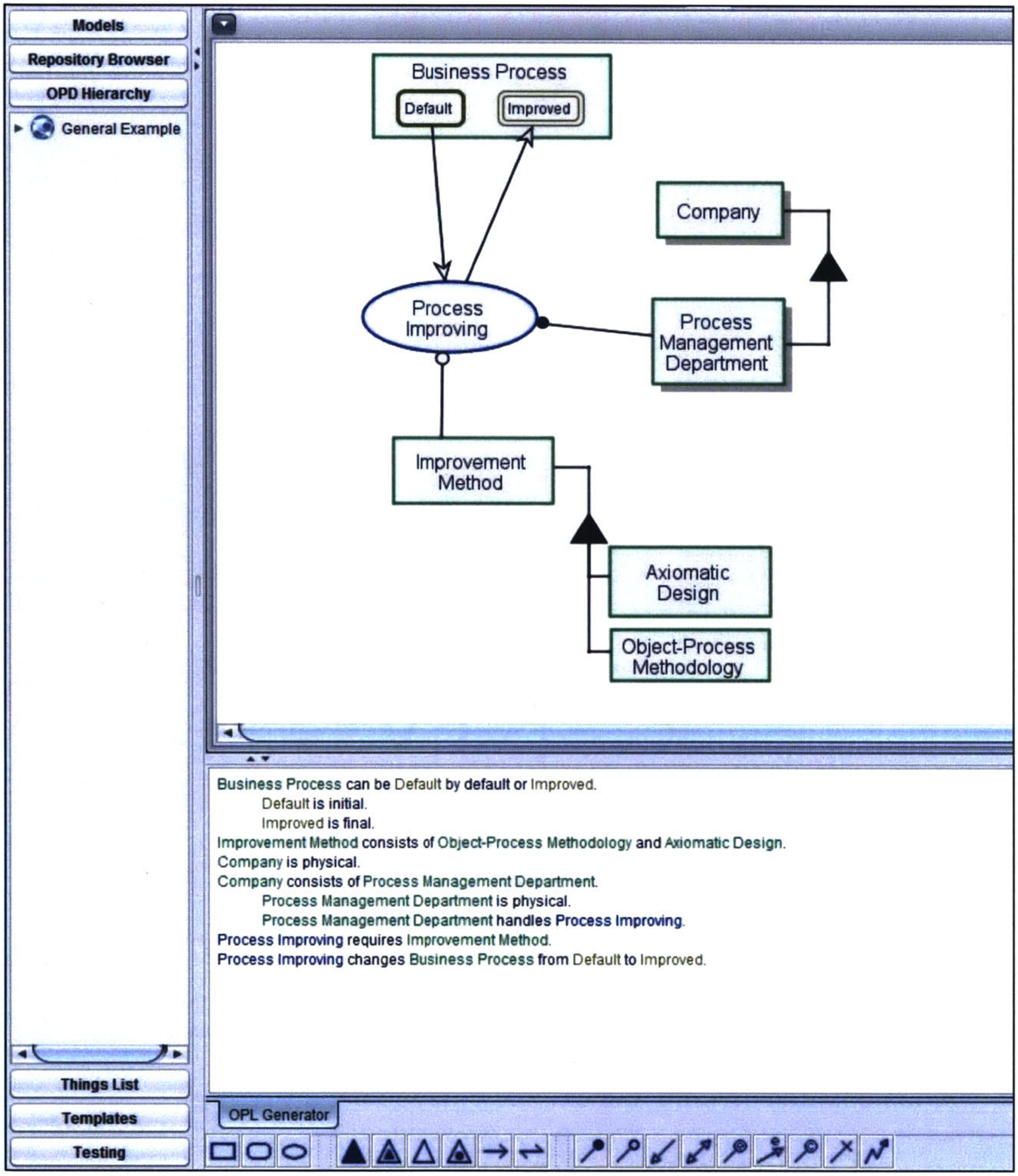

Figure 5.1: OPM Model of AD-OPM BPI Method

To use OPM, the freely available CASE tool OPCAT provides an environment that enables users to design OPM models, which are referred to as Object-Process Diagrams ("OPDs"). (Dori 2002) OPDs created in OPCAT automatically generate Object-Process Language ("OPL") text in a separate panel, which is a textual description of the OPD in a subset of English. In addition to model creation, OPCAT enables model simulation through executing the model for behavior verification and validation (Ibid.). Figure 5.1 demonstrates the concept of this thesis through a simple OPM model of **Process Improving** using **Axiomatic Design** and **Object-Process Methodology** that are both aggregated into an **Improvement Method**. This is visualized through OPCAT's OPD (top) and OPL (bottom) views.

The Building Blocks

	Visual Representation	Textual Form	Definition	Description
Entities	Object	Nouns; capitalized first letter in every word; if ending with "ing", "Object" is placed as a suffix	*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(ing)	Nouns in gerund form; capitalized first letter in every word; if not ending with "ing", "Process" is placed as a suffix	*A **process** is a pattern of transformation that an object undergoes.*	Dynamic things. Are recognizable by the changes they cause to objects.
	Object state	Nouns, adjectives or adverbs; non-capitalized	*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.

Figure 5.2: OPM Entities (Dori 2002)

Within OPM, a system is comprised of physical (tangible) or informatical (intangible) things—objects and processes—that are represented by rectangles and ovals respectively (Dori 2002). A key premise of OPM is that objects and processes are of equal importance and complement each other for providing a complete structural and procedural specification of the system. Objects are things that exist in some state, and are represented by nouns. Processes, represented by verbs, preferably in their gerund form (ending with "ing"), are things that transform objects through creating or destroying objects, or changing object states.

The Four Fundamental Structural Relations

Shorthand Name	Aggregation	Exhibition	Generalization	Instantiation
Symbol				
Meaning	Relates a whole to its parts	Relates an exhibitor to its attributes	Relates a general thing to its specializations	Relates a class of things to its instances

Figure 5.3: OPM Structural Relation Symbols (Dori 2002)

To supplement the objects, processes, and states, OPM supports structural and procedural relations, expressed graphically as links, as well as hierarchical organization for complexity management. The four fundamental structural links, represented and defined in Figure 5.3, are aggregation-participation, generalization-specialization, exhibition-characterization, and classification-instantiation.

Procedural Links

These links are generally used between an object and a process. They cannot be used to link objects together.

Link Name	OPD Symbol	OPL Sentence	Description
Consumption	Processing – Object	**Processing** consumes **Object.**	Process uses object up entirely during its occurrence.
Result	Processing – Object	**Processing** yields **Object.**	Process creates an entirely new object during its occurrence.
Effect	Processing – Object	**Processing** affects **Object.**	Process changes the state of the object in an unspecified manner.
Agent	Object – Processing	**Object** handles **Processing.**	Object is a human that is not changed by the process; process needs the agent object in order to occur.
Instrument	Object – Processing	**Processing** requires **Object.**	Object is a non-human that is not changed by the process; process needs the instrument object in order to occur.

Figure 5.4: OPM Procedural Links (Dori 2002)

While structural links connect objects to objects or processes to processes, procedural links connect processes to objects or to object states. Procedural links include transforming links (consumption, result, input-output, and effect), enabling links (agent and instrument), and control links (which are out of scope for this thesis). Consumption implies that the process consumes the object. Result links indicate that the process generates the object. An input-output link pair denotes that the process changes an object from an input state to an output state. The effect link denotes that the process changes the object without specifying the input and output states. These are demonstrated in Figure 5.4 (Ibid.).

Enabling links, also presented in Figure 5.4, denote objects that are needed for the process to occur but themselves are not transformed. The agent link expresses the fact that the agent (a human) enables the process. An instrument link denotes a non-human enabler.

As noted, beyond visualization, OPCAT generates OPL to evaluate the system through textual description in English (Ibid.). OPL has two purposes. First, it enables domain experts and systems architects to better analyze and design a system by providing a description-based model to validate or contrast their graphic-based OPD model. Second, OPL establishes a firm basis for automatically generating the designed application. An OPL example is displayed in the bottom portion of Figure 5.1.

OPM Summary

OPM is a dual approach that uses graphic-based modeling with text-based validation to construct a system. Through the freely available OPCAT software and the minimal number of selectable entities, OPM is easy to obtain, learn, and use. Despite its simplicity, it enables robust system exploration beyond architecture, including states, aggregation, and zooming within systems-of-systems. Its recent emergence as an international standard provides for its use as a consistent method for the foreseeable future.

Chapter 6: Process Analysis and Optimization with Object-Process Methodology

To perform the OPM portion of the AD-OPM BPI method, the design of a business process must be (i) decomposed, (ii) rationalized, and (iii) optimized. Modern systems architectural principles provide a basis from which OPM can be used for these purposes (Crawley, Cameron and Selva 2015, 121-122). This thesis applies these modern systems architectural principles to optimization or improvement of business processes.

1. Decomposition

The first step is to decompose the design into its entities so that it can be evaluated. Using OPM, each entity of the design is identified as either an object or a process. The focus of the first step should be accuracy of the identification, not the relationships between the objects and processes; relationship association will take place in the next step.

2. Rationalization

The second step is to rationalize the entities that were identified in Step 1, namely, to express meaningful and useful relations among them. With OPM, this involves connecting the objects and processes that were identified with structural and procedural relations. Modern systems architecting provides the basis for this linkage (Ibid.).

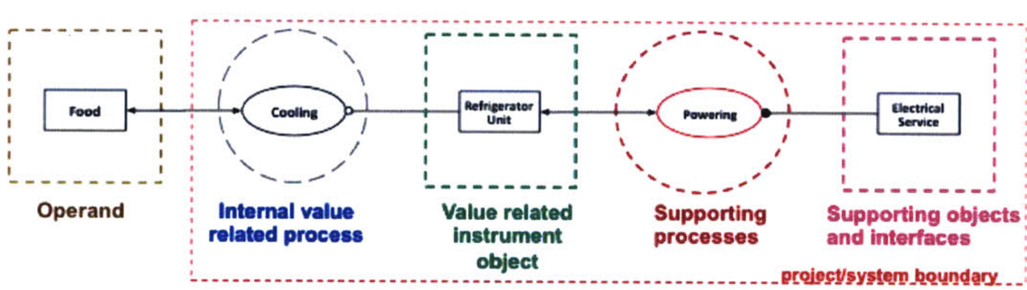

Figure 6.1: OPM-Based Layered Systems Architecture (Cameron 2014)

The concept of layered architecting within OPM is the starting point of performing BPI on individual processes. Following this approach, the system's objects and processes are identified and separated into the operand object – the major object transformed by the system, value-related objects and processes, and finally supporting processes and objects (Ibid.). Figure 6.1 provides an example of this rationalization approach, resulting in a layered architecture. This approach rationalizes not just the relationships, but also the value-adding role that each object and process plays in the context of the system's intended function.

3. Optimization

After rationalization is complete, the AD-OPM BPI method takes a different point of view than the layered systems architecture approach proposed by Crawley *et al.* (Ibid.). Where Crawley *et al.* suggests that supporting objects and processes provide structure that enables the value-related objects and processes to perform their respective functions, the AD-OPM BPI method maintains that the supporting objects and processes serve as both waste and complexity to a process performance. The concept portrayed in Figure 6.2 proposes that the operand, as well as the value-related objects and processes, are considered to be value-adding and are therefore desired. The non-value-adding waste that exists as the supporting objects and processes should be minimized or eliminated to reduce the process as much as possible to fulfilling its intended function, and therefore increasing efficiency.

The concept underlying this view is that the additional layers of architecture in a business or industrial process serve to complicate, add time, and otherwise hinder a process to perform its pure function. This departs significantly from product development, where such additional structures serve to support the system by design. The key is properly identifying value-adding entities as those that if removed would degrade or otherwise prevent the intended function of the business process from occurring. The remaining entities-- those that do not meet this standard—are therefore considered non-value-adding.

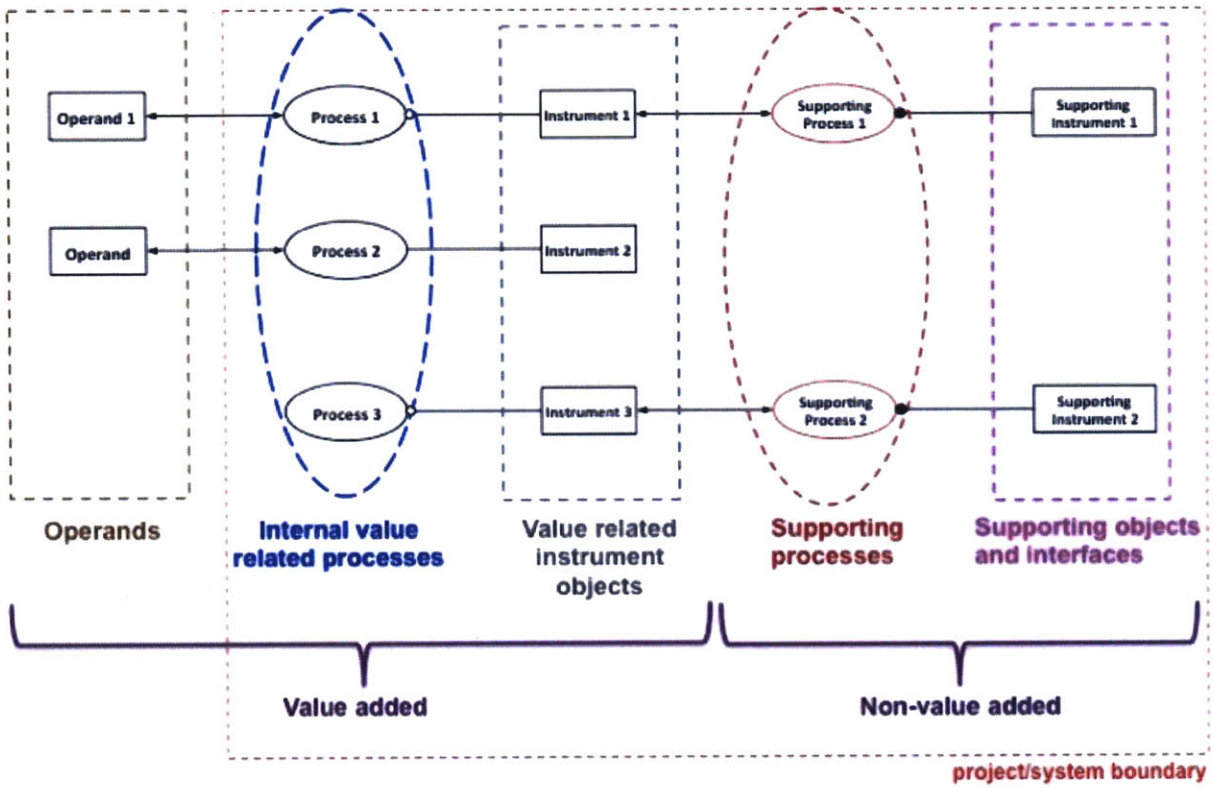

Figure 6.2: AD-OPM BPI Value Definition (Modified from Cameron 2014)

Once the entities have been rationalized into value-adding and non-value-adding objects and processes, then optimization occurs through any combination of one or more of the following solution-neutral actions: (i) delete, (ii) combine, (iii) reduce/simplify, (iv) automate, (v) offload/outsource, and (vi) upgrade. These are evaluated for each individual entity, as well as groupings of entities together. Similar to other guiding concepts, such as TRIZ, this AD-OPM BPI method narrows the focus of a user's creativity to improving certain aspects of a process with specific types of optimization actions.

The non-value-adding objects and processes should be the major focal point, since their removal or simplification does not disrupt the theoretical function of the business process. Therefore, these things are potentially waste, so their deletion is preferred over other optimization options, since waste eliminated is preferred over waste reduced. Value-adding objects and value-adding processes should be reviewed as well, but with an opposite intent, because deletion of a value-adding thing undermines the proper functioning of the business process under design. In these cases, simplification activities, such as automation, are preferred over deletion, which is *per se* harmful for a value-adding object or process.

The result of the AD-OPM BPI method is an identified set of solution-neutral process improvements that optimize the system and preserve intended function. The solution-specific means of implementing the improvements should be determined by the expertise and resources available at a company using the method. Therefore, AD-OPM BPI will not provide solution-specific improvements by itself, but instead it will identify solution-neutral means to generate such improvements.

Chapter 7: AD-OPM BPI Case Study

Introduction

The AD-OPM BPI method is demonstrated through a case study, in which the author had access to a large American aerospace manufacturing company, identified with the pseudonym "*Aviator Aerospace*" ("AA"). While specific company nomenclature is disguised in this case study, the AD-OPM BPI method is applied to demonstrate analysis and optimization of both a system of business processes and an individual process within that system. The case study evaluates and validates a combination of concepts, including those that AA has already identified, along with new conclusions and recommendations identified through this AD-OPM BPI analysis.

Background

AA is a manufacturing company that produces aerospace parts and assemblies in accordance with government quality system regulations and aerospace industry standards Consistent with best practices, AA divides its internal control documentation into policies, procedures, and processes that drive its operations (Page 2002). AA has a nine-person work group that focuses just on management of quality assurance processes that quality assurance inspectors and factory mechanics use to perform their work. BPI and process optimization are among the responsibilities of this quality assurance process management group.

The following are the requirements, standards, and internal company policies, procedures, and processes that guide the work performed by AA.

Requirements: Code of Federal Regulations

The Code of Federal Regulations ("CFR") is the codification of the rules and regulations established by the departments and agencies of the United States Federal Government (U.S. National Archives and Records Administration 2015). It is comprised of 50 Titles that represent broad areas that are subject to Federal regulation. The Titles are then subdivided into Chapters that bear the name of the issuing agency and Parts that cover specific regulatory areas.

The primary regulation applicable here is referred to as "*14 CFR 21.137*", which is comprised as follows:

- Title 14 "*Aeronautics and Space*"
- Chapter 1 "*Federal Aviation Administration, Department of Transportation*"
- Subchapter C "*Aircraft*"
- Part 21 "*Certification Procedures for Products and Parts*"
- Subpart G "*Production Certificates*"
- Section 21.137 "*Quality System*".

Companies that either hold or apply for a production certificate must comply with this section. A production certificate is an approval document issued by the Federal Aviation Administration ("FAA") that allows the holder to manufacture aerospace products under an FAA-approved type design. To comply with this section, companies must "*establish and describe in writing a quality system that ensures that each product … conforms to its approved design and is a condition for safe operation*" (Ibid.). The CFR specifically states that this quality system must include procedures for:

a) Design data control,

b) Document control,

c) Supplier control,

c1) Ensuring supplier products confirm to approved designs,

c2) Requiring suppliers to report release of nonconforming products,

d) Manufacturing process control,

e) Inspecting and testing,

e1) A flight test of each aircraft, unless exported as unassembled,

e2) A functional test of each aircraft engine and propeller,

f) Inspection, measuring, and test equipment control,

g) Inspection and test status,

h) Nonconforming product and article control,

h1) Ensuring that only products that conform to approved design are installed on type-certified aircraft,

h2) Ensuring that discarded products are rendered unusable,

i) Corrective and preventative actions,

j) Handling and storage,

k) Control of quality records,

l) Internal audits,

m) In-service feedback,

m1) Addressing any in-service problems involving design changes,

m2) Determining if any changes to Instructions for "Continued Airworthiness" are necessary, and

n) Quality escapes

International Standards: ISO 9001

The International Standards Organization publishes the ISO 9000 Quality Series of standards. The first standard, ISO 9000:2015 "*Quality Management Systems – Fundamentals and Vocabulary*", is a series of quality management system principles as well as the vocabulary that will be used throughout the standard family (International Standards Organization 2015a). The second standard in the series is ISO 9004:2009 "*Managing for the sustained success of an organization – A quality management approach*", expands on those principles (International Standards Organization 2011).

The standard that is more relevant to this thesis is ISO 9001:2015 "*Quality Management Systems – Requirements*". Unlike the other two standards that are treated as supplements, the ISO 9001 requirements are directly audited against by third party assessors to verify that a standardized quality management system is in place (International Standards Organization 2015b). ISO 9001 was recently updated in September 2015 to be less prescriptive, but it also requires top-level organization leaders to be more accountable, and it integrates better with other international standards.

The ISO 9001 standard begins with an introduction (Ibid.). It then addresses several organizational requirements in detail. The main sections of the quality management system requirements are:

1. Scope,
2. Normative References,
3. Terms and Definitions,
4. Context of the Organization,
5. Leadership,
6. Planning,

7. Support,

8. Operation,

9. Performance Evaluation, and

10. Improvement.

For this thesis, AA is assumed to be compliant to ISO 9001:2015. In addition to being a best practice and international standard, ISO 9001:2015 has been incorporated into the international aerospace standard AS9100 (SAE Aerospace 2009). As regulatory bodies, such as the FAA, audit AA to AS9100, compliance to AS9100 therefore demonstrates compliance to ISO 9001:2015.

Aerospace Standards: AS9100

In 1999, the Society of Automotive Engineers ("SAE") began publishing the international aerospace standard AS9100 "*Quality Management Systems – Requirements for Aviation, Space and Defense Organizations*" (Ibid.). The current release, AS9100C, was published in January 2009 and is written to include the requirements of ISO 9001, the general quality management system on which it is based, but it also adds aerospace industry-specific content as well.

The AS9100 standard begins by summarizing its rationale and approach (Ibid.). It then addresses several main sections, with many subsections of specific content. The main sections of the quality management system requirements are:

1. Scope,
2. Normative References,
3. Terms and Definitions,
4. Quality Management System,
5. Management Responsibility,

6. Resource Management,

7. Product Realization, and

8. Measurement, Analysis and Improvement.

For this thesis, AA is assumed to be compliant to AS9100, as it is a best practice within the aerospace industry. Though not listed as a regulation, AS9100 compliance is important, because it makes global aerospace manufacturing, at all levels of the supply chain, consistent in the verifiable application of an aerospace quality management system (Ibid.). As regulatory bodies such as the FAA audit compliance to AS9100, the standard is therefore a requirement for doing business in aerospace.

Quality Manual

The CFR requires companies to create a quality manual. While the scope of this thesis focuses the analysis on the 14 CFR 21.137 requirements of a quality system, 14 CFR 138 "*Quality Manual*" requires each applicant or holder of a production certificate to describe its quality system through a manual and provide that manual to the FAA for approval (U.S. National Archives and Records Administration 2009). Therefore, the quality manual is a company's demonstration artifact of its quality system in a manner that demonstrates that it will produce safe aerospace parts in a manner that can be audited by the FAA.

The format of the quality manual must be "*in a form acceptable to the FAA*", but there is some latitude with the formatting (Ibid.). Companies have the latitude to format quality manuals with any method of numbering, headings, and content provisions. Leading aerospace companies such as United Technologies Corporation choose to format their quality manuals with the numbering, heading, and content that corresponds to the same conventions in the AS9100 aerospace standard (United Technologies Corporation 2015). Many companies take this approach to provide correlation to AS9100, as to directly link quality systems to the aerospace standard to which the FAA audits compliance. As such, AA is assumed to confirm to this formatting to be consistent with leading practices within the industry.

Enterprise Command Media: Policies, Procedures, Processes

Enterprise Command Media is the collective name for policies, procedures, and processes that are defined by industry standards, such as ISO 9000, and for which deployment is integrated throughout the entire ISO 9000 Quality Series of standards (Page 2002, 24-25). The documents contained within Enterprise Command Media act together "*like a state road map*" (Ibid.). These documents represent the sphere of control for most companies; where, unlike regulations and standards that are authored by governments and third party entities, companies themselves control the authoring of the Enterprise Command Media that govern company activities.

"*A policy points out the general direction (objective) to reach a destination or goal*" (Ibid., xv). Other definitions include (i) a document that conveys general strategy or purpose, and is the direction behind procedures and processes (Ibid., xviii), and (ii) "*intentions and direction of an organization*" (International Standards Organization 2015a, 18). An example of a policy from AA is "*Policy-3 Quality*", which acts more of a mission statement than a specific detail direction. In the requirements for the policy, it is stated that "*Quality is instilled into every aspect of the business*" and that the company's "*Quality Management System defines requirements and enables process improvement to drive performance and customer satisfaction*" (Aviator Aerospace 2014, 1-2). Therefore, in this thesis, policies will be considered the "*strategy*" documents of Enterprise Command Media.

If a policy provides the general direction, then procedures provide the highways (requirements) to accomplish the objectives and goals. "*The procedure lays out the steps usually followed when performing repeatable types of work*" (Page 2002, xv). Here, the more formal definitions for procedure include (i) "*a plan of action for achieving a policy; it is a method by which a policy can be accomplished and it provides the instructions needed to carry out a policy*" (Ibid., xviii), and (ii) a specified way to carry out an activity or process (International Standards Organization 2015a, 18). An example of a procedure from AA is Procedure-8 "*Configuration Management Objectives*". This procedure lists Policy-3 "*Quality*" as its requirement, and expands on the general direction of the policy by providing the objectives of a configuration management system (Aviator Aerospace 2015, 1-2). In this thesis, procedures are considered the "*plan*" documents of Enterprise Command Media.

Continuing the driving-direction analogy, a process is similar to the turn-by-turn directions that occur after exiting the highway (procedure), which encompasses certain side-streets to finally reaching the destination. More formally, a process can be defined as (i) "*a sequence of steps performed for a given purpose, for instance, the software development process*" (Page 2002, xviii), and a "*set of interrelated or interacting activities that use inputs to deliver an intended result*" (International Standards Organization 2015a, 15). A process is always behind every policy or procedure (Page 2002, xviii). An example from AA aerospace is company Process-10 "*Uninstall Part or Assembly*", which references Procedure-8 "*Configuration Management Objectives*" as its requirement, but provides more detail by describing the exact method of performing a part uninstallation and reinstallation process so that the requirements of the procedure are satisfied (Aviator Aerospace 2016).

Processes are essentially where all of these requirements, standards, strategies (policies), and plans (procedures) intersect to enable the work that is to be performed. While it is important for process improvement personnel and process users to understand the meaning behind the work that is being performed, the processes embody the work that is actually occurring. During audits, for example, the Enterprise Command Media is reviewed to determine that it clearly meets the requirements and standards imposed on the system, but also process users are evaluated to determine that they are correctly using these processes that embody the procedures, policies, and ultimately the requirements and standards that govern the system (Page 2000). Therefore, in this thesis, processes are considered the "action" documents of Enterprise Command Media.

System and Individual Process to be Optimized

This case study applies the AD-OPM BPI method to optimizing the system of Enterprise Command Media (that comprises over several hundred policies, procedures, and processes) as well as the specific functionality for an individual process. Process-10 "*Uninstall Part or Assembly Process*" (abbreviated, the "*Uninstall Process*") was selected due to feedback from AA for it being one of the more difficult processes to improve through BPI efforts. According to working team time trials, the Uninstall Process takes on average approximately 84 minutes to perform. The process involves 14 written steps, featuring frequent exchanges between Factory Mechanics and Quality Assurance Inspectors at different intervals. The purpose of the process is to maintain an auditable record the uninstallation and reinstallation of previously inspected parts or assemblies to demonstrate that engineering requirements are returned to a satisfactory state.

The 14 steps in the Uninstall Process are as follows:

1. Either the Factory Mechanic or the Quality Assurance Inspector initiates both the Uninstall Record and Uninstall Order to begin the process.
2. The Factory Mechanic makes a request to the Quality Assurance Inspector for authorization to uninstall the part or assembly.
3. The Quality Assurance Inspector authorizes the Factory Mechanic's request for authorization.

4. The Factory Mechanic uninstalls the part or assembly.
5. The Factory Mechanic makes a request to the Quality Assurance Inspector for authorization to reinstall the part or assembly.
6. The Quality Assurance Inspector authorizes the Factory Mechanic's request for authorization.
7. The Factory Mechanic reinstalls the part or assembly.
8. The Quality Assurance Inspector verifies that the part or assembly reinstallation was performed correctly.
9. The Factory Mechanic and the Quality Assurance Inspector determine if a retest of the reinstalled part or assembly is necessary.
10. If necessary, the Factory Mechanic and the Quality Assurance Inspector retest the reinstalled party or assembly.
11. If necessary, the Quality Assurance Inspector verifies that the retest was performed correctly.
12. The Federal Aviation Administration (FAA) Coordinator inspects the reinstalled part or assembly.
13. The Quality Assurance Inspector completes the Order.
14. The Factory Mechanic completes the Record to end the process.

Wait times between steps are noted as one role triggers another role to queue up to begin their next step.

Applying the AD-OPM BPI Method

Having described the details of the requirements, standards, Enterprise Command Media, and the specific Uninstall Process, the AD-OPM BPI method is now applied.

OPM Model Context of the System to Be Optimized

An optional, but helpful step to begin with it to develop a simple model of the system operation to keep context of the system's operation to guide the users understanding throughout the remaining steps. This activity also demonstrates an example of the benefit of using MBSE in general, and OPM in particular. While the previous descriptions of the requirements and standards that, as a system, guide AA's manufacturing activities, modeling enables synthesis and understanding of the architecture with much more context.

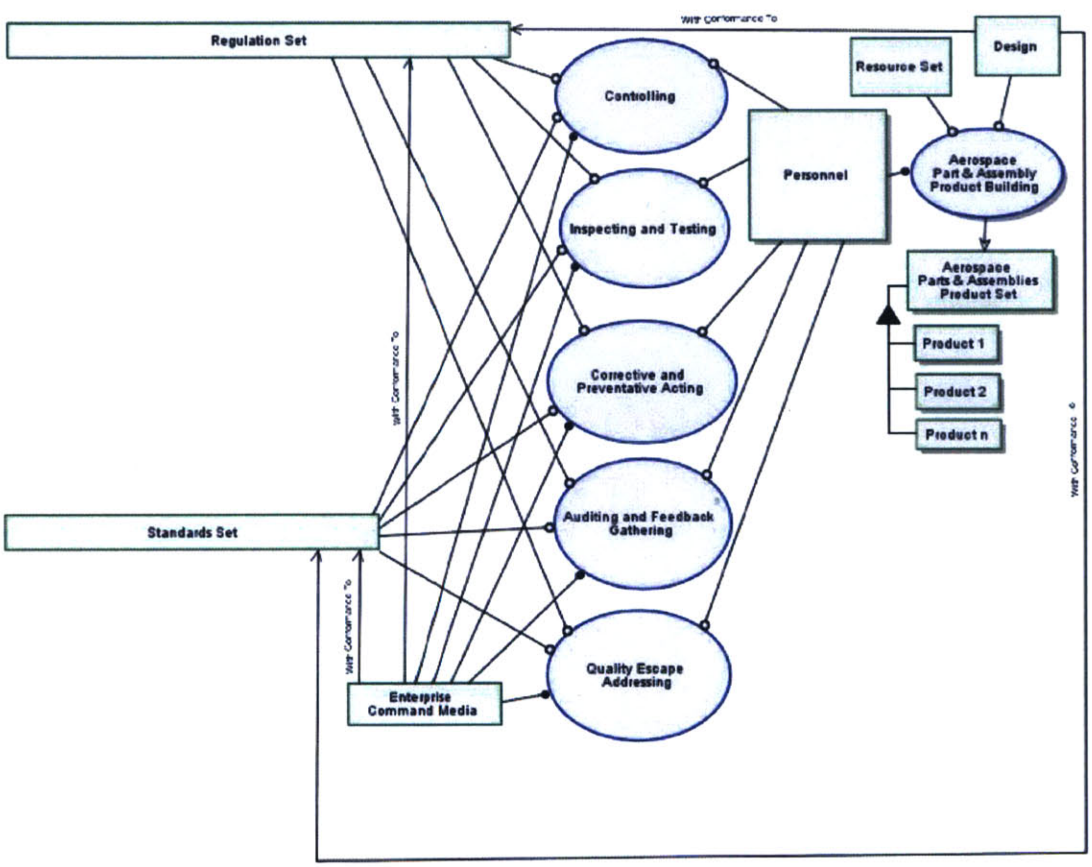

Figure 7.1: Simple OPM of Aerospace Product Manufacturing System

The model in Figure 7.1 displays how the Enterprise Command Media that guides the personnel to conduct their work is influenced by regulations and standards. While this seems simple, the wide breadth of the regulations and standards, as applied to an even vaster array of policies, procedures, and processes within Enterprise Command Media create a complex network that must be managed to fulfill the goals of the system. Figure 7.2 displays how decomposing one additional layer further begins to saturate the design model, even without decomposing the hundreds of Enterprise Command Media documents or other objects. This demonstrates the importance of using models to fulfill the goals of the system.

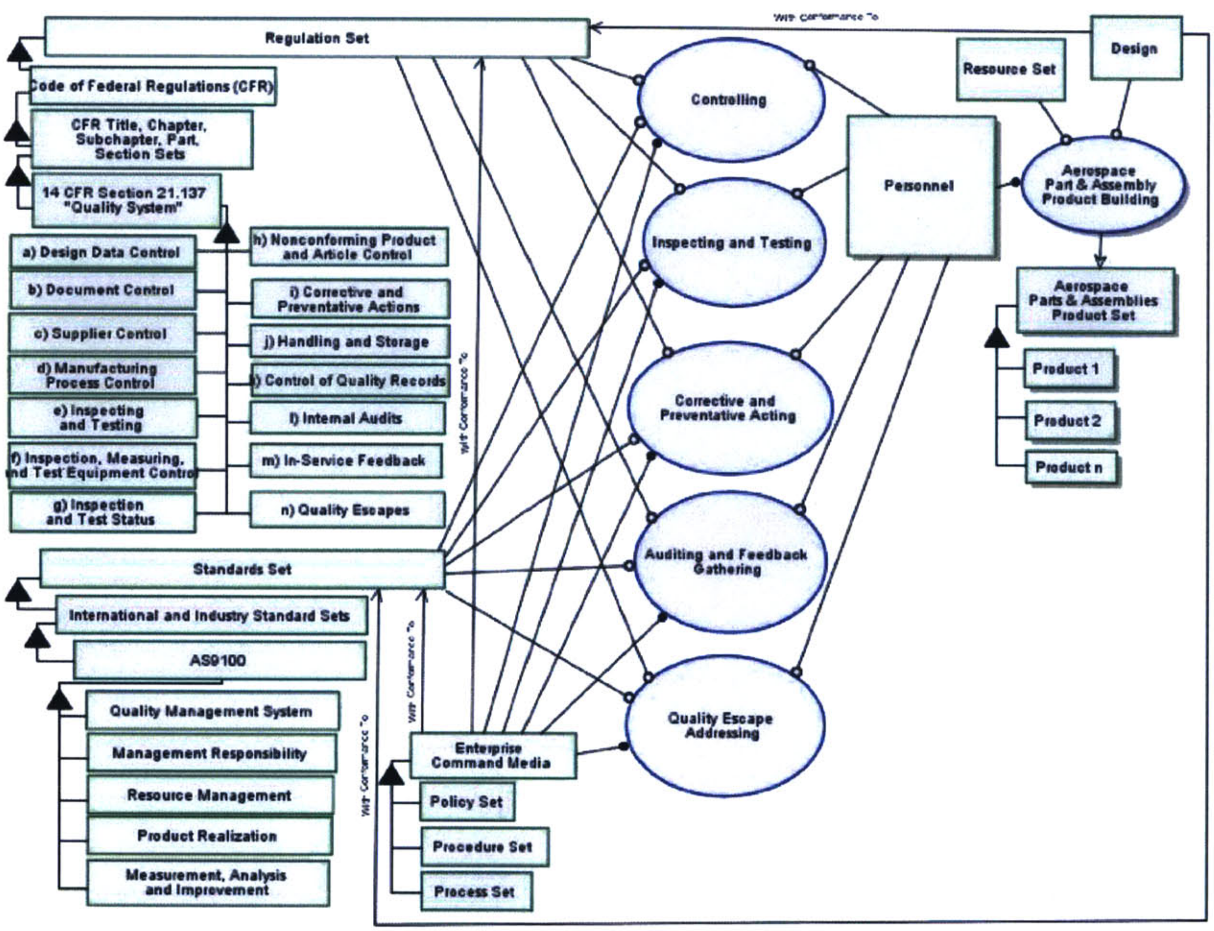

Figure 7.2: OPM with Decomposed Regulations and Standards

System Analysis and Optimization with Axiomatic Design

With the relevant aspects that shape the system now defined, AD will now be used to decompose the current state of the system so that recommend changes can be proposed based on objectively criteria. The focus is determining if an efficient system design currently exists, and to the extent that it can be better, recommending a re-engineered and re-architected solution to improve it. Here, AA's Quality System is evaluated, starting from Tier-1 Federal Requirements and decomposing through limited Tier-5 business processes. Tier-5 is presented as limited-scope due to the scope of the entire system being too large to practically illustrate.

Step 1 – Define the FRs of the System

The process of defining the FRs of this existing system comes from decomposing the system itself, in contrast to starting over with a new design. This begins with decomposing the system from the originating Tier-1 requirements and tracing the current state fulfillment of the requirements through the existing DPs.

To decompose the system, process trees and compliance matrices within the AA Quality Manual and other compliance mapping computer systems were examined. As previously mentioned, these sources provide links between requirements, standards, and Enterprise Command Media that are sufficient to demonstrate compliance to regulations and standards, but lack information regarding a conscious design of the system that could more clearly articulate this and be more beneficial to BPI efforts. For example, these resources clearly demonstrate how the requirements of a specific government CFR regulation links to one or more standards, policies, and procedures. Although these are linked, a key missing system architectural and business decision component is the absence of a holistic view exists to evaluate all of the design decisions that embody those requirements through design. AD mapping will accomplish this.

Top-Down Decomposition: Tier-1 Requirements

As described earlier, the requirements for AA and its aerospace product and assembly manufacturing system originate with the Code of Federal Regulations that is issued by the United States Government, specifically, 14 CFR 21.137 subsections (a) through (n). As provided in Figure 7.3, these requirements have been arranged into a design matrix where the left column displays the FRs addressed by the regulations and the top row contains the specific DP regulation sub-sections that embody the FRs.

Tier-1 Regulations	CFR Regulation	14 CFR 21.127(a)	14 CFR 21.127(b)	14 CFR 21.127(c1)	14 CFR 21.127(c2)	14 CFR 21.127(d)	14 CFR 21.127(e)	14 CFR 21.127(e1)	14 CFR 21.127(e2)	14 CFR 21.127(f)	14 CFR 21.127(g)	14 CFR 21.127(h)	14 CFR 21.127(h1)	14 CFR 21.127(h2)	14 CFR 21.127(i)	14 CFR 21.127(j)	14 CFR 21.127(k)	14 CFR 21.127(l)	14 CFR 21.127(m)	14 CFR 21.127(m1)	14 CFR 21.127(m2)	14 CFR 21.127(n)
Government Requirement		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Design Data Control	1	1																				
Document Control	2		2																			
Supplier Products Confirm to Designs	3			3																		
Suppliers Report Release of NC Prod.	4				4																	
Mfg Process Control	5					5																
Inspecting and Testing	6						6															
Flight Test of Each Aircraft	7							7														
Functional Test of Ea. Engine and Propeller	8								8													
Inspect/Measur/Test Equipment Control	9									9												
Inspection and Test Status	10										10											
Nonconforming Product & Article Control	11											11										
Product Conform to Approved Design	12												12									
Render Disgarded Products Unusable	13													13								
Corrective and Preventative Actions	14														14							
Handling and Storage	15															15						
Control of Quality Records	16																16					
Internal Audits	17																	17				
In-Service Feedback	18																		18			
In-Service Problemsw. Design Changes	19																			19		
Continued Airworthiness Determination	20																				20	
Quality Escapes	21																					21

Figure 7.3: Tier-1 AD Matrix for CFR Regulations

The result is a diagonal uncoupled matrix. This is rational given that the FRs and DPs are both defined and satisfied in the same regulation. While the design of the regulation is considered "good" by the Axiomatic Design principles (since the FR-DP mapping results in a uncoupled design), the evaluation being performed will now be moved into lower DP tiers (i.e. generally the Enterprise Command Media, and specifically the company processes) to evaluate efficiency of fulfilling the system requirements.

Top-Down Decomposition: Tier-2 Standards

Beyond the regulatory requirements, AA is subject to international and industry standards to operate is aerospace manufacturing system. Effectively the applicable ISO industry standards are considered merged into the AS9100 aerospace standard. Therefore, the standards of AS9100 have been added as Tier-2 FRs and DPs in Figure 7.4.

Tier-2 External Regulations and Standards		CFR Regulation	14 CFR 21.127(a)	14 CFR 21.127(b)	14 CFR 21.127(c1)	14 CFR 21.127(c2)		14 CFR 21.127(d)	14 CFR 21.127(e)	14 CFR 21.127(e1)		14 CFR 21.127(e2)	14 CFR 21.127(f)	14 CFR 21.127(g)	14 CFR 21.127(h)	14 CFR 21.127(h1)	14 CFR 21.127(h2)	14 CFR 21.127(i)		14 CFR 21.127(j)	14 CFR 21.127(k)	14 CFR 21.127(l)	14 CFR 21.127(m)	14 CFR 21.127(m1)	14 CFR 21.127(m2)	14 CFR 21.127(n)
		AS9100	AS9100C 7.3	AS9100C 4.2.3	AC9100C 7.4.3	AS9100C 8.3	AS9100C 7.4.2	AS9100C 7.5.1	AS9100C 8.2.4	AS9100C 7.3.5	AS9100C 7.3.6		AS9100C 7.6	AS9100C 7.5.3	AS9100C 8.3			AS9100C 8.5.2	AS9100C 8.5.3	AS9100C 7.5.5	AS9100C 4.2.4	AS9100C 8.2.2	AS9100C 7.5.1.4			AS9100C 8.3
Government Requirement	Industry Standard		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Design Data Control	Inspection and Traceability	1	1																							
Document Control	Control of Documents	2		2																						
Supplier Products Confirm to Designs	Verification of Purchased Product	3			3																					
Suppliers Report Release of NC Prod.	Control of Nonconforming Product	4				4																				
	Purchasing Information	5					5																			
Mfg Process Control	Control of Production & Service	6						6																		
Inspecting and Testing	Monitoring and Measurement of Product	7							7																	
Flight Test of Each Aircraft	Design and Development Verification	8								8																
	Design and Development Validation	9									9	1														
Functional Test of Ea. Engine and Propeller		10										10														
Inspect/Measuring/Test Equipment Control	Control of Monitoring and Measuring Equipment	11											11													
Inspection and Test Status	Identification and Traceability	12												12												
Nonconforming Product & Article Control	Control of Nonconforming Product	13													13	1	1									
Product Conforms to Approved Design		14													1	14	1									
Render Disgarded Products Unusable		15													1	1	15									
Corrective and Preventative Actions	Corrective Action	16																16								
	Preventive Action	17																	17							
Handling and Storage	Preservation of Product	18																		18						
Control of Quality Records	Control of Records	19																			19					
Internal Audits	Internal Audit	20																				20				
In-Service Feedback	Post-Delivery Support	21																					21	1	1	
In-Service Problems, Design Changes		22																					1	22	1	
Continued Airworthiness Determination		23																					1	1	23	
Quality Escapes	Control of Nonconforming Product	24																								24

Figure 7.4: Tier-2 AD Matrix for CFR Regulations and AS9100 Standards

The blue highlighted cells represent single requirements that are fulfilled by multiple standards. The yellow highlighted cells represents single standards that fulfill multiple requirements. These colors are then shaded into the design matrix to demonstrate the effect of misalignment as coupled-groups, and binary inputs where the lowest-tier appears multiple times. The lone green shaded binary cell represents a partial overlap of a requirement grouping and the standard grouping.

An observation is that the requirements are consistent with the standards, but are not perfectly aligned. This misalignment is apparent on the design matrix, where the headers for the rows and columns are out of sync. Though the coupling results in several shaded or binary groupings, the groupings are compact, which is preferred over other looser or larger groupings that result in a more iterative design. Another observation is the misalignment that can occur when decomposing subsequent tiers, especially when the misaligned system designs were created by different entities (government vs. industry), with different audiences (domestic vs. international/industry), and under the assumption that AD design principles were not used for creating either the regulations or standards.

Top-Down Decomposition: Tiers 3 & 4 Policies & Procedures

The next level of decomposition is to the Tier-3 Policies that are part of AA's Enterprise Command Media. These are the strategies that AA pursues, with alignment to the regulations and standards that it is subject to. At this Tier-3, the AA Policies had such a high degree of connectivity to the regulations and standards, that the highly-iterative structure was almost un-mappable through AD. This result is understandable given that the AA Enterprise Command Media system was created without using AD mapping, and therefore may not be readily designed for it.

Though AD mapping, as previously described, could not be accomplished, an alternate method of demonstrating AD-like concepts is used to visualize the iterative relationships between the regulations, standards, policies, and procedures. In Figure 7.5, the highly iterative policy layer, as well as a primary layer of procedures were removed from the design matrix and placed to the side as a Domain Mapping Matrix ("DMM"), which transitions the AD matrix into a Multidomain Architecture Model ("MDM") (Eppinger and Browning 2012, 233-244).

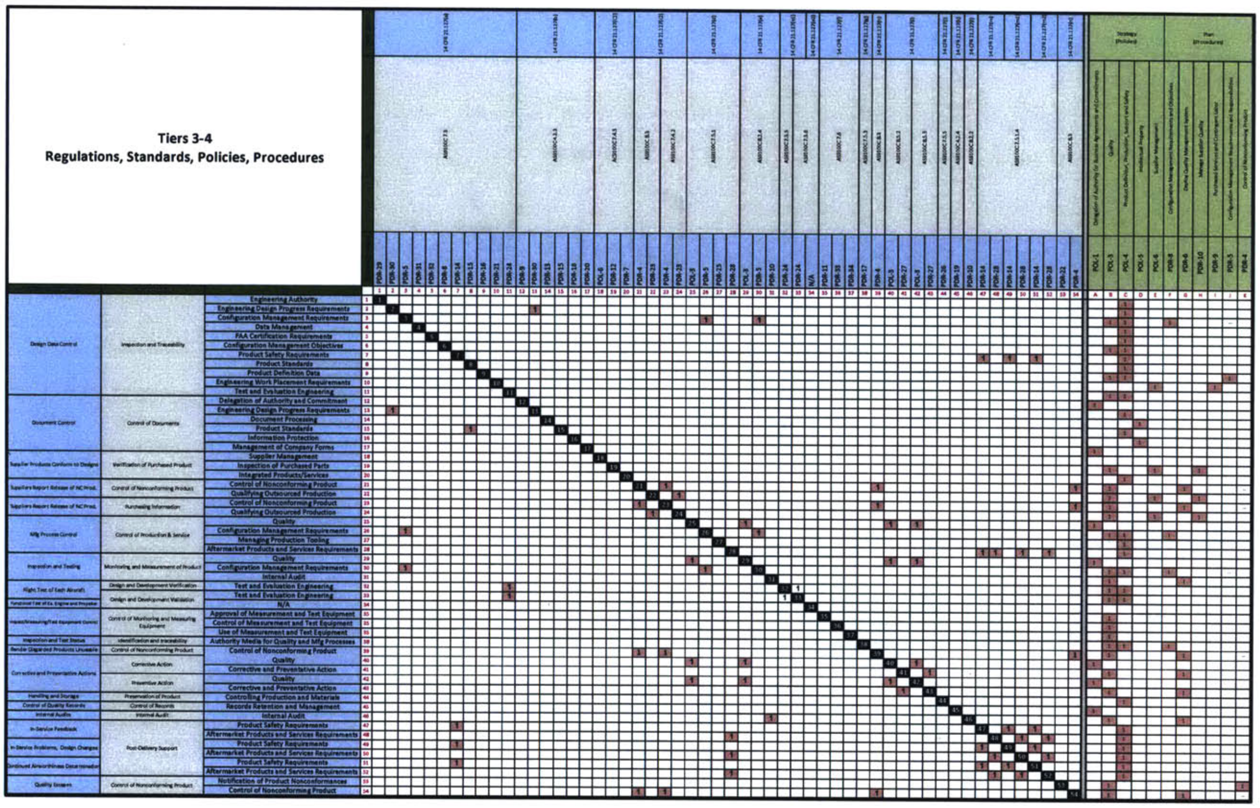

Figure 7.5: Tier-3 and Tier-4 MDM with Regulations, Standards, Policies and Procedures

As the scale of Figure 7.5 makes individual data values almost unreadable, the red cells demonstrate the degree of procedure iteration on both the lower and upper triangles (resulting in a coupled design), as well as the large number of entries in the MDM columns that demonstrate additional layers of highly iterative policies and procedures. This shows that certain procedures fulfilling the requirements of multiple regulations and industry standards.

By focusing on the DMM with the green headers, one can observe that certain policies span a significant portion of the entire matrix. For example, POL-4 "*Product Definition, Production, Support and Safety*" is a policy reference for 27 of the 54 procedures. Furthermore, some procedures note up to three references, including multiple policies or an additional layer of procedures. This is better-displayed in Figure 7.6, which limits the scope of the design matrix to the 14 CFR 21.137(a) "*Design Data Control*" regulation. Here, 10 of 11 procedures support POL-4, but 5 procedures have more than one other policies and/or procedures that are supported, and 3 of 11 procedures have 3 policies and/or procedures that are supported. The result is a structure where compliance can be demonstrated through process trees and compliance matrices, but lacks objective evidence of a clean and purposeful design.

Tiers 3-4 (Limited Scope)
Regulations, Standards, Policies, Procedures

14 CFR 21.127 — 14 CFR 21.127(a)
AS9100 — AS9100C 7.3
Enterprise
Strategy (Policies): Quality; Product Definition, Production, Support and Safety; Supplier Management
Plan (Procedures): Configuration Management Requirements and Objectives; Purchased Services and Contingent Labor; Configuration Management Requirements and Responsibilities

Government Requirement	Industry Standard	Plan		PDR-29	PDR-30	PDR-5	PDR-31	PDR-32	PDR-8	PDR-14	PDR-15	PDR-16	PDR-21	PDR-24	POL-3	POL-4	POL-6	PDR-8	PDR-9	PDR-5
				1	2	3	4	5	6	7	8	9	10	11	B	C	E	F	I	J
Design Data Control	Inspection and Traceability	Engineering Authority	1	1												1				
		Engineering Design Progress Requirements	2		2											1				
		Configuration Management Requirements	3			3									1	1		1		--
		Data Management	4				4									1				
		FAA Certification Requirements	5					5								1				
		Configuration Management Objectives	6						6						1	1				
		Product Safety Requirements	7							7						1				
		Product Standards	8								8					1				
		Product Definition Data	9									9			1	1				1
		Engineering Work Placement Requirements	10										10				1		1	
		Test and Evaluation Engineering	11											11	1	1				

Figure 7.6: Tier-3 and Tier-4 (Limited Scope) MDM with Regulations, Standards, Policies and Procedures

Bottom-Up Decomposition: Tiers 3, 4, and 5 Enterprise Command Media

With the top-down AD-mapping being hindered by a highly-iterative and un-mappable design, the approach shifts to a bottoms up decomposition to demonstrate the AD-mapping method. Figure 7.7 demonstrates three tiers of AD-mapping of a class of Enterprise Command Media to fulfill the function of Customer Quality Support.

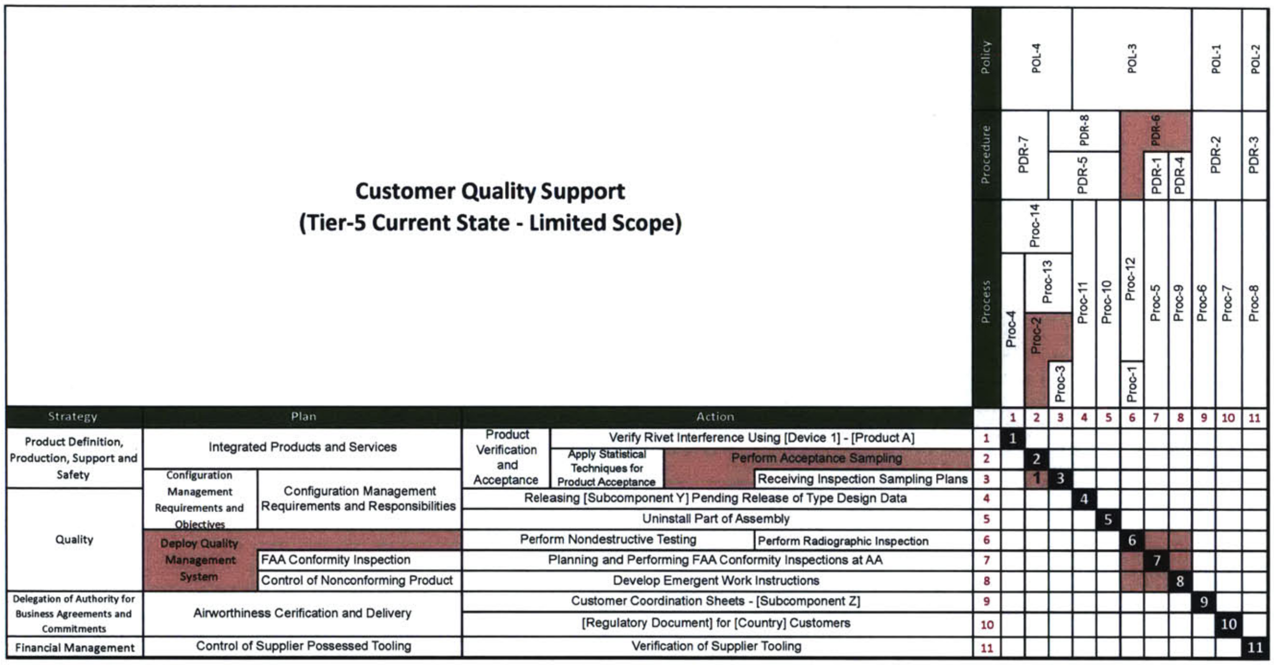

Figure 7.7: Bottom-Up Decomposition: Tiers 3, 4, and 5 Enterprise Command Media

This view is more compatible with AD-mapping of processes within a manufacturing environment. It can be read inside-out, where it can trace specific processes at the center to the over-archiving policies (strategies) that are at least partially-fulfilled by the specific process. This is similar to a process tree or a more visual version of a compliance matrix.

More importantly and in the alternative, this can also be read outside-in. For a given the function, such as Customer Quality Support, it demonstrates the strategies (policies) on the outside that enable that function. AD-mapping then demonstrates the procedures (plans) that enable the policies, as well as the processes (actions) that further those procedures.

While the limited scope in Figure 7.7 is more consistent with AD than the highly iterative views from Figures 7.5 and 7.6, the current state limited scope design still contains inconsistencies:

1. Sequential Process Fulfillment – The first inconsistency is sequential process fulfillment. For example, the process tier shows that there are up to four sequential processes involved to perform a specific action, or fulfill a specific procedure. Proc-3 "*Receiving Inspection Sampling Plans*" is shown to fulfill the earlier design decisions of Proc-2 "*Perform Acceptance Sampling*", which itself fulfills Proc-13 "*Apply Statistical Techniques for Product Acceptance*", and which also fulfills Proc-14 "*Product Verification and Acceptance*". This creates a design much more difficult to map than one where there is a uniform number of layers.
2. Misalignment - The second inconsistency is misalignment. For example, Proc-14 "*Product Verification and Acceptance*" fulfills the procedures of PDR-7 "*Integrated Products and Services*" and PDR-5 "*Configuration Management Requirements and Responsibilities*". While a multi-to-one ratio is common with AD-mapping, it occurs where the design splits from one higher-tier element into one or more lower-tier elements that enable it in more fidelity. Here, multiple higher-tier elements (PDR-7 and PDR-5) are satisfied by a single lower-tier requirement (Proc-14).
3. Dual Roles: The third inconsistency is that some leaves serve multiple roles as both leaves and trees, even within the same tier. These are indicated in cells shaded red. Proc-2 "*Perform Acceptance Sampling*" exists both as an independent process that performs an individual function, but also as a requirement to performing Proc-3 "*Receiving Inspection Sampling Plans.*" At the

procedure level, this occurs with PDR-6 "*Develop Quality Management System*" which is another procedure itself, as well as with a requirement to other procedures PDR-1 "*FAA Conformity Inspection*" and PDR-4 "*Control of Nonconforming Product*". This essentially takes plans that have additional layers of plans as requirements instead of a single articulable plan being designed.

4. Limited Context: The fourth inconsistency is that highly iterative elements that extend beyond the limited scope of view provide an incomplete understanding of the DPs and FRs that are to be addressed. As noted in Figure 7.5, AA's policies were so highly iterative across regulation and standards categories, that using a limited context view does not demonstrate all of the requirements and decisions necessary to fulfill those policies are not taken into account. This defeats the purpose of using AD.

5. Unclear Functional Link: The fifth and final inconsistency is that AA's compliance media does not perform well to demonstrating the link between business functions such as Customer Quality Support and regulations/standards that are presumably furthered by such functions. The AA Quality Manual and other systems reviewed lack description how this function relates to the requirements and standards placed upon the system. Instead, compliance matrices merely describe the contents Enterprise Command Media that related to functions like Customer Quality Support, without demonstrating how such contents fulfill the design of the system.

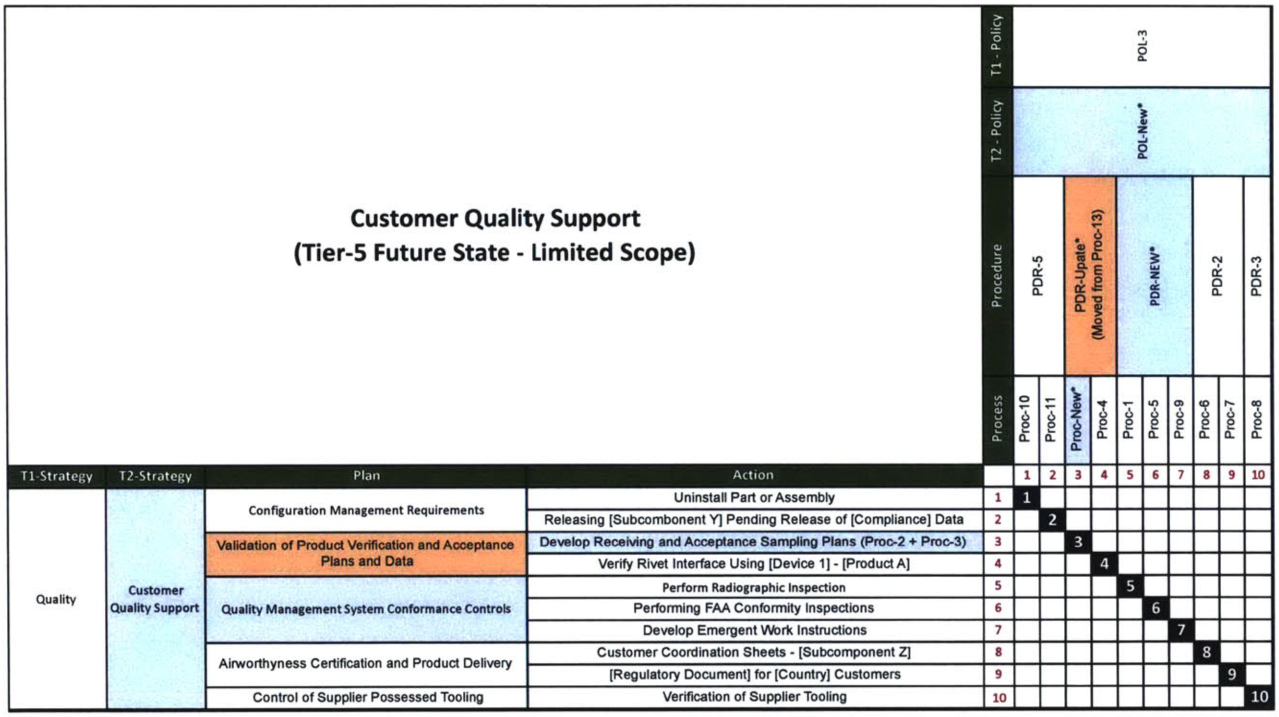

Figure 7.8: Bottom-Up Decomposition: Redesign Example

To address the criticisms that were contained in the bottoms-up decomposition of the Customer Quality Support function in Figure 7.7, Figure 7.8 demonstrates how the principles of AD-mapping can be used to create a clean, efficient, and demonstrable design of business processes. The following are remedies to inconsistencies observed in earlier Figures.

1. Design Consolidation - By reviewing the processes and relationships on Figure 7.7, the assumption was made that elements could be re-arranged or consolidated to create efficiency. For example Proc-2 "*Perform Acceptance Sampling*" and Proc-3 "*Receiving Inspection Sampling Plans*" were consolidated into "*Proc-New**" that is shaded blue in Figure 7.8. In addition, PDR-5 "*Configuration Management Requirements and Responsibilities*" and PDR-8 "*Configuration Management Requirements and Objectives*" were consolidated

into a single PDR-5 "Configuration Management Requirements" to eliminate redundancy. Other new items are demonstrated in blue, with a new procedure shaded orange to consolidate lower-level processes into a higher-tier procedure.

2. Diagonal Matrix – The design features a diagonal matrix without upper or lower triangle values, which indicates that Figure 7.8 is a desirable uncoupled "good" design without iterations.

3. Non-Layered Functionality – As opposed to Figure 7.7, where multiple sequential layers of processes were used to fulfill others layers that included procedures and other processes, the design here does not contain that sequential structure. Instead, the sequential elements were either consolidated as described earlier.

4. Alignment – Unlike the misalignment in the policies and procedures that was apparent in Figure 7.7, Figure 7.8 has remedied that inefficiency with very organized alignment that clearly demonstrates which actions fulfill which plans, which fulfill which strategies. Top accomplish this, the content and structure of the elements were re-examined to determine how the misalignment could be re-aligned through modifying or consolidating other elements.

5. Single Roles – Another difference from Figure 7.7 is that Figure 7.8 demonstrates single roles. No proposed element serves as both a leaf and a branch, meaning that lowest-tier FR-DP pairings are no longer constraints on other lowest-tier FR-DP pairings.

6. Broad Context – While Figure 7.7 lacked context of how the policies were linked into the rest of the system, Figure 7.8 partially remedies that with the inclusion of sub-policies. A sub-policy here is a Tier-2 Policy (i.e. strategy) that provides additional context to a boarder higher-tier Policy. For example, POL-3 "*Quality*"

still presumably spans many functions, regulations, and standards, but the newly created Tier-2 POL-New* "*Customer Quality Support*" does not. Therefore, when the procedures or processes plans are designed as DPs from these sub-policies, then the entire context of the sub-policy is shown for decision making and BPI efforts. The added benefit is that the function of Customer Quality Support, which was previously ignored as a discrete function with its own FRs and DPs, can now be efficiently structured through Enterprise Command Media as well.

Summary and Conclusions of System Analysis and Optimization with Axiomatic Design

AD-mapping as applied to a complex system of regulations, standards, and Enterprise Command Media has varied results. It worked best when starting with a new design and adhering that design's structure to AD design principles to give a clean, efficient, and conscious demonstration of how the DPs of that design fulfill the FRs. The decomposition of an existing design, especially one that was not designed with AD in mind, was challenging and became increasingly difficult or impossible as the size of the system grew. Therefore, the conclusion is that AD would be ideal to use for designing new systems or re-aligning existing small systems of Enterprise Command Media. To the extent that a system of Enterprise Command Media is not small enough to efficiently decompose the architecture with AD-mapping, then other tools to deal with such complexity (such as MDM) can be used to bring context to the system before a large-scale redesign of the system with AD-mapping can be conducted. Otherwise, additional inefficiencies may hinder efforts to simply re-align a large system of Enterprise Command Media to demonstrate a conscious intent of the system.

Process Analysis and Optimization with Object-Process Methodology

Assuming that AA's process system for manufacturing activities will be analyzed and optimized with AD, each individual process within the system can be analyzed and optimized with OPM. This transitions the focus from engineering the larger process system of multiple processes, to engineering a single process to efficiently fulfill requirements, standards, and other business decisions with similar efficiency. Here, AA's Uninstall Process is used as the example due to its complexity and mixture of company-proposed and analysis-produced opportunities for improvement.

Step 1: Process Decomposition

The decomposition of the entities (things, i.e., objects and processes) of the Uninstall Process is straightforward, because the 14 process steps were already identified, along with inputs and outputs for each step, the performers of each step, and the systems used by the performers. Such information is considered best practice to include in business process documentation (Page 2002). In addition to objects and processes being identified in business process documentation, an object's beginning and end states are identified if they are changed through performing the process.

Each of the 14 steps listed in the Uninstall Process document were converted into separate processes in the OPM model. To help relate the written process steps, each process in the OPM model was numbered with the respective process step number. The workers and systems that perform or affect the process, specifically the (i) **Quality Assurance Inspector**, (ii) **Factory Mechanic**, (iii) **FAA Coordinator**, (iv) **Manufacturing Data System**, and (v) the **Requirements Data System**, are represented in the OPM model as objects. Lastly, state changes were modeled with careful attention to the orders and records being opened and closed. For example, in Figure 7.9, **installed** is both the initial and final state of **Part**. Process

Step 4, **Uninstall Performing**, changes **Part** from **installed** to **uninstalled**, and Process Step 7, **Uninstall Performing**, does the opposite.

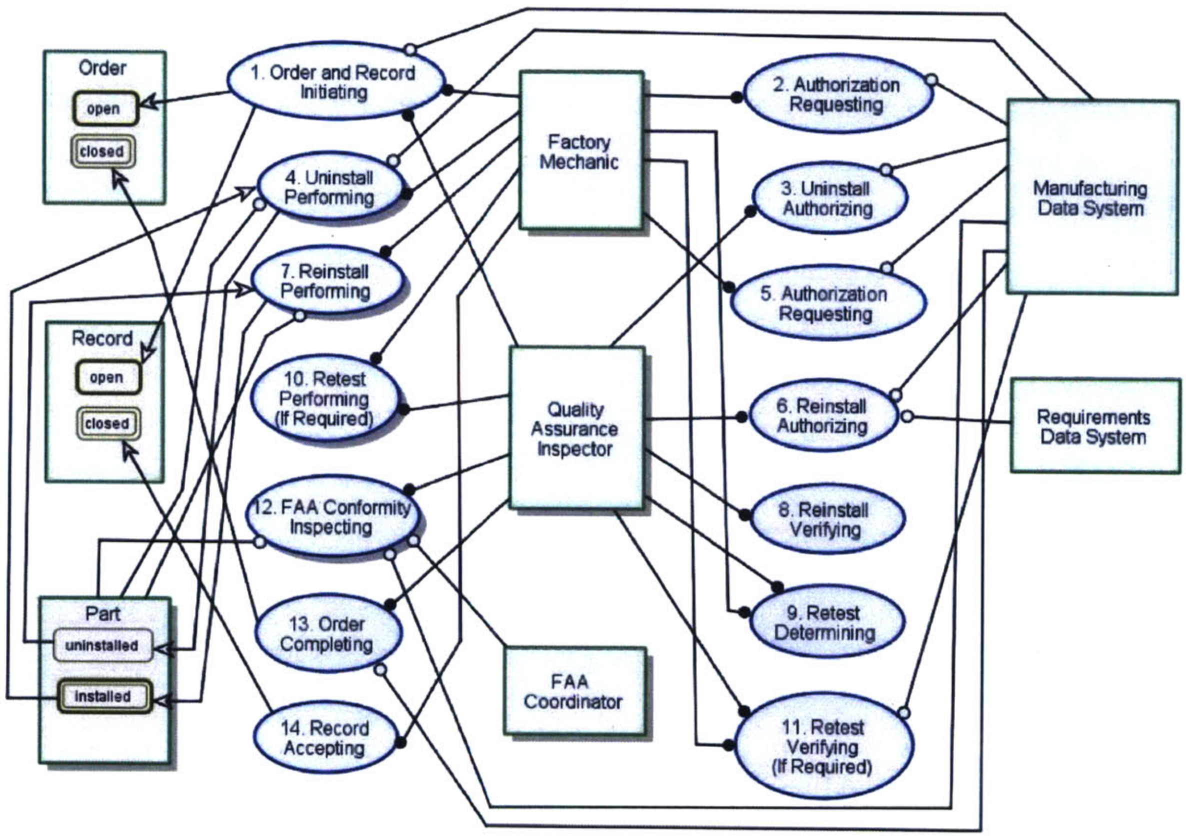

Figure 7.9: OPM Layered Architecture of Uninstall Process

Step 2: Process Rationalization

After the decomposition, the next step is to rationalize those entities into layered architecture. Figure 7.9 demonstrates the primary value-creation function of the process by selecting and grouping respective entities into operands—essential objects that the system transforms, thereby adding value (on the left), internal value-related processes to the left of the operands, value-related instrument objects next, then supporting processes, and finally auxiliary objects. The classification of these things was based on reviewing the process documentation to verify that the intended output is the reinstallation of an uninstalled part or assembly. Stated differently, that the state of the part or assembly changes from **installed** to **uninstalled**, and back to **installed**.

The selection of the internal value-related processes was more subjective from the point of view of the process improvement architect. The concept was to identify which of the process steps, if removed, would undermine the intended-function of the documented process. The ones selected as value-adding processes were: 1) **Order and Record Initiating**, 4) **Uninstall Performing**, 7) **Reinstall Performing**, 10) **Retest Performing (If Required)**, 12) **FAA Conformity Inspecting**, 13) **Order Completing**, and 14) **Record Accepting**.

By description, steps 10) **Retest Performing (If Required)** and 12) **FAA Conformity Inspection** may seem to be non-value-adding processes, since they are by definition either rework or verification. Still, value-adding processes were listed since this AA views them as a quality assurance processes, which are considered value-adding, though from a manufacturing perspective such activities could be considered non-value-adding. The decision not to list steps 10 **Retest Performing (If Required)** and 12 **FAA Conformity Inspecting** as supporting processes, unlike the other verification-type processes, is that they are imposed by the auditing entities, and therefore are value-adding to the extent that they satisfy mandatory external constraints.

The value-related instrument and agent objects (middle column) were then listed as those that either perform or are essential to function. This was straightforward from the process documentation that lists the performers as the OPM agents (humans) **Factory Mechanic**, **Quality Assurance Inspector**, and **FAA Coordinator**. The less straightforward one is the **Record**, which may seem non-essential, but is also a requirement-constraint imposed by external sources.

Supporting processes and supporting objects/interfaces are those entities that, if they were simply deleted, would not disrupt the intended function of the documented process. For processes, these are all the remaining *requesting, authorizing, verifying,* and *determining* steps.

These are differentiated from internal value-related processes because these steps are not imposed requirements that must be satisfied. The supporting objects/interfaces similarly assist the process, but would undermine the intended function if they were deleted.

Step 3: Process Optimization

For optimization to take place, non-value-added columns were evaluated to determine if the entities, individually or collectively, could be reduced or deleted. This means that the manufacturing and requirements systems, as well as the *requesting, authorizing, verifying,* and *determining* steps, were targeted for (i) deletion, (ii) combination, (iii) reduction or simplification, (iv) automation, (v) offload or outsource, and/or (vi) upgrade. Figure 7.10 demonstrates the three objects and five processes targeted to be optimized as pink (darker than the rest).

Optimization 1: Current AA Proposal – Delete Inspection by Quality Assurance Inspector (Supporting Processes): The improvement of Operator Self-Inspection ("OSI") has been considered by AA leadership. The concept of OSI has existed for a few decades, but still has not been fully deployed into some manufacturing companies like AA (Whittingham 1986). OSI shifts responsibility of quality inspection from the **Quality Assurance Inspector** to the operator of the process, which in the case study is the AA **Factory Mechanic**. **Quality Assurance Inspectors** then perform a separate external function of monitoring the certification of the self-inspecting operators. One of the goals of the OSI is to eliminate the need for the operator to stop and wait for an inspector to come and inspect the product. Here, the incorporation of OSI would result in deletion of non-value-adding Step 8 **Reinstall Verifying**, where the **Quality Assurance Inspector** would otherwise inspect the reinstalled **Part** or assembly, and instead merge that inspection back into Step 7 **Reinstall Performing** for the **Factory Mechanic** to perform during the reinstallation. This could be alternatively viewed as (i) deletion, (ii) combination, or (iii) reduction or simplification, depending on the perspective of the process architect.

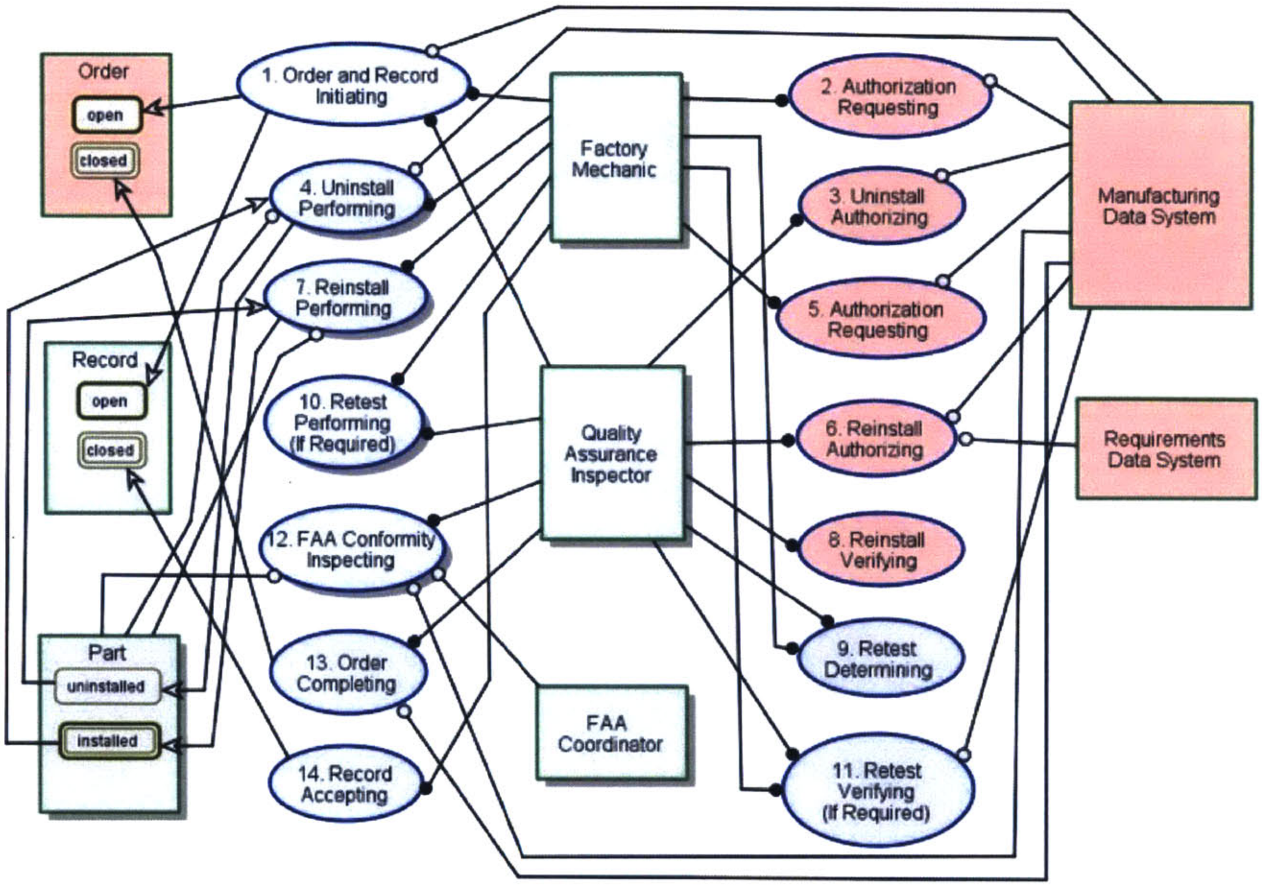

Figure 7.10: OPM of Targeted Improvements (Highlighted in Red)

Optimization 2: Current AA Proposal – Delete Authorizations Conducted by Factory Mechanics (Supporting Processes): Proposed after considering OSI, the concept of Operator Self-Authorization (OSA) has been identified by AA leadership as a potential inverse to OSI. Here, **Factory Mechanics** would self-authorize themselves to perform the uninstallation, which would therefore eliminate the need for Step 2 **Authorization Requesting** (where the **Factory Mechanic** requests authority from the **Quality Assurance Inspector** to uninstall the part or assembly), Step 3 **Uninstall Authorizing** (where the **Quality Assurance Inspector** authorizes the request), and Step 6 **Reinstall Authorizing** (where the **Factory Mechanic** requests authorization to reinstall the part or assembly). OSA would therefore result in three non-value-adding steps removed from the process. Similar to OSI, the perspective of the process architect will determine how the targeted improvement is classified.

Optimization 3: Current AA Proposal – Combined or Automated Data Systems (Supporting Objects): Another improvement being considered is not specific to a process entity. Rather, it is attributed to simplification of object entities. AA uses two different data systems to manage manufacturing and requirement data, **Manufacturing Data System** and **Requirements Data System** respectively. These systems both require manual input each time that information is accessed. While these systems are important, they are classified as non-value-added, because the systems themselves could be deleted without disrupting pure process function; though an alternative for accessing data to accomplish value-adding steps would need to be addressed.

AA is currently studying its requirements for a next generation data system. The AD-OPM BPI method presents a visual platform for which AA can model what types of requirements would also improve process simplification. Here, following the OPI-BPM method of finding a solution-neutral optimization, both systems could be (ii) combined into a single data system to reduce the architecture even further. Depending on preferences of the process architect, this could take the form of (ii) combination, or (i) deletion of one data system and (vi) upgrade of the other for the same requirements. In addition, other opportunities could exist for automating sub-processes to moderate inputs-outputs of the data system to increase efficiency further and eliminate waiting on manual inputs.

Optimization 4: Proposal by Thesis Author – Combining Order into Record (Primary Operand): Opening and closing both an **Order** and a **Record** are currently performed for two different purposes. The **Order** signals work to be performed, while **Record** maintains configuration control, as required by the regulations and standards previously described.

Though practical use varies, the conceptual usage of both these informatical objects is redundant when displayed through OPM. Therefore using the AD-OPM BPI method, a solution-neutral (ii) combination of the **Order** and **Record** to eliminate this redundancy. A solution-specific manner of performing this (ii) combination can now be explored by AA's experts for the feasibility and specific means of implementation.

Optimization 5: Proposal by Thesis Author – Simplifying Order and Record Initiating into Record Initiating (Value Process): One effect of Optimization 4 above is that another optimization occurs: (iv) reducing/simplifying the Step 1 **Order** and **Record Initiating** from initiating both the **Order** and the **Record** to initiating only the **Record**.

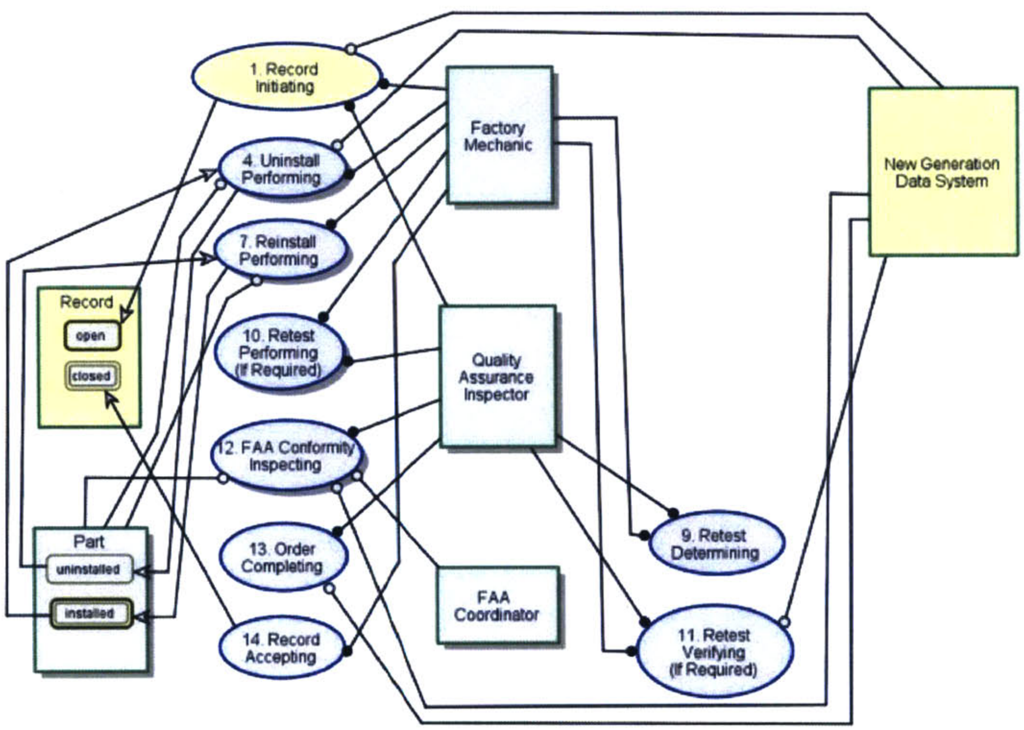

Figure 7.11: OPM of Optimized Uninstall Process (Waste Removed; Consolidations in Yellow)

Summary and Conclusions of Process Analysis and Optimization with OPM

Depending on the choices to be selected by AA's senior management, the OPM process analysis and optimization that completes the AD-OPM BPI method has identified or validated that up to five non-value-adding process steps could be eliminated, four value-adding and non-value-adding objects could be combined into as little as two, and one value-adding process could be simplified. These are identified in Figure 7.11 by the absence of the pink entities that were present in Figure 7.10, and highlighting yellow entities that consume the combined or simplified objects. OPM has identified that these solution-neutral improvements have an optimization effect on the process as a system, without disrupting the value-adding function performed by the business process. Therefore, identification and validation of new options has indeed occurred. The next steps are for these solution-neutral opportunities to be explored by AA, its process management team, and technical experts to find solution-specific means to implement these solutions that are consistent with process function, resources, and other synergies throughout the company.

Chapter 8: AD-OPM BPI Summary and Conclusions

This thesis proposes AD-OPM BPI as a new method of conducting business process improvement by combining AD's system mapping and optimization tools with OPM's object and process modeling capabilities. Through the AA case study, the benefits and limits of each individual method have been illustrated. AD is used to design or redesign a system of processes based on efficient design principles and emphasizing the DPs that fulfill the system's FRs. This approach can be particularly effective for both designing a new system of processes and for improving smaller simple systems of processes, but becomes much more difficult to use for larger systems of complex processes. In contrast, OPM did not demonstrate any difficulty modeling a complex process and can be used for new or pre-existing process sets. OPM's use of modern systems architecture layering principles enables it to partition non-value adding aspects of a process for focused improvement. For the combined AD-OPM BPI method, its application will depend on the limitations of each individual approach. For example, AD's limitation on larger systems of complex processes will result in the AD-OPM BPI method having similar challenges. Even if the scale or scope of a system precludes using AD-OPM BPI, OPM's application as a method to conduct BPI on individual processes remains a viable option.

Bibliographic References

Aviator Aerospace (Pseudonym). 2016. "Procedure-10: Uninstall Part or Assembly." *Unpublished Company Document.*

Aviator Aerospace (Pseudonym). 2015. "Procedure-8: Configuration Management Objectives." *Unpublished Company Document.*

Aviator Aerospace (Pseudonym). 2014. "Policy-3: Quality." *Unpublished Company Document.*

Browning, Tyson R. Ernst Fricke, and Herbert Negele. 2006. "Key Concepts In Modeling Product Development Processes." *Systems Engineering Syst. Engin.* 9 (2): pp.104–128. doi:10.1002/sys.20047.

Cameron, Bruce. 2014. "Functional Architecture And Operations." Lecture. *MIT Course ESD.411: Foundations Of System Design and Management I.* September 23.

Crawley, Edward, Bruce Cameron, and Daniel Selva. 2015. *System Architecture: Strategy And Product Development for Complex Systems.* Hoboken: Prentice Hall.

Damelio, Robert. 2011. *The Basics of Process Mapping 2nd Edition.* New York: Productivity Press.

Dori, Dov. 2002. *Object-Process Methodology – A Holistic Systems Paradigm.* Berlin: Springer Verlag.

Eppinger, Steven D., and Tyson R. Browning. 2012. *Design Structure Matrix Methods and Applications.* Cambridge: The MIT Press.

Harrington, H. James. 1991. *Business Process Improvement: The Breakthrough Strategy for Total Quality, Productivity, and Competitiveness.* New York: McGraw-Hill.

Harmon, Paul and Wolf, Celia. 2014. "The State of the BPM Market – 2014." *Business Process Trends.*

International Standards Organization. 2015a. "ISO 9000:2015: Quality Management Systems – Fundamentals and Vocabulary."

International Standards Organization. 2015b. "ISO 9001:2015 Quality Management Systems - Requirements."

International Standards Organization. 2015c. “ISO/PAS 19450:2015 Automated Systems and Integration – Object-Process Methodology.”

International Standards Organization. 2011. “ISO 9004:2009 Managing for the Sustained Success of an Organization – A Quality Management Approach.”

LeanOhio. 2016. “LeanOhio Overview.” *LeanOhio | Kaizen, Lean And Six Sigma in Ohio State Government*. Accessed April 27. http://www.lean.ohio.gov/.

Mordecai, Yaniv and Dov Dori. 2015. “Agile modeling of an evolving ballistic missile defense system with Object-Process Methodology”. *Proc. SysCon 2015, 9th Annual IEEE International Systems Conference*, Vancouver, Canada, pp. 839-846, April 13-16.

Page, Stephen B. 2002. *Establishing a System of Policies and Procedures*. Westerville: Process Improvement Publishing.

Page, Stephen B. 2000. *Achieving 100% Compliance of Policies and Procedures*. Westerville: Process Improvement Publishing.

SAE Aerospace. 2009. “Aerospace Standard AS9100C.”

Suh, Nam P. 2001. *Axiomatic Design: Advances and Applications*. New York: Oxford University Press.

Suh, Nam P. 1998. “Axiomatic Design Theory for Systems.” *Research in Engineering Design*.

United Technologies Corporation. 2015. “Supplier Quality System Requirements”, *ASQR-01 Rev. 9*. February 2. *Published Company Document*.

U.S. National Archives and Records Administration. 2015. “Quality System.” *14 CFR Part 21.137*, December 17. Amended.

U.S. National Archives and Records Administration, 2009. “Quality Manual.” *14 CFR Part 21.138*. October 16.

Van Eikema Hommes, Qi D. 2015. “System Interfaces On Design.” Lecture. *MIT Course ESD.413, Foundations Of System Design and Management III* March 11.

Whittingham, P.R.B. 1986. “Operator Self Inspection.” *American Society of Quality - 41st Annual Quality Congress*. Vol. 41. pp. 278-286.