2015-12

Un-building blocks: a model of reverse engineering and applicable heuristics

Garcia, Jorge F.

Monterey, California: Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

DISSERTATION

UN-BUILDING BLOCKS: A MODEL OF REVERSE ENGINEERING AND APPLICABLE HEURISTICS

by

Jorge F. Garcia

December 2015

Dissertation Supervisor Robert Harney

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** December 2015	**3. REPORT TYPE AND DATES COVERED** Dissertation

4. TITLE AND SUBTITLE
UN-BUILDING BLOCKS: A MODEL OF REVERSE ENGINEERING AND APPLICABLE HEURISTICS

5. FUNDING NUMBERS

6. AUTHOR(S) Jorge F. Garcia

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

Reverse engineering is the problem-solving activity that ensues when one takes a human-made system, whole or in part, and attempts—through systematic analysis of its physical characteristics and other available evidence—to answer one or more of the following questions: What is this for? What does it do? How does it do it? What is inside it? How was it made? A model developed from a synthesis of the technical literature is used to infer modes of failure in the process of reverse engineering and identify and catalog applicable experience-based techniques known as heuristics. The model is then cast in an executable formal language in order to further test its assumptions, and explore its implications. Hands-on, historic, and virtual case studies are used to validate and refine the model. The modes of failure, heuristics, and the model itself in its original and formal language expressions, introduce a new descriptive terminology of reverse engineering and provide a new framework to interpret real world reverse engineering activity.

14. SUBJECT TERMS
systems engineering, reverse engineering, heuristics, process modeling

15. NUMBER OF PAGES
281

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

UN-BUILDING BLOCKS: A MODEL OF REVERSE ENGINEERING AND APPLICABLE HEURISTICS

Jorge F. Garcia
Commander, United States Navy
B.S., United States Naval Academy, 1995
M.S., Naval Postgraduate School, 2001
M.A., Naval War College, 2011

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN SYSTEMS ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 2015

Approved by: Robert Harney
Associate Professor of Systems Engineering
Dissertation Supervisor

Ravi Vaidyanathan
Professor of Systems Engineering

Kristin Giammarco
Associate Professor of Systems Engineering

Douglas Nelson
Associate Professor of Systems Engineering

Don Brutzman
Associate Professor of Modeling Virtual Environments & Simulation

Approved by: Ronald Giachetti, Chair, Department of Systems Engineering

Approved by: Douglas Moses, Vice Provost for Academic Affairs

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Reverse engineering is the problem-solving activity that ensues when one takes a human-made system, in whole or in part, and attempts—through systematic analysis of its physical characteristics and other available evidence—to answer one or more of the following questions: What is this for? What does it do? How does it do it? What is inside it? How was it made? A model developed from a synthesis of the technical literature is used to infer modes of failure in the process of reverse engineering and identify and catalog applicable experience-based techniques known as heuristics. The model is then cast in an executable formal language in order to further test its assumptions and explore its implications. Hands-on, historic, and virtual case studies are used to validate and refine the model. The modes of failure, heuristics, and the model itself in its original and formal language expressions introduce a new descriptive terminology of reverse engineering and provide a new framework to interpret real-world reverse engineering activity.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

> The steam engine has done much more for science than science has done for the steam engine.
>
> William Thomson, Lord Kelvin

Background

The work begins with a tentative definition of reverse engineering as: "The process for discovering the fundamental principles that underlie and enable a device, object, product, substance, material, structure, assembly, or system through the systematic analysis of its structure and, if possible, its function and operation" (Messler 2013, 16). From this definition, six distinct reasons establish the relevance of the subject matter. First, reverse engineering is important for economic reasons. For example, it is a less expensive alternative to traditional design; it is an enabler for technological leapfrogging (i.e., bypassing the initial stages of a technology development process); it is also an enabler of system longevity, holding particular appeal to a society increasingly concerned with sustainability. Second, reverse engineering is important for reasons of technological competence and currency. It is important if we intend to stay technologically competitive at personal as well as societal levels. Third, reverse engineering is important for pedagogical reasons. It is an excellent way to learn and to teach about engineering and design). Reverse engineering is a multidisciplinary activity (and field of study) closely related to systems engineering, thus practitioners and students from both, could benefit from exchanging knowledge. Therefore, reverse engineering is important to systems engineers. Reverse engineering is a critical component in the evolution of technology, thus understanding it may shed light on the notoriously difficult task of technological forecasting. Finally, reverse engineering is important for defense/military reasons, as it offers a technology-based approach to gaining tactical, operational, and strategic advantages over the enemy.

Problem

Reverse engineering is an important field of engineering study and practice, and yet, it has received relatively little attention from the academic community. Existing literature about reverse engineering of systems other than circuits and software is scarce. The problem is compounded by the fact that the term "reverse engineering" is inconsistently defined, thus the handful of academic books about reverse engineering are often not speaking about the same thing.

Goal

Consequently, this work is undertaken with two broad objectives. First, to provide a meaningful contribution to the study of reverse engineering by introducing a better definition, a new descriptive terminology, and model to serve as a framework for interpreting reverse engineering activities. Second, to provide a meaningful contribution to the practice of reverse engineering by collecting and cataloguing a set of heuristics applicable in that field

Methodology

First, a literature review and synthesis is used to arrive at an improved definition of reverse engineering. The goal is for the new definition to be inclusive of existing work, while at the same time remaining succinct and clear. The analysis culminates in the following proposed definition: reverse engineering is the problem-solving activity that ensues when one takes a human-made system, in whole or in part, and attempts—through systematic analysis of its physical characteristics and other available evidence—to answer one or more of the following questions: What is this for? What does it do? How does it do it? What is inside it? How was it made?

Next, a system model (or diagram) is developed for the analysis of reverse engineering as an activity that centers upon a target system (the human-made system about which the reverse engineer seeks answers). A set of conventions for the depiction of the target system as well as the reverse engineer's actions on the target system is suggested (equivalent to a set of conventions for map making, as well as for charting the progress of travelers upon the terrain mapped). The development of these conventions

shows reverse engineering to be an iterative process of progressively inward inquiry. That is, the reverse engineer discovers information by repeatedly pursuing answers to the same set of questions or problems at progressively deeper levels within the target system's structure. These levels become exposed as the reverse engineer physically interacts with the target system: tearing it down layer by layer.

Third, this process of progressive inward focus is summarized and generalized in the form of a process model of reverse engineering. Specifically, the process model states that reverse engineering involves three phases: (1) Context exploration; (2) System exploration and testing; and (3) Boundary breach. The boundary breach at the system level exposes the context at the subsystem level. Therefore, the boundary breach is followed by a second (physically smaller) context exploration. The exploration and testing for each subsystem ensues followed by subsystem boundary breach, and so forth.

The process model contains an additional layer of description in that it breaks down the system exploration an testing phase into three components: (1) Function discovery; (2) Interface allocation; and (3) Working model generation and subsequent revisions.

Function discovery results when the reverse engineer observes or infers an interaction between target system and its surroundings. An example of this would be the discovery that a particular vehicle can travel along a smooth vertical wall. Interface allocation results when a physical element is discovered in the target system and conceptually linked to a function (perhaps suction cups, or magnets might account for the function in the preceding example). Working model generation refers to the fact that the reverse engineer seeks to arrive at a coherent mental picture that integrates all the functions and interfaces discovered.

A working model is likely to be coarse or even false at the start of the reverse engineering process, and become more refined and truer as the process unfolds. Depending upon the reverse engineer's experience and ability, a working model may precede the reverse engineering activity. On the other hand it may never arrive ("I have taken this apart to the last bolt, and still have no idea how it works").

Arrival at a false or incomplete working model (or at no model at all) represents a failure of the reverse engineering endeavor. Several paths might lead to such an outcome. These are reverse engineering's modes of failure. Two general types of modes of failure are suggested. Either some true information is missed and never incorporated into the working model, or some false information is incorporated into the working model.

After this, the model, its assumptions and its implications are validated through the analysis of a number of case studies. The author performed four real-world case studies (attempting to reverse engineer four different target systems). A fifth case study consisted of the historical and ongoing efforts to reverse engineer an ancient artifact that has come to be known as the Antikythera Mechanism. Finally, five additional case studies (designated here as "virtual case studies") were computer generated event traces derived from the process model cast into an executable formal language (Monterey Phoenix).

Contributions

Analysis of case studies serves to confirm and refine the process model. In addition, the same analysis yields a number of heuristics (rules of thumb or experience based techniques) that may be applicable in the practice of reverse engineering. Finally, the modes of failure, heuristics, and the system and process models (particularly the process model in its formal language expression), introduce a new descriptive terminology of reverse engineering and provide a new framework for thinking and communicating about real world reverse engineering activity.

LIST OF REFERENCES

Messler, R. 2013. *Reverse Engineering: Mechanisms, Structures, Systems and Materials.* New York, NY: McGraw-Hill Professional.

ACKNOWLEDGMENTS

I want to thank my advisor Bob Harney, for taking me under his wing, indulging my many and disparate ideas, guiding me through this adventure, and never failing to stoke my scientific curiosity. I am also deeply grateful toward the rest of my dissertation committee (Kristin Giammarco, Ravi Vaidyanathan, Douglas Nelson, and Don Brutzman), who stuck with me when things got uncertain, and continued to carve impossible amounts of time out of their already full schedules to read my often-appalling early drafts and nudge my work little by little into something I am proud of.

Every word in these pages stands for some amount of time I was absent from home, and my duties as a dad and a husband. So I thank above all, my beautiful wife, Katya, for her rock-solid love against which I could always lean when I felt tired or beaten, and for her sometimes less solid patience (*…if I hear the word "dissertation" one more time!*).

Finally, I dedicate every word of this work to Winston and Isabelle Garcia, who heard their mom say "*papa cannot come, he has to work on his dissertation*" a few hundred times over the last four years. If (and only if) by this example I managed to instill in them a love of learning, and books, and science, it will have been worth it.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

> We live in a society exquisitely dependent on science and technology, in which hardly anyone knows anything about science and technology. Sooner or later this combustible mixture of ignorance and power is going to blow up in our faces.
>
> —Carl Sagan, *Why We Need to Understand Science*

A. BACKGROUND

Occasionally, an engineer is tasked to discover the function, operational principles, or manufacturing processes for a system of which he has very little information. The engineer must then analyze the system—and its context—and attempt to answer questions such as: What does the system do, how does it do it, and how was it built. At present, there exists no framework that allows engineers to develop answers to these questions in a systematic fashion.

This work begins with the idea that the scenario described above, and the tool-set required to deal with it are of particular relevance in the present-day technological environment. Reverse engineering is both important and complex and should be approached as a field of study.

One definition of reverse engineering is "The process for discovering the fundamental principles that underlie and enable a device, object, product, substance, material, structure, assembly, or system through the systematic analysis of its structure and, if possible, its function and operation" (Messler 2013, 16). This definition provides an adequate starting point from which to launch the first question that concerns this dissertation: Why is reverse engineering important?

1. Relevance

A number of reasons are offered in support the claim that reverse engineering is relevant as a field of engineering study in general and to students and practitioners of systems engineering in particular.

First, reverse engineering is important for economic reasons. (1) Reverse engineering takes as its starting point a system whose performance, strengths and weaknesses have already been exposed to and tested by real-world operation and the marketplace, thus the costs associated with measuring and testing these factors can in theory be circumvented or at least reduced. It follows that "firms undertaking creative imitation benefit from lower R&D costs, and in turn lower prices, lower consumer education costs, and lower risk of market uncertainty;" in other words, reverse engineering offers a "smarter strategy for growth and profit" (Kim 1997, 230). (2) Reverse engineering can help develop a technological foundation in a relatively compressed time scale. This was demonstrated in the last third of the twentieth century, on the scale of global economics by South Korea.[1] (3) Reverse engineering is an enabler of system longevity. Components that have reached obsolescence[2] without obvious replacement can be reverse engineered, with the result that larger and expensive systems that would otherwise have required retirement and replacement at great cost can be kept on duty for longer (Ingle, 1994). (4) Reverse engineering appeals to a society increasingly concerned with sustainability, for instance, many retired systems continue to have functioning components. Yet even when the operational principles and other design considerations of the retired systems are well documented it is less likely that their components will be sufficiently documented for future users. With the aid of reverse engineering, these components could be extracted, understood, and re-utilized in other systems. An area of research known as "Design for Disassembly" (Penev, 1996; Siuru, 1990) is driven by a similar imperative as reverse engineering.[3]

Second, reverse engineering is a core competency of modern society where technological diversity and growth rate have outpaced even biology. For reference, there

[1] According to Kim (1997) South Korean Consumer Electronics exports went from $47 Million in 1970 to $22.5 Billion in 1994 the 4th largest producer of consumer electronics in the world -with similar transformations taking place in the automotive and semiconductor industries.

[2] There are different uses of the term *obsolescence.* As used here it means a scarcity of manufacturers or suppliers of the given product or component due to superseding technologies.

[3] One example of the potential impact of these initiatives is given in a lecture by Leyla Acaroglu in a TED Lecture (*Paper Beats plastic? How to Rethink Environmental Folklore* TED2013 · 18:07 · Filmed Feb 2013). In it she cites the 152 million phones discarded in the U.S. in 2012 only 11 percent were recycled.

are approximately 4.7 million types of technological artifacts, compared to only 1.5 million biological species (Basalla, 1988). The traceability and documentation of physical technological systems cannot be taken for granted because of the vast numbers of types involved, and because adequate documentation is not always a high priority for the inventors and designers of new technologies. Additionally, increasingly affordable manufacturing and fabrication technologies and resources are resulting in a diffusion of sources of technology (Siegele, 2014). In other words, the "ecosystem of hardware" is showing characteristics that have previously been a concern only in software. For example, with software, the discovery of architectures that lack supporting documentation is not uncommon. Software may undergo unregulated modifications to the original code due to errors in transcription or intentional departures from the original functionality. These departures can range from well meant but poorly documented upgrades and patches, to the surreptitious insertion of malicious code. The result is that software can diverge between its supposed and documented function, and what it actually does. This may explain why reverse engineering is more firmly established within software engineering than within the other—hardware-centered—engineering subspecialties. However, hardware may soon be affected in similar ways to software by open source coupled with 3D printing, desktop manufacturing and related technologies. This will probably bring about advantages like reduction in cost, and leveraging the vast amounts of expertise and manpower of masses of technologically savvy individuals, accompanied by problems like quality assurance, revision control and lack of documentation. In short, the boundary between hardware and software systems is becoming increasingly blurred (Anderson, 2012).[4]

Third, reverse engineering can be a pedagogical tool (Otto & Wood, 2000; O'Brien, 2010; Halsmer, 2013). However, to effectively implement reverse engineering into an educational program requires a determination of vital elements and, a detailed understanding of the reverse engineering process. Such an understanding is a prerequisite for the fulfillment of reverse engineering as an engineering educational product and learning tool. Beyond its use within the boundaries of engineering education, there are

[4] A Google search for the phrase "Hardware is the new Software" results in about 120,000 hits. An interesting implication is the ongoing debate over 3D Printed Guns (Reynolds 2014).

broader implications in the adoption/acceptance of reverse engineering by society to understand and interact with technology. As mentioned earlier, the example of South Korea is instructive. This is a nation that has embraced reverse engineering at a state-scale. The result is one of the fastest rates of economic growth in modern times (Kim, 1997).

Fourth, reverse engineering is important because it is a cross-disciplinary undertaking calling for a systems approach. This suggests that reverse engineering and systems engineering are closely related disciplines and can benefit from exchanging knowledge. In fact, it may be best to think of reverse engineering as a subset of systems engineering. Yet the lack of acknowledgment of this fact is glaring: the International Council on Systems Engineering (INCOSE) does not even have a definition of reverse engineering.[5] At present, reverse engineering consists of a number of diverse activities that are learned and practiced separately.

The following is a sampling of activities that incorporate reverse engineering:

1. Reverse engineering of components for aging systems: out-of-production components are often reverse engineered in order to extend the life of large and costly systems of which they are part. (Ingle, 1994)

2. Reverse engineering of interfaces for new systems: interfaces are often reverse engineered in order to provide third party aftermarket accessories to cell phones and other personal electronics.

3. Explosive ordnance disposal: an activity that—like reverse engineering—relies heavily on the process of systematic disassembly, and the need to know, or safely and efficiently learn, what is inside and how it works.

4. Archaeology: attempting to uncover the function of ancient machines, or the technologies that enabled puzzling feats of architecture, archeologists may find themselves in the role of reverse engineers. (Shelley, 1996)

5. Forensic engineering: as reverse engineers must consider operational systems in search of unknown functions, forensic engineers consider failed systems in search of unknown causes (Casey, 1998).

6. Materials science: examining the chemistry and microstructure of parts and components in order to infer performance specifications or methods of

[5] *Handbook of Systems Engineering* (Haskins 2006) and other INCOSE publications and websites were searched for a definition of reverse engineering; none was found.

fabrication, materials science techniques can supplement the knowledge of the reverse engineer (Shipman, 2013).

7. Political/military intrigue: an American UAV downed and taken by an unfriendly nation's military raises the question: how much can they learn from it? (Axe, 2012)

8. Cultural phenomena: personal electronics, telecommunications networks, and a variety of other systems are "hacked"—their software and electronic components reverse engineered—in order to circumvent manufacturer-imposed restrictions or for a variety of other purposes (Grand, 2011).

9. Battlefield forensics: Weapons of unknown origin found in the combat zone can be torn down and studied with the object of designing effective countermeasures or of tracing their provenance back to their country of origin.

10. Safe cracking: the systematic defeat of physical security systems is another activity that—like reverse engineering—focuses on the discovery of hidden functions and features, and the breaching of system boundaries for a variety of purposes (Blaze, 2004).

All of these activities are or incorporate reverse engineering. Studying them as one discipline could result in productive synergies to advance the general understanding and practice of reverse engineering and thus of systems engineering.

Fifth, reverse engineering plays a critical role in the evolution of technology. As existing technologies are explored, tinkered with, understood in new ways and creatively repurposed, new technologies arise out of the parts of old ones (Arthur, 2009; Kelly, 2010). The individuals responsible for this process are often not scientists, nor even engineering experts from within the field that generated a particular old technology, but rather "outsider tinkerers" who come to understand the old technology on their own terms, through their own experiments. (Rosen, 2010; Basalla, 1988). The mechanism for this hands-on understanding is reverse engineering or something very like it.[6]

Lastly, reverse engineering is important now, and has always been important to the military. Reverse engineering has been long employed in the sidelines of warfare. Most

[6] A case for the process of innovation in general (not only in the context of technology) as a continuous re-shuffling and re-purposing of old elements is shown eloquently and entertainingly in a series of short videos by Kirby Ferguson entitled "Everything is a Remix." The series can be found in the author's website everythingisaremix.info and in YouTube. Its last episode includes a thought-provoking criticism of the patent system.

weapon systems are to a lesser or greater extent, designed based on a threat whose specifications and capabilities are often learned through reverse engineering of captured equipment, footage, soldiers' accounts or other such materials. There are famous cases of military reverse engineering. For example, during World War II Russia captured and reverse engineered the American B-29 bomber and the Allies captured and reverse engineered the German V-2 rocket (Messler, 2013). While these cases may appear dated, the practice of military reverse engineering remains alive today. One expert reports that,

> In 2012, DSS [Defense Security Service] found that the top four most targeted [by economic espionage for reverse engineering] technology categories were… information systems; lasers, optics and sensors; aeronautics systems; and electronics. Armaments and energetic materials came in fifth, with a growing interest in technologies for processing and manufacturing, directed energy, and space systems. (Van Cleave, 2013)

There are specific recent examples such as Iran's capture and claims to have reverse engineered an American RQ-170 UAV (Axe, 2012).

In spite of the high profile nature of these examples, technical literature about actual cases of reverse engineering used by the military is sparse. A primary military use of reverse engineering appears to be as a means to overcome obsolescence (Ingle, 1994).

2. Statement of the Problem

There is a moderate amount of literature that addresses a limited number of subjects in reverse engineering. However, there appears to be no work to date directly concerned with one of the central questions of this work: *What is it—in general—that reverse engineers do?*

The previous section attempted to establish the importance of reverse engineering. In the process, the existence of a problem has been suggested. Explicitly, the problem has three parts. First, there exists no adequate general definition of reverse engineering. Most of the authors of related technical literature offer their own definitions that either ignore or openly criticize other points of view (Messler, 2013; Raja, 2008). Part of the reason for this probably resides in the fact that most of the technical literature embraces a narrow view of reverse engineering. A broad, general view and definition of reverse engineering are called

for. Second, there exists no language of reverse engineering. It is contended here that all of reverse engineering shares a number of common elements. For example, reverse engineers will almost always be involved in some form of physical opening, disassembling, or tearing down of the target system.[7] These engineers are also likely to find themselves engaged in an effort to infer function from form. These elements and others that are part of reverse engineering in general have not been named or described in a way that can be applied to all reverse engineering endeavors. Finally, a consequence of the lack of a definition and lexicon of reverse engineering is that there is no growing or accumulated body of knowledge that addresses the relevant issues under a common heading. Reverse engineering does not seem to have the status of field of knowledge. In summary, reverse engineering is an important field of engineering study and practice that has received little attention from the academic community. This statement of the problem will be further addressed as part of the literature review in Chapter II.

3. Objectives

The objective of this work is to provide a meaningful contribution to the field of reverse engineering. This contribution will have two general components: first, it will produce a model of reverse engineering; second, it will establish a set of heuristics applicable in the context of reverse engineering. A model can be used as a theoretical framework for thinking about this process, and also serve as the basis for the analysis of reverse engineering activities. Among other things, this analysis may help in the identification of useful heuristics. The heuristics will be a contribution to the practice of reverse engineering. As with heuristics in other fields, the goal is to present a list of experience-based techniques that that will help navigate the unique set of problems presented by reverse engineering.

[7] The term *target system* originated with this research. It is used to refer to the technological physical system under consideration as part of a reverse engineering effort.

B. METHODOLOGY

The methodology described herein can be broken down into four parts.

First, there needs to be a methodology for arriving at a general definition. The existing literature of reverse engineering will be reviewed. Various definitions and other discussions pertaining to reverse engineering from the technical literature will be compared and contrasted (Messler, 2013; Ingle, 1994; Otto & Wood, 2000; Rekoff, 1985; Chikofsky, 1990). Gaps in the academic coverage will be pointed out. Important works on subjects that are related to reverse engineering will also be reviewed (Shelley, 1996; Gigerenzer, 1999; Arthur, 2009; Kelly, 2010 et al.). Through this exploration of literature of reverse engineering and related topics, the notion will emerge that reverse engineering is a subset of a more general category of activities: problem-solving activities. Reverse engineering encompasses a unique set of problem types. For example, how does one disassemble systems that were not meant to be disassembled? How may reverse engineers overcome irreversible assembly methods such as gluing or welding, access-denial design features such as anti-tamper mechanisms? How can one infer purpose or function, or lack thereof, from physical attributes and apparent interfaces?

From this study, the concept of heuristics will emerge. Heuristics hold a central role in the study and practice of problem-solving in general. Finally, a general definition of reverse engineering will be proposed to synthesize the information gathered in the literature review described in depth in Chapter II.

Second, a methodology for developing a language (and a model) of reverse engineering needs to be presented. The definition arrived at in Chapter II will serve as a basis for directed speculation. Specifically, we will ask: *Given that definition, what might reverse engineering look like? What practical considerations might it bring forth?* These questions are the focus of Chapter III. It is an attempt to develop a general sense for the genuine issues faced in reverse engineering by considering each piece of the definition as if we were in fact *doing it*. After this, the focus will shift to the development of a *System Diagram for the Analysis of Reverse Engineering* (also known as a system model). The objective is to arrive at a visualization of physical systems that can be used as a map for

operations of reverse engineering to be expressed and recorded. The development of this *System Diagram for the Analysis of Reverse Engineering* will be the subject of Chapter IV. Finally, the practical considerations and the system diagram will be brought together in order to describe a useful general model of the reverse engineering process; this will be the subject of Chapter V.

A third area for analysis in this section is a methodology for the discovery of a set of heuristics of reverse engineering. The proposed model of reverse engineering will suggest a means to identify and catalog a set of heuristics as follows: The model shows reverse engineering to be an iterative process of information gathering. According to this model, reverse engineering is composed of four stages that are repeated as the process "drills down" from system to subsystem, to component, to subcomponent, and so on. In each stage, the reverse engineer's attention and activity are directed toward obtaining a particular type of information, or solving a particular type of problem. The type of information or problem that occupies the reverse engineer at a given point can be described in terms of the hierarchical structure of the target system, and the kinds of objects and relationships of which a system is made up. Information gathered in one stage accumulates until the reverse engineer decides to proceed to the next stage. Steps taken and information gathered in a given stage have important effects both on the objective completeness of information available in subsequent levels of analysis, as well as on the reverse engineer's subjective ability to discern this information accurately. In other words, actions driven by the reverse engineering process in one stage could result in a loss of information, or in predisposing the reverse engineer to discover a less than complete amount of information in a subsequent stage. Also, when information gathered in one stage is both accurate and complete, the reverse engineer's task in the next stage is more likely to succeed. Given this model, we see that transitions between two stages are critical events. A transition may be considered unsuccessful, or partially successful, if the reverse engineer has attained only incomplete or inaccurate information from a previous stage before moving to the next. The model itself suggests the reasons why these unsuccessful transitions, henceforth called modes of failure, may occur. These modes of failure represent the essence of the types of problem we expect to encounter when doing reverse engineering. By definition heuristics

of reverse engineering are those experience-based techniques that can aid the reverse engineer in avoiding modes of failure. An exploration of the modes of failure in reverse engineering is the subject of Chapter VI.

Fourth, there needs to be a methodology for obtaining and cataloguing a set of heuristics while validating and refining the model of reverse engineering. In order to identify a set of heuristics used during actual instances of reverse engineering, the author undertook a number of case studies. These were subsequently analyzed through the lens of the model as follows: Where a theoretical mode of failure became an actual pitfall in practice, we identify the presence of a heuristic by asking: *How might this have been avoided?* Where a theoretical mode of failure was circumvented in practice, there we identify the presence of a heuristic by asking: *Why was this potential mode of failure not an issue here?* Finally, heuristics can be grouped according to the mode of failure to which they are applied to solve. This part of the work is the subject of Chapter VII.[8]

C. OVERVIEW OF THE WORK

This chapter establishes the importance of reverse engineering as a field of study. This is followed by a statement of the problem: the academic state of the field of reverse engineering is not commensurate with its importance. The objectives of the work are then stated: To contribute meaningfully to the understanding of reverse engineering by offering a precise definition, creating a model, and a identifying a set of applicable heuristics. It is implied that the essence of reverse engineering is that it—like other fields of engineering and design—is a type of problem-solving. It follows that heuristics must play a central role in the theory and practice of reverse engineering. From this emerge the objectives of this research: First, to provide a model of reverse engineering and second, to provide a set of heuristics applicable in the unique context of reverse engineering.

[8] The initial set of case studies consisted of four hands-on reverse engineering projects. This was later supplemented by a historic case study. These five real-world case studies were eventually supplemented by five simulated case studies (See Chapter VII and Appendix B). The use of three distinct approaches provided additional confidence in the soundness of the model.

II. LITERATURE REVIEW AND SYNTHESIS

> When we are trying to make out the nature of a confused and unfamiliar object, we perform various acts with a view to establishing a new relationship to it… we turn it over, bring it into a better light, rattle it and shake it, thump, push, and press it, and so on… the intent of these acts is to make changes which will elicit some previously unperceived qualities, and by varying conditions of perception shake loose some property which as it stands blinds or misleads us
>
> —John Dewey, *The Quest for Certainty*

A. INTRODUCTION

The introduction to the previous chapter established the relevance of reverse engineering on the basis of its potential contributions to economy in design and manufacture, technological competence, engineering pedagogy, synergistic research with other fields, and the pursuit of military advantage. The second problem introduced was that the literature of general/hardware reverse engineering is scarce, and where it exists it has remained narrowly focused. For clarification, the literature of reverse engineering is in fact only *scarce* when one excludes books and papers about the process as it applies to software and integrated circuitry. The practice of software reverse engineering and the technical literature to support or describe it are prevalent due to the "very low barrier of entry. Any willing youth with access to a computer and an Internet connection can..." take up reverse engineering of software (Grand, Russell, & Mitnick 2004). As will be discussed, a similar situation is now arising with hardware. The entry barrier for participating in reverse engineering of hardware is lowering.

Once these two subjects are excluded, not much remains. While it is obvious that a general approach to the study of reverse engineering should hold relevance to the practice as it applies to circuitry and software, the converse is less true. A study dealing with a narrow application of a subject is likely to be devoted to problems that are only relevant within the narrow application. Therefore, this delimits the area where the literature review takes place: in reverse engineering NOT merely reverse engineering of

circuitry and software.[9] Through a review of the literature that fits these parameters, this chapter will expound on the gaps in the academic coverage of reverse engineering. The review will also encompass work that is not nominally about reverse engineering but is about closely related academic and practical fields. Finally, a synthesis of the existing, missing, and related work will be offered in the form of a definition that answers the question: What is reverse engineering?

B. ANALYSIS

1. Messler

In the previous chapter the definition of reverse engineering given by Messler (2013) was suggested as a starting point. Messler's *Reverse Engineering—Mechanisms, Structures, Systems, and Materials*, has the merit of being the only book that attempts a comprehensive look at reverse engineering. More than any other author writing about this subject, Messler recognizes the importance of reverse engineering across society and throughout history, and he attempts to present it as the multidisciplinary study that this dissertation also asserts it to be. He alone brings subjects such as history of technological innovation, practical aspects of product tear-down, forensic engineering, archeological reverse engineering, materials science, methods of construction, design, form and function, and even the law and ethics of technology-copying under the unifying heading of reverse engineering. The information presented in this book tends to proceed from personal anecdote—the author has experience as a reverse engineer. This in itself is not a fault, especially when pioneering a new field. However, Messler's coverage of the many subjects in his book tends to be idiosyncratic and opinionated rather than academic. Subjects on which Messler does not have professional experience (such as the pyramid of Khufu, or the Antikythera mechanism) appear to be loosely researched from one or two online sources and are presented in a rambling style that tends to convolute facts and opinions. For example, there is a chapter on the problem of inferring function from form. This is a serious philosophical problem (Kroes, 1998; Newberry, 2013; Ridder, 2007;

[9] Some of the literature reviewed is primarily about software, yet includes some discussion of how reverse engineering pertains to other technological systems. For example (Chikofsky 1990).

Vaesen, 2011 et al.). Messler's use of vague and overlapping definitions for concepts like role, purpose, and functionality, or form, fit and function does little to help clarify the nature of the problem. The effect on the reader is that of knowing that here he has been exposed to some interesting questions, but that they have been poorly answered. Nevertheless, because of its scope and novelty, Messler's remains an important book.[10]

2. Ingle

A search of the literature of reverse engineering (excluding circuitry and software) will turn up K. A. Ingle's book, *Reverse Engineering* (1994), somewhere near the top—possibly because of the book's title. However, Ingle is concerned with a very limited definition of reverse engineering as a process that seeks to improve (i.e., make cheaper without loss in performance) a finite number of components within a system. According to Ingle—whose experience appears to be closely tied to reverse engineering of components for large military systems—the reverse engineer begins by considering a system believed to be underperforming and by looking for components within that system that make for good reverse engineering candidates. To highlight the usefulness of reverse engineering in this context, Ingle produces a graph (reproduced here as Figure 1) that contrasts the life expectancy of military systems with the much shorter life cycle of some of its componentry.

[10] For this reason, Messler was used as the textbook for a class in Reverse Engineering taught by the author of this dissertation.

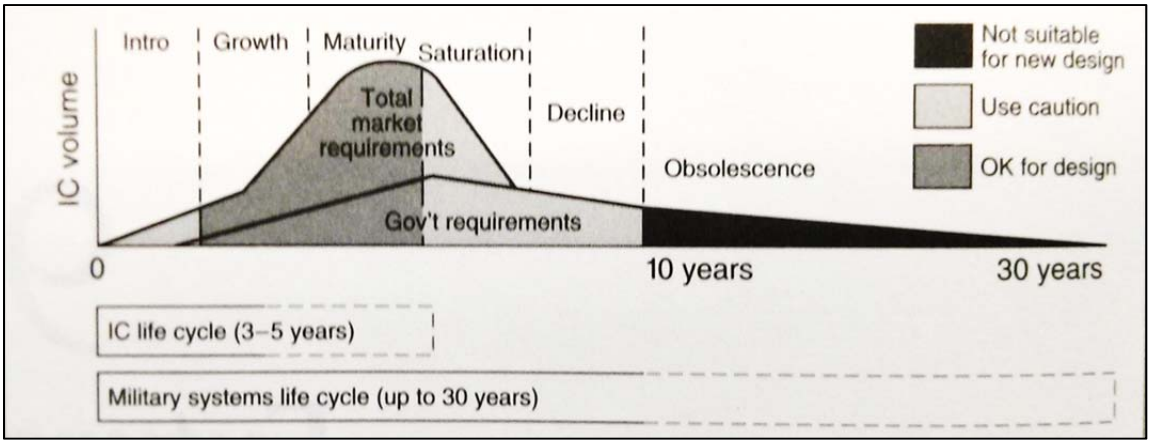

Figure 1. Integrated-circuit production over time

Shown here, the disparity between the life expectancy of military systems (approximately 30 years), and the life-cycle expectancy of its integrated circuit components, which reach maturity within 3–5 years, and obsolescence within 10 years. Source: K. Ingle, 1994, *Reverse Engineering*, New York, NY: McGraw-Hill Inc., 140.

Reverse engineering and subsequent reengineering[11] of a judiciously selected component will result in improved system performance. For Ingle the determination of good candidates for reverse engineering is central and directly correlated to return on investment. Return on investment in turn is linked to whether the component is complex (high complexity is a poor candidate for reverse engineering), and whether it is expensive, heavily used, or exhibits a high failure rate (these all suggest a good candidate for reverse engineering). While this is an undeniably practical approach, it necessarily steers reverse engineering away from complex and more interesting problems. It also gives no consideration to any application of reverse engineering outside the realm of "small" components. In spite of the book's focus on replacing underperforming components and its driving concern with return on investment, Ingle does introduce several practical applications of reverse engineering of varying intensity. At one end of the spectrum are undertakings like product verification and data enhancement, where reverse engineering is used to improve the existing documentation for a given

[11] A reverse engineered system and a reengineered system are related but distinct things. A reverse engineered system is the subject of a process that draws information from the physical system. A **reengineered** system is a "new" system has as its starting point the information obtained from the reverse engineered system.

component. At the other end of the spectrum lies the full reverse engineering and reengineering of a candidate component such as a circuit board or a valve.

Notably Ingle never seems to acknowledge the usefulness or possibility of reverse engineering entire systems. This is representative of the majority view of reverse engineering among the literature reviewed. For example the *U.S. Army Reverse Engineering Handbook* (1987) mirrors the concerns and procedures voiced by Ingle.

In final analysis, Ingle's treatment of the subject has several relevant threads. For example, one important theme that is brought up throughout the book is the importance of treating reverse engineering as a multidisciplinary effort. There is an intriguing suggestion that reverse engineering could serve as an agent for cultural change within industry (Ingle, 1994, 141). However, Ingle's *Reverse Engineering* is generally not theoretical but procedural—the thrust of the book is to offer a step-by-step approach to successful reverse engineering in the somewhat limited sense that she has assigned to it.[12]

3. Wang

It is true of most authors writing about reverse engineering, that the first chapter or two of their book tends to extol the value of reverse engineering in the general sense used here, but that they subsequently focus on a more narrow view of reverse engineering for the rest of their book. So it is with W. Wang and *Reverse Engineering—Technology of Reinvention* (2011).

In the first chapter Wang introduces an interesting take on the relevance of the subject. Reverse engineering—he postulates—is a sort of mechanism for technological cross-pollination. If technology evolves through combinatorial evolution (Arthur, 2009), reverse engineering is a process that un-restricts the flow of information, accelerating the evolution of technology. Wang also introduces the role of reverse engineering in spurring continued growth in technologies that have reached maturity and would therefore otherwise stagnate. For example, he believes that the evolution of aircraft technology

[12] According to Ingle, the stages of reverse engineering are: Data collection (documentation, specs, tests record etc.); Visual inspection; Disassembly; Material analysis; Operational testing; Failure analysis (if applicable); Technical data generation; Prototyping (which is not always necessary); Testing (which is not always possible); Failure analysis and redesign (of prototype, if applicable).

would have ended (or stagnated) once industry reached the point—in the recent past—in which commercial airplanes are fast and reliable "enough." However, he also posits that the existence of reverse engineering enables competition, and spurs continued advances. In support of this view, Wang quotes the Supreme Court "the competitive reality of reverse engineering may act as a spur to the inventor, creating an incentive to develop inventions that meet the rigorous requirements of patentability." (Bonito Boats v. Thunder Craft Boats Inc.)

In subsequent chapters Wang shifts focus to the reverse engineering of homogeneous material components whose key characteristics arise from geometry, tolerance, and material choice and treatment. As he points out, "nuts and bolts are among the most frequently reverse engineered parts in aviation industries" (Wang, 2011, 273). In particular, the book is structured around a number of technologies and sciences that can be used as tools by the reverse engineer whose target is the homogeneous material component. These include: scanning/imaging technologies; materials science with focus on inherent material characteristics (such as hardness or ductility), on failure modes (such as fatigue or corrosion), on identification of chemical composition and microstructure (such as spectroscopy, x-ray analysis, or scanning electron microscopy); and statistics (as applied to dimensional measurement and questions of tolerance).

The bulk of Wang's work is thus focused on a specific set of tools. However, *Technology of Reinvention* does offer a number of insightful observations of general applicability. Wang suggests, for instance, the existence of a synergistic relationship with forensic engineering: the reverse engineer asks "What is going on in here?" while forensic engineer asks the similar question "What went wrong here?"

Wang also brings attention to the fact that reverse engineering must confront questions of long-term performance and reliability. The reverse engineer must not only ask "what does this thing do and how?" but also "how does it continue to do it reliably over an extended period of time—or if it does not, why does it fail to do so?" He also points out that judgments about reliability must be considered in balance against judgments about marketability. For example, it may be a success if reverse engineers uncover the means to reengineer a part that lasts only half as long as the original, if it also

costs one tenth of the price to produce and replace. He also introduces the notion of *signatures* such as surface texture or hardness, which may be used as cues pointing toward the method of manufacture. He points out that one of the challenges of reverse engineering of parts lies in the question of interoperability. Compatibility between the original component and the system of which it is a part is likely to have been achieved through the original process of design and manufacture. However, a part reinvented using reverse engineering does not have the benefit of such processes of co-development.

In the end, while it remains focused on the narrow application of reverse engineering to homogeneous material components, this book does bring up a number of interesting questions. Perhaps more importantly, it includes a number of persuasive reasons to support the claim that reverse engineering is important. In the final chapter, Wang gives the following reasons to do reverse engineering: to learn, to provide a service in relation to a product (perhaps education, or maintenance), to change or repair an existing product, to develop compatible ancillary products or accessories, to create a clone of the product (Wang, 2011).

4. Otto and Wood

Product Design: Techniques in Reverse Engineering and New Product Development (Otto & Wood, 2000) was the largest book in this literature review. The authors are both professors of engineering product development. The book is supposed to be based on the authors' teaching experience. Their approach is characterized by the use of reverse engineering as a context upon which they teach their lessons in design. It should be noted that in addition to Otto and Wood, several of the works reviewed for this dissertation fall under the same category of reverse engineering as a teaching tool (Halsmer, 2009; Hess, 2000; Martinez, 2013; O'Brien, 2010; Ogot, 2006; Rad, 2012; Shooter, 2008; Sheppard, 1992; and Wankat, 1992). Their writings offer varying degrees of evidence to show that a course in reverse engineering increases a student's grasp of essential engineering and design principles.

Otto and Wood's book raises a number of important issues. It emphasizes the pedagogical value of using real physical technological systems as the basis for education.

In contrast with the stylized problems and solutions found in most math, science, and technology books, the use of real systems presents at least three advantages. First, it shows real-world solutions to real-world problems. Second, it is more apt to motivate learning, especially when students—who are presumably already interested in technology—find the particular product under consideration to be interesting or relevant to their lives. Third, the fact that a technological system can be understood at varying levels of abstraction/depth means that a suitable technological system can serve as a platform for learning at any level, from introductory to advanced.[13] According to Otto and Wood, they have routinely employed reverse engineering as the basis for teaching customer needs analysis, functional modeling, optimization, and design of experiments.

One problem—from the point of view of this work—is that Otto and Wood's book is only incidentally about reverse engineering. It is really a book about design. Redesign is almost always the end goal. Reverse engineering is only considered insofar as it supports design, or teaching it. Questions for example, about the nature of reverse engineering, or about the usefulness of reverse engineer to other practical endeavors like the assessment of secure systems, the design of military countermeasures to the target system, or the deciphering of technology-based archeological mysteries, are left unasked.

The authors do bring up many practical points. In preparation for reverse engineering, for example, the problem solver must begin by asking what rather than how. These answers will evolve into functional models that will in turn play an important role in guiding the physical steps of the tear-down. The reverse engineer must begin by ensuring he has the necessary tools. Once again, in preparation for the undertaking, the reverse engineer must thoroughly examine external clues that include the available packaging, instructions, and manuals. Also, orderly and fruitful disassembly will avail itself of the use of photographs, careful measurements, and meticulously kept bills of materials. Finally, destruction is often an unavoidable part of reverse engineering; however, destructive/irreversible steps should be deferred to the later stages of the process.

[13] The author of this dissertation holds the view that reverse engineering is a valuable competence even at the entry level/for the non-engineer.

Another unique contribution of this book is the inclusion of the concept of heuristic as a part of the activity of reverse engineering. This is a concept that will have a central role throughout this dissertation. However, Otto and Wood offer only a limited discussion of heuristics use in reverse engineering. They present three heuristics or proven techniques for accomplishing the task of modularization.[14] These are *Dominant Flow*, *Branching Flow*, and *Flow Conversion* (Otto & Wood, 2000, 170–180). The three heuristics roughly operate as shown in Figure 2. The three images in the bottom show the different results obtained depending on whether one chooses to define a module in terms of continuous flow, flow branching, or flow type. It is noteworthy that each heuristic yields a different answer. Even a single heuristic does not yield a unique answer. As will be shown later, this apparent shortcoming is one of the defining characteristics of heuristics.

[14] Modularization or partitioning is a cognitive step; a way to think about the system in order to facilitate our analysis of it. It is also part of a model of reverse engineering that will be derived in Chapter V of this dissertation.

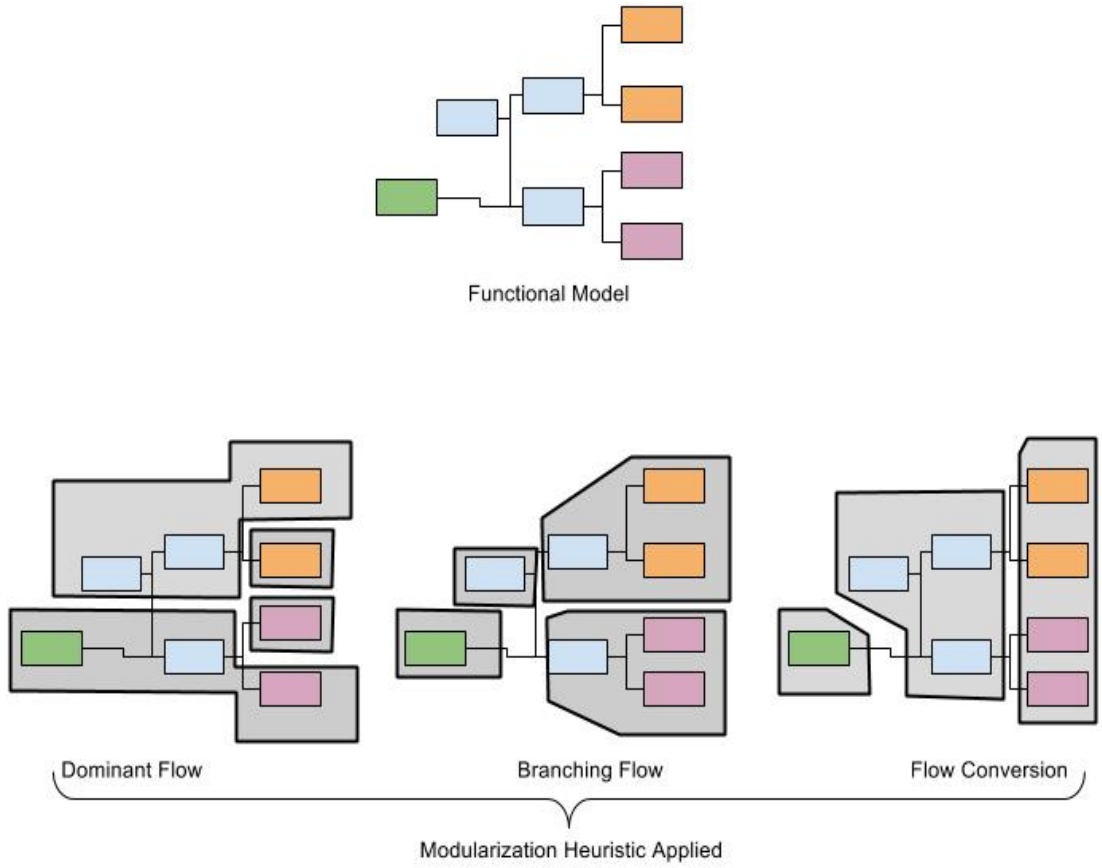

Figure 2. Three heuristics for finding modules in a design

A functional flow diagram for a generic system is shown at the top with no attempt to identify its modules. The results from applying three heuristics for module identification are shown in the bottom. Adapted from: K. Otto, and K. Wood, 2000, *Product Design: Techniques in Reverse Engineering and New Product Development*, Upper Saddle River, NJ: Prentice Hall, 170–180.

In their discussion of functional models, Otto and Wood unintentionally also bring up a problem that has remained a lurking—if not central—concern throughout this work: The problem is that while the two professors extol the virtues of the functional model as a preliminary step in reverse engineering and design, they seem oblivious to the fact that the functional models they show at best fall short of the explanatory power of the physical systems they purport to explain; at worse, the models are virtually unrecognizable as representations of these systems, and they clarify nothing. For an illustration of this problem, consider Figure 3, in which a nail clipper—a system with

three barely moving parts—is represented in a functional diagram that involves 27 boxes and approximately 40 arrows of four different thicknesses.

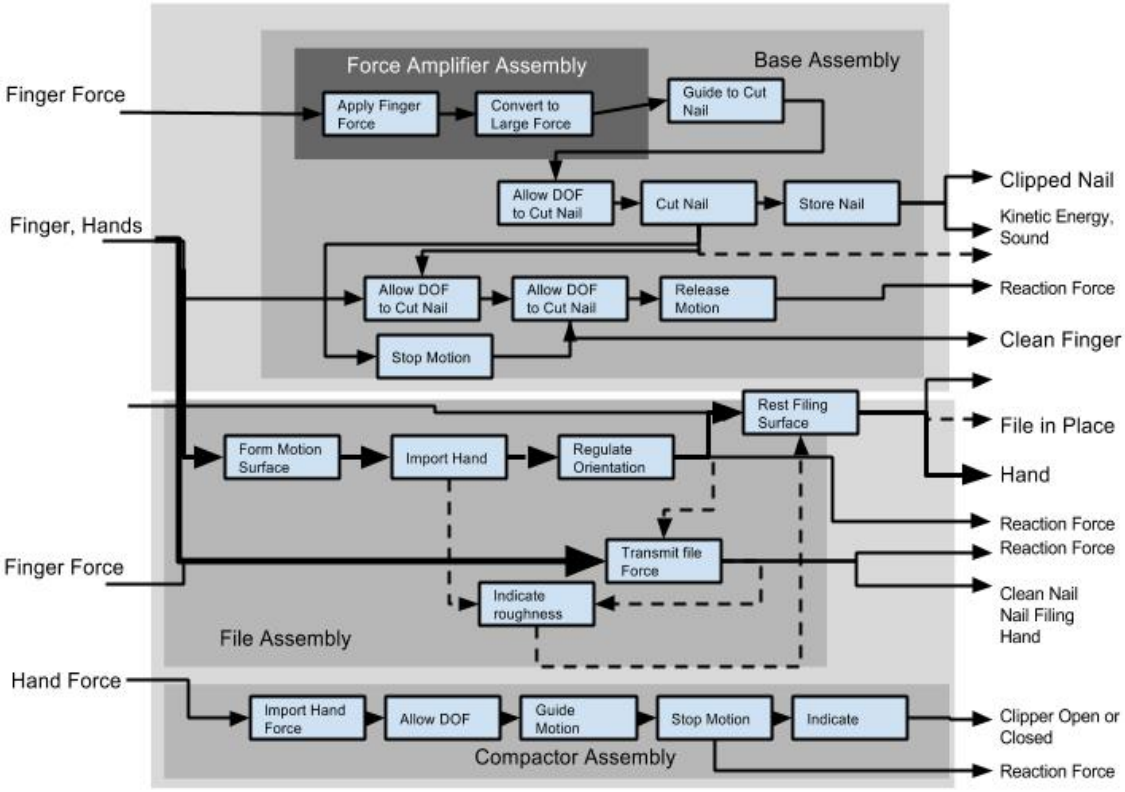

Figure 3. Functional decomposition of a nail clipper

One author uses a similar figure to explain functional decomposition. The original figure contains a significant number of labels and details that have been omitted here. Adapted from: K. Otto, and K. Wood, 2000, *Product Design: Techniques in Reverse Engineering and New Product Development*, Upper Saddle River, NJ: Prentice Hall, p. 175.

In addition to the doubtful usefulness of such a diagram, its complexity also points to a practical problem. If this is what it takes to do a functional analysis of a nail clipper, this method does not seem practical beyond the realm of extremely simple mechanical devices.

Another contribution of Otto and Wood is the concept of Subtract and Operate (SOP). This may be the most basic heuristic for reverse engineering. The idea is that in order to identify the function of a component where no other clues are available, a

procedure can be employed wherein the system is operated in successive states as follows: (1) operate the system in its normal configuration; (2) remove the mystery component then operate the system again. (3) The difference in the system-level behavior between the two states is the function of the mystery component. In this way, the functions of multiple components may be elucidated, one at a time. In spite of the apparent logic of SOP, this author believes the technique to be of very limited usefulness in reverse engineering of all but the simplest of machines. The reason is that it ignores the existence of combined functions—functions that emerge only when two or more components all perform their respective role together. Secondly, this procedure offers no way to account for a component that performs a redundant or backup function. Finally, as the name implies, Subtract and Operate ignores the possibility of nonlinear component behaviors. For example, components that multiply, inhibit, or modulate the output of other components, or that have any role that is not a simple, mechanically additive function. Perhaps SOP can be used as the basis for another technique.

There is one final contribution from Otto and Wood that is worth mentioning. The authors caution the prospective reverse engineer/designer against bias (Otto & Wood, 2000, 222). An effective reverse engineer will—according to them—abstain from thinking that he knows how a system operates prior to the actual tear-down. If this is impossible, then the reverse engineer will at least abstain from bringing this presumed knowledge into his conduct of the tear-down. This is undoubtedly difficult if not outright paradoxical, as the reverse engineer is also urged to form a model prior to—and in order to guide—the tear-down. Nevertheless, later chapters of this dissertation will show that this warning is in effect important.

5. Raja

Unlike other works, *Reverse Engineering: An Industrial Perspective* (Raja, 2008) dispenses from the outset with grand notions of the generality of reverse engineering. As used by Raja, reverse engineering is not an endeavor concerned with system functions, operational principles, or even manufacturing methods. While Raja acknowledges the usage of the phrase “reverse engineering” to denote an activity whose goal is “to reveal

the inner workings of a system to figure out what makes it tick... To develop a high level description of a system without a priori knowledge," he explicitly states that such an activity is not what this book is about (Raja, 2008, 12–15).

Insofar as Raja is concerned, reverse engineering is the reproduction of parts, through computer aided design (CAD) technologies. He provides an overview of the methods whereby CAD-relevant system information can be obtained (i.e., contact, non-contact, and destructive methods).

While this book's narrow treatment is not especially insightful about the general process of reverse engineering in the sense considered important here, it did stimulate a research thread that may prove productive. For Raja, the digitized geometrical representation of the system constitutes a complete model. In response to this point of view, an interesting question can be raised: what constitutes a complete model of the system for the purposes of someone concerned with reverse engineering in the general sense? This will be the subject of Chapter IV.

6. Rekoff

Aside from the books just discussed, there is not much else written on the subject of reverse engineering as the subject is being pursued here. However, one other comparatively short work does stand out. In "On Reverse Engineering" (Rekoff, 1985), M.G. Rekoff makes many observations that are precisely relevant to the general view of reverse engineering. To start, Rekoff introduces the notion that reverse engineering is a special case of systems engineering. He then gives an overview of the pervasiveness of the practice (similar to the one given elsewhere on this document) stating that "Reverse engineering might seem to be an unusual application of the art and science of engineering, but it is a fact of everyday life." (Rekoff, 1985, 244)

According to Rekoff, the reverse engineering process is usually undertaken with the ultimate goal of a hardware reproduction which may be in one of two forms: a ***clone*** or a ***surrogate.*** A clone reproduces the form, fit, and function of the reverse engineered original—for example a vacuum tube fails and is reverse engineered—the resulting product is vacuum tube (although possibly manufactured with more modern technologies

than those that produced the original). A surrogate reproduces the form and maybe the fit, but not necessarily the function—for example, if the vacuum tube is reverse engineered, and the end result is a transistor whose current/voltage specifications reproduce the original, as do the dimensions and placement of the leads, but the operational principle has been supplanted.

Rekoff observes that the reverse engineer is engaged in a task that requires him to gather two distinct kinds of information: functional and dimensional. He suggests that a complete functional analysis must precede the dimensional analysis.

Rekoff also introduces an important problem: unintentional obfuscation. That is he suggests that there might be elements in a design that could draw the reverse engineer's attention and resources away from the main task of identifying functions and real design choices. Rekoff observes that an apparent interface with an unknown function may turn out to be nothing more than the byproduct of a manufacturing process or perhaps a reflection of constraint-driven rather than function-driven decision making. For example, a material that is chosen due to its availability at the time of the original design, rather than for having particularly useful material properties. Finally, Rekoff also brings up the possibility that certain interfaces may have no immediate use but may be incorporated to allow for growth, upgrading, for adjustment and troubleshooting, or for in-assembly testing but serving no function in the final system.

Rekoff suggests the importance of being acquainted with the modus operandi or technical culture of the original engineer to include the following: design philosophy, manufacturing philosophy, maintenance philosophy, logistic support philosophy, and intended use of the system. He also underscores the importance of incorporating technical specialists as part of the reverse engineering team because they will be familiar with a range of solution-patterns that are not self-evident. As he puts it, "There is a vast amount of technical folklore that can significantly expedite the reverse engineering process" (Rekoff, 1985, 248)

When multiple specimens of the system being reverse engineered exist, Rekoff emphasizes the importance of keeping "a good one" (Rekoff, 1985, 247). He means one

that is in the original, operating condition. He also presents a formal documentation procedure that he believes performs two important functions: to guide and to record. His procedure employs three distinct types of documents:

1. Equipment breakdown Hierarchy—This document forces the reverse engineer to record the progress of disassembly in terms of system->subsystem->assembly->subassembly and so forth.

2. Configuration—This document forces the reverse engineer to track flows (such as information, energy, material,) and thus the functional connections among the parts of the system.

3. Performance specification—This document mirrors the equipment breakdown hierarchy but adds functional characteristics of each item.

Rekoff emphasizes that the method herein described is predicated upon the validity of the assumption that a piece of hardware can be characterized as having hierarchical structure, and therefore the documents described will be generated over time as the reverse engineer moves from consideration of the system, to the subsystem, to the assembly, and so forth. This approach foreshadows the model of reverse engineering that will be offered in Chapter V, but it also has several key differences.

7. The Problems and Gaps

The works discussed in the preceding sections exemplify the main points of view and definitions of reverse engineering found in the existing literature. A number of other papers about reverse engineering were surveyed, and they will occasionally be referred to throughout this work (Shelley, 1996; Gigerenzer, 1999; Arthur, 2009; Kelly, 2010 et al.). However, the majority of what is left can be fit into one or several of the themes already discussed. Thus, an overview of the situation suggests there are several problems.

The first problem is the dearth of existing academic work on the subject of reverse engineering in general and about hardware. For example, in an online search of a common bookstore for the subject “reverse engineering” twenty of the top twenty-three results referred to reverse engineering of software or integrated circuits. The remaining three books have all been reviewed in the preceding pages.

The second problem is in the existing relevant work, the various definitions that do exist tend to be narrow in scope or ambiguous—there appears to be no single integrative definition of reverse engineering. Consider the sampling of definitions in Table 1.

Table 1. Some definitions of reverse engineering

U.S. Supreme Court[15]	*A fair and honest means of starting with the known product and working backward to divine the process which aided in its development and manufacture*
***O'Brien* (2010, 3)**	*A technique in which a student learns how a particular piece of equipment works by breaking it down into its fundamental parts. If the analysis is successful, the student will understand the purpose of each individual element contained within the system structure.*
***Oxford English Dictionary* (Oxford year, p)**	*The reproduction of another manufacturer's product following detailed examination of its construction or composition*
***Wang* (2011, 1)**	*A process of measuring, analyzing, and testing to reconstruct the mirror image of an object or retrieve a past event. It is a technology of reinvention, a roadmap leading to reconstruction and reproduction. It is also the art of applied science for preservation of the design intent of the original part*
***Rekoff* (1985, 1)**	*The act of creating a set of specifications for a piece of hardware by someone other than the original designers, primarily based upon analyzing and dimensioning a specimen or collection of specimens*
***Chikofsky* (1990, 15)**	*The process of analyzing a subject system to: a) identify the system's components and their interrelationships and b) create representations of the system in another form or at a higher level of abstraction*
***Otto & Wood* (2000, 1)**	*A process that starts with a product in the marketplace and a vision to redesign it... it entails a prediction of what the product should do, followed by modeling, analysis, dissection and experimentation. Reverse engineering is followed by redesign.*
***Messler* (2013, 16)**	*The process for discovering the fundamental principles that underlie and enable a device, object, product, substance, material, structure, assembly, or system through the systematic analysis of its structure and, if possible, its function and operation.*
INCOSE (Haskins 2006)	*No Definition*

[15] Kewanee Oil Co. v. Bicron Corp. 416 U.S. 470, p. 476 (1974). The complete quote reads: "A trade secret law, however, does not offer protection against discovery by fair and honest means, such as by independent invention, accidental disclosure, or by so-called reverse engineering, that is, by starting with the known product and working backward to divine the process which aided in its development or manufacture."

The multiple definitions in Table 1 underscore the lack of consensus among technical authors. Ingle sees reverse engineering as an activity that focuses on componentry—the simpler the better—thus failing to note the possibility or relevance of applying it to whole systems. Wang focuses on the tools and methods of reverse engineering and makes little mention of the motivations behind it. The U.S. Supreme Court narrow definition of reverse engineering reasonably applies to the case under consideration—that of ascertaining the methods of production—and ignores others, such as understanding the structure, function, functional allocation, or purpose of a system. Otto and Wood conceive of reverse engineering as an activity subordinate to redesign, while O'Brien considers it to be primarily a form of engineering education. The definition in the Oxford English Dictionary (OED) hinges upon the distinction between the original engineer and the reverse engineer. Raja—whose version of reverse engineering centers on digitizing the geometries of physical objects in the form of CAD—suggests reverse engineering is an integral part of an iterative design process carried out by a single design team. Finally, Messler is sufficiently broad in what he envisions to be the scope of reverse engineering, yet his definition—and general treatment of the subject—lacks solidity.

A third problem is that there exists little or no literature that identifies or attempts to exploit links between reverse engineering and related subjects and practices. Accordingly, works in several fields considered by this author to be closely linked to reverse engineering were surveyed, and have influenced this work in varying degrees. The following paragraphs list some such supplementary subjects and works.

Heuristics—The subject of heuristics is vast and academically fruitful. A lot has been and continues to be written about the role of heuristics in different aspects of our lives. Certainly, not all of what falls under "heuristics" is equally relevant to reverse engineering.[16] Some authors are concerned with heuristics as a general phenomenon of human cognition—the question of heuristics is the question of "how do humans decide."

[16] As with "reverse engineering" there is also no general agreement on what constitutes a heuristic. Appendix E. summarizes the viewpoints on heuristics from the works that were surveyed as part of this dissertation.

Here there are two opposing camps: Some such as Gigerenzer (1999) emphasize heuristic decision making as a remarkably powerful tool, given hard problems and incomplete information. Others, Kahneman (2011) for example, emphasize heuristic decision making as a source of systematic errors that ought to be, but usually is not, supplanted by reason-based decision making. Still others, like Lenat (1981), are concerned with heuristics in this sense, because of their application to artificial intelligence or computer based search algorithms. Key authors highlight the central role of heuristics in systems engineering. B.V. Koen in *Definition of the Engineering Method* (1985) first planted the idea in this author's mind that heuristics pervade everything that engineers do. Following the "heuristics thread" inevitably leads to the famous work of G. Polya. In *How to Solve It* (1973), Polya presents heuristics as both accessible and powerful tools that can be taught and applied in all types of problem-solving. In *Heuristics for Solving Technical Problems* (2004), E. Sickafus shows a method for the discovery of new heuristics for design. Equally useful, he also sets a standard for taking a complex question, "What is design all about?" and breaking it down to its atomic components, in order to tackle the task of answering it. Finally, he employs a visual modeling approach that served as a persuasive example of why a visual approach is valuable.

Problem Solving—Problem-solving and heuristics are almost synonymous topics. At least, they are two aspects of the same subject. Figure 4 shows as one of the fundamental questions about problem-solving: what type of problem do you face? The works of Miyake (1986), Wankat (2012), and Lochhead (1987) all highlighted the importance and overlap of problem solving in engineering and reverse engineering particularly in the context of engineering education.

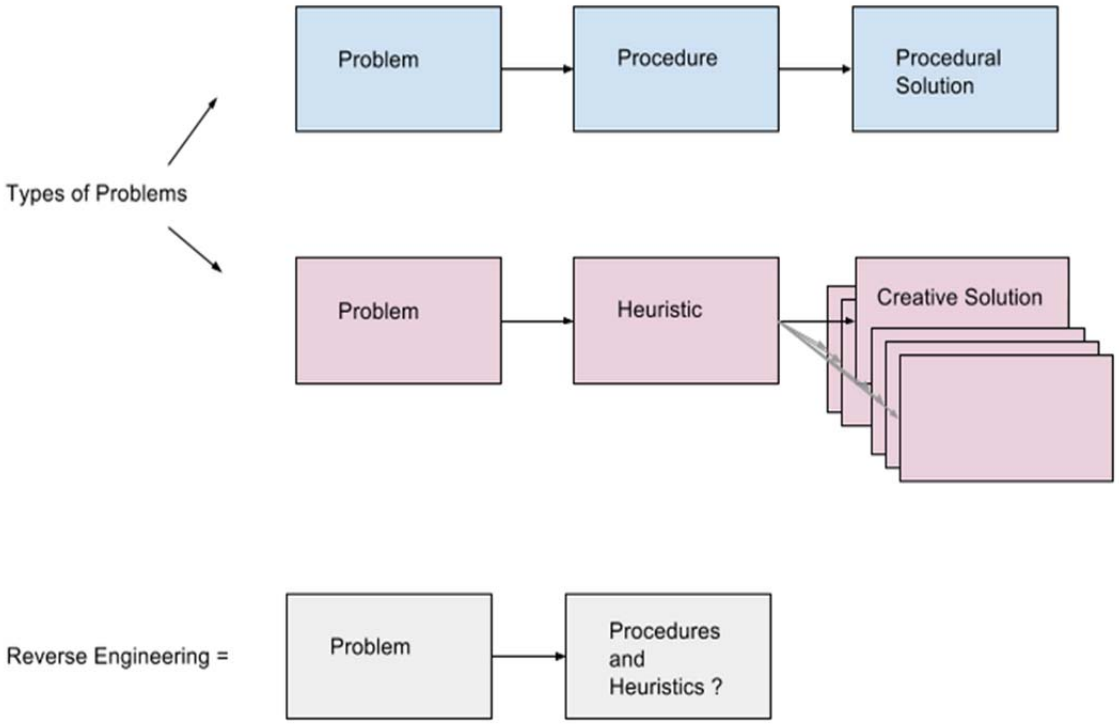

Figure 4. Reverse engineering problems—procedural or heuristic?

An algorithm tells how to solve a problem. A heuristic tells how to approach a problem. If you know the algorithm, you know how to solve the problem. If you know the heuristics, you do not know how to solve the problem... But you may yet solve it. An algorithm-solvable problem and its solution are two equivalent statements on opposite sides of an equal sign. A heuristic-solvable problem is separated from its solution by an irreversible event: discovery.

Philosophy of Engineering and Design—What is engineering? Is it "applied science"? How is engineering different from science? What is essentially different between engineered systems and a natural ones, or between an engineered systems and art, or perhaps between sufficiently advanced engineered systems and magic? What is it that engineers know, and how did they learn it? These and other "philosophical" questions seem to be inextricably linked to reverse engineering—an activity that centers on the systematic acquisition of engineering knowledge from artifacts that are the outcome of engineering knowledge and activity. This dissertation incorporates insights from the two books written on these subjects: *What Engineers Know and How the Know It* (1993) by W. Vincenti is a lucid, example-based exposition of the types of knowledge that constitute engineering knowledge. Reading Vincenti prompted me to ask, which of

these different types of engineering knowledge are actually "encoded" into the built technological systems, such that they can be subsequently "decoded" through reverse engineering? *Thinking Through Technology* by Mitcham (1994) provided numerous insights into the nature of technological systems and the way that we interact with them. A number of the more philosophical ideas that have influenced this dissertation also come from a "classic" work about design, purpose, and functionality: *The Design of Everyday Things* (2002), by D. Norman. There are several other smaller philosophical works on the niche of philosophy of technology—these tend to be concerned with the "dual nature" (physical/structural and intentional/functional) of technological artifacts and how the engineer and designer moves from one nature to the other (Enrong 2013; Vaesen 2011). In particular, reverse engineering seemed to be tied to the problem of what constitutes a good explanation. It seems that a reverse engineer may be successful to the extent that she can fully explain the technological system under consideration. Accordingly, Hempel and Oppenheim "Studies in the Logic of Explanation" (1948)—considered the seminal work on what constitutes a scientific explanation—was reviewed. Another branch of philosophy that bears upon the subject is teleology, the study of purpose or the particular approach that we may take toward a thing when we know its purpose, or we suspect the presence of a purpose. For example, chemistry and physics are not generally teleological sciences, while biology and psychology tend to be. It may be said that sciences concerned with products of systems engineering are teleological. For example, of every feature discovered in an organism, the biologist tends to ask "and what is this for?" Biology and reverse engineering share this critical question. It gives the researcher access to teleological language (to speak about purpose) and normative language (to speak about fitness to that purpose). For example, an engineer may speak of a good chair, a bad pen, or a broken car… a biologist may refer to the parallel concepts of a healthy, sick, or dead organism. This points to a peculiarity at the heart of our question: reverse engineering seems to reside somewhere between science and engineering. Reverse engineering seeks to explain an object, by treating it as if design is inherent in it. (Simon 1996). Reverse engineering is in a sense a science of engineered things.

The Evolution of Technology—Another related topic is the question of the evolution of technology. How does it evolve? Where is technology going? What is the history of this evolution? What are its limits? To begin with, what is the nature of technology? *The Nature of Technology* (Arthur, 2009) offers fundamental answers. The works of Kelly (2010), Basalla (1988), Rosen (2010), and others offer supplementary views. A coherent picture emerges from reviewing these. The picture is consistent with the following proposition: reverse engineering plays a critical role in the evolution of technology. On a different vein, works such as Clarke's and Kurzweil's that are concerned with distant future and limits of technology also informed this dissertation. (Clarke, 2000; Kurzweil, 2006) These authors tend to use "reverse engineering" in a unique context. They envision future artificial systems based upon design principles gleaned from the study of living systems. An artificial neural network, for example, employs design principles that have been borrowed directly from our understanding of brains, thus the neural network is the outcome of a form of reverse engineering. But futurist authors like Clarke and Kurzweil go beyond mere biologically inspired systems and extrapolate the power of reverse engineering until it becomes a means to recreate life, intelligence, and consciousness. Although more often encountered in non-academic work, this is an intriguing usage of "reverse engineering."

Archeology—As mentioned in the overview of Messler's book, there is a synergy between reverse engineering and some problems in archeology (Messler, 2013). The clearest example of this can be found in the history of the collective effort aimed at the gradual decoding of an archeological artifact known as the Antikythera Mechanism. The best popular account is J. Marchant's *Decoding the Heavens* (2010). Also relevant is the work of C. Shelley, an academic investigation into the mental process through which archaeologists apprehend the presence of human made artifacts (Shelley, 1996).

Hacking and Making—Amateur engineering may be as old as engineering itself. However, under the names of "hacking" and the more ambitious sounding "maker movement" the phenomenon has come into vogue in recent years. Makers and hackers are important from the point of view of reverse engineering for two reasons: (1) reverse engineering is often a part of what they do, (2) hacker and maker originated technologies

proliferate “in the wild,” outside the conventions of industry like version control and documentation. The following scenario is becoming more likely every day: We found this machine, it seems to do *X*, but we do not know how it does it—queue the reverse engineer. Accordingly there were several books to review in this field. These included Anderson’s *Makers, the New Industrial Revolution* (2012), *Hardware Hacking* (2004) by J. Grand, and *Invent to Learn* (2013) by Martinez.

Reverse Engineering as a Cultural, Political or Economics Subject—There is a category of literature that considers reverse engineering not as a method but as a social phenomenon within the larger context of technological progress and society. This topic overlaps with the topic of reverse engineering as a mechanism for the evolution of technology. Authors in this group tend to extol the virtues of imitation as a catalyst of technological progress that can be unleashed by reverse engineering and which can propel late starters forward beyond the cutting edge (Kim, 1997; Shenkar, 2010; Niosi, 2012; Zhou, 2006). Similarly, others suggest that the constant presence of reverse engineering provides a kind of pressure that drives healthy competition and prevents stagnation (Wang, 2011). The most dramatic version of this literature focuses on the government sponsored program of nation-wide, multi-industry reverse engineering that transformed South Korea from a “non-player” into a nation on the technological forefront in the span of one generation (Kim, 1997). There exists a considerable amount of literature of this type, but it was not exhaustively reviewed here as it does not bear directly upon this work’s hypothesis. However, a partial review of literature in this category serves to validate the work.

Other Related Practical Endeavors—There is also a potential for productive intellectual cross-pollination between reverse engineering and some activities that are not commonly thought of legitimate fields of academic inquiry. For example, J. Slocum’s work on puzzles offered a categorization of physical puzzles (2001), which led to one of the basic goals in this work: obtain a clear categorization of the types of problems encountered in reverse engineering. Also, there is a strong intuition that the challenges encountered by the reverse engineer in his attempts to extract meaning from the thing before him, as well in as in the effort to disassemble without destroying valuable

information, are mirrored by challenges found in puzzle solving. A less reputable endeavor that is nevertheless related to both puzzle solving and reverse engineering is that of defeating physical security systems. Attacks on security systems are a special case of system tear-down; it is a tear-down of a system that has been designed not to repel that very tear-down. "Illegal Engineering" (1994) by Tim Hunkin[17] is a good introduction to this subject and brings up several interesting ideas. For example, Hunkin explains that safes and locks are not intended to be impervious to attack, but merely to delay the attacker such that other components of the overall security system (such as alarms, and the security forces responding to them) can do their function more effectively. Some design elements that are primary considerations in locks and safes are likely to be employed by designers of economically or strategically competitive technologies intent upon denying access to critical information contained within their systems to their competitors or enemies in the battlefield. *Safe Cracking for the Computer Scientist* (2004) by M. Blaze is the most cited work on the subject. Once again the relevance of this work and others like it lies in the fact that the reverse engineer may sometimes encounter counter-reverse-engineering elements in the design of the system under consideration. These defensive design elements are likely to borrow from security systems like locks and safes. A reverse engineer finding herself in such a situation will have to borrow from safe cracking and illegal engineering techniques. Smaller works such as Penev's *Design for Disassembly* (1996) were also reviewed—his work cites motivations for implementing a new design paradigm that overlap with this work's motivations for a better understanding of reverse engineering.

8. An Overview of What We Know

Thus, begins the search for heuristics in reverse engineering. As proposed in Chapter I, the first step in this quest was to review the relevant literature in order to arrive at an integrative definition. Such a definition is necessary as a prerequisite for productive thinking or talking about reverse engineering. The definition sought should describe a common ground and incorporate important elements of reverse engineering from the

[17] Hunkin's other popular work includes a TV Series titled The Secret Lives of Machines, as well as the design of several exhibits in San Francisco's Exploratorium.

existing technical literature without falling into contradiction or unhelpful vagueness. The following are offered as critical components of a new definition:

1. It is a process or practice related to traditional or forward engineering and design but distinguished from these by a reversal of starting point and end goal.

2. It is a subset of the larger set of activities called problem-solving. Reverse engineering is a problem domain

3. Its starting point is provided by a machine (or some other human-made physical technological system, to include a part of a larger physical system).18 This machine is the primary source of information, and the object of the reverse engineer's attention, but it is not the only object. Sources may also include user or maintenance manuals among other things.

4. Its means involve physical manipulations of the technological system—looking inside, examining components, dissecting, tearing down, opening, measuring and testing.

5. Its end goal is information. There are several types of information that are in some way implicit or encoded in a human-made physical technological system. For example:

    (a) intentional and contextual (i.e., what is the thing's purpose? Why was it built?)
    (b) functional (i.e., what does it actually do? What is it capable of doing?)
    (c) structural (i.e., what is its configuration? What and where are its parts?)
    (d) operational (i.e., what are the operational principles? How are the functions and structures allocated?)
    (e) manufacturing (i.e., how was the thing put together? What processes were used to achieve its essential characteristics?)
    (f) physical (i.e., what are the relevant measurable characteristics, including shape, size, weight, strength, hardness, and others?)
    (g) any or all of these may constitute the legitimate objective of the reverse engineer.[19]

[18] While starting-point must be an existing system, it need not be physically present. Some of the more notorious cases of reverse engineering have used photographs and other forms of information as proxy for the actual system. (Messler 2013).

[19] One author (Jenkins 1984) suggests that even other more subtle types of information can be imagined to exist embedded or encoded in the structure and materials of a system, such as the values prevalent during the time of the system's invention, or perhaps even the identity of the system's inventor or manufacturer.

6. It could be said that where ordinary or forward engineering aims to model to (i.e., a process that leads from model to physical system), reverse engineering strives to model from (i.e., a process that arrives at a model from the examination of the physical system).

A synthesis—Here is a proposed definition: Reverse engineering is the problem-solving activity that ensues when one takes a human-made system (whole or in part) and attempts—through systematic analysis of its physical[20] characteristics and other available evidence—to answer one or more of the following questions: What is this for? What does it do? How does it do it? What is inside it? How does it work? How was it made?

C. CONCLUSION

This chapter consisted of a survey of the major and most influential works about reverse engineering (but excluded—for reasons given earlier—most works about reverse engineering of software and integrated circuits). From this, some of the existing gaps in the academic treatment of reverse engineering were pointed out. That was followed by a survey of "other" literature covering an assortment of subjects on the basis of their potential (but not explicit) relevance to this work (i.e., works on subjects other than reverse engineering). Finally, a general definition of reverse engineering was proposed.

[20] In this dissertation the phrase *human-made system* refers to a physical artifact. Whether the outcome of this research can be generalized to encompass non-physical systems such as organizations is an open question.

THIS PAGE INTENTIONALLY LEFT BLANK

III. A PROTOTYPE OF REVERSE ENGINEERING

> It has been observed that an electrical engineer of the early twentieth century would be far more mystified by the modern transistor, than Imhotep himself would be by the Empire State Building.
>
> —Jean F. Brennan, *The Elegant Solution*

A. INTRODUCTION

From definition to practical considerations. Consider an activity like riding a bicycle. It is one thing to have a definition for "riding a bicycle." It is a very different thing to ride a bicycle. It is yet a third and distinct type of thing—a sort of bridge between the other two—to imagine what riding a bicycle may entail.

The goal of this chapter is to achieve a similar intermediary step. The objective is to methodically develop a sense of the preparations, decisions, challenges, actions and other practical considerations that may be involved in reverse engineering. In order to do this, the definition from Chapter II will be used as a framework or scaffolding on which a prototype of reverse engineering will be built.[21] This will be done by considering each part of the definition in turn. In the previous chapter we asked what is reverse engineering? In this chapter we ask given that definition, what does it look like, when someone **does** reverse engineering?

B. ANALYSIS

A human-made system, whole or in part. In the following pages, a hypothetical reverse engineer is faced with a technological system in the traditional nuts-bolts-and-wires sense. It could be a cell-phone, a laser printer, or an intercontinental ballistic missile.

[21] This scaffolding is not entirely imagined, as the author since undertaking this research has also engaged in numerous exercises of reverse engineering at home. These have included a printer, a vacuum cleaner, a cassette player, and a number of unfortunate toys. His own children have been the eager recipients of an education in technology, and countless magnets and shiny bits.

Since these systems' qualification as "human-made" is unambiguous, are there any practical implications raised by this part of the definition? There is one important consideration: drawing the boundary. Consider two examples.

First example: The target system is a remote-controlled television. Before we begin reverse engineering it, we know about the endeavor that it will soon lead to questions related to what does this thing do? We also may know—or suspect—that much of what a remote-controlled television in fact does will be more readily apparent if we conduct our inquiry by operating the television as it was designed to be operated: by using the remote control as the primary interface for operation. In other words: if we choose to draw the boundary of the target system such that it includes both television and remote control device, the task of discovering system-level functions is potentially simplified. In a variation of the same example, most of the functions of the television will remain undiscovered if there is no content-bearing signal. The reverse engineer therefore faces the real question of whether the target system boundary should be stretched to include the receiving antenna. Perhaps it should have the broadcasting equipment. Where does he draw the target system boundary line?

Second example: The reverse engineer's services are being used by a computer company to benchmark a competitor's laptop whose battery life is superior.[22] In this case, it may be desirable to draw the target system boundary around the battery only, excluding the rest of the competitor's computer. Later, if the battery is not found to hold significant innovations that can account for the superior performance of the computer, the target system boundary may be expanded to include other power-consuming componentry.

It is not implied by this line of reasoning that the reverse engineer must remain oblivious to the things he chooses to leave out when he defines the target system boundary. As will be seen later, these are still subject to the attention of the reverse engineer, as they belong to the target system's context. Nevertheless, the choice of boundary will guide the reverse engineer's attention and allocation of resources and thus it is the first important practical consideration. As the computer example shows,

[22] The term "benchmarking" is synonymous with reverse engineering in contexts where the process is employed to assess the capabilities of a system designed by a competitor (Little 1997).

externally imposed constraints such as limits in time, budget, or space, sometimes inform the reverse engineering project.

Thus, the first practical implication of doing reverse engineering is the need for a judgment: Where do I draw the boundary for my target system?

Systematic analysis. The word *systematic* suggests the use of a process—an approach that is methodical, organized and repeatable. The word *analysis* indicates that the enterprise hinges upon a detailed examination of the parts in relation to each other and the target system. What this systematic analysis actually entails will be the subject of the next several pages. However, one key element of reverse engineering is subtly suggested by the phrase "through systematic analysis." The implication is that the reverse engineer is not required to bring to the project other sources of a priori knowledge about the target system.

The goal of the reverse engineer is to arrive at certain types of engineering knowledge. The types of knowledge in question are not uncommon among engineers, whether they practice reverse engineering or not. But the immediate source of knowledge in reverse engineering is the human-made system. And the path the reverse engineer takes to reach this knowledge in a systematic analysis of physical evidence within and outside that human-made system. This choice of source and path is what distinguishes reverse engineering learning from all other forms of engineering learning. The reverse engineer's knowledge of the system proceeds neither from book, nor teacher, nor carefully constructed lab experiment. It proceeds from systematic analysis of the physical system, and the evidence around it.

That is not to say that system-specific expertise is not useful in a reverse engineering project. It almost certainly is. The distinction between general engineering knowledge and target system knowledge is important. It is not being suggested that the reverse engineer should strive to be a blank slate. On the contrary, a solid foundation of the basic engineering principles underlying the target system is desirable. A reverse engineer working on a flashlight must understand electricity. A reverse engineer working on a tire must understand the pertinent concepts from physics and materials science. A

solid base of general engineering knowledge is always advantageous. Potential problems arise when there is a priori target system knowledge, as it could form the basis for bias that can negatively impact the attainment of the reverse engineer's goals. This will be discussed in greater detail in Chapters V and VI.

On the other hand, it is possible—even desirable—to use reverse engineering in order to acquire or strengthen general engineering knowledge. Some educators advocate the use of reverse engineering to teach fundamental engineering principles. For example, Otto and Wood (2000) advocate throughout their book that reverse engineering is the best way to teach design. Others advocating the pedagogical value of reverse engineering include (Hess, 2000), (O'Brien, 2010), and (Halsmer, 2013).

In any case, the second practical implication of doing reverse engineering is an apparent absence of a need for expertise on the particular system. In other words, in preparation for a project I do not need to become an expert on the particular type of system I intend to reverse engineer. I do, however, need three things. First, I need the correct tools and instruments to enable disassembly, and to examine the physical system at the desired level of detail and sensitivity. Second, I need a solid base of general knowledge covering the engineering principles believed to be in operation within the target system. Third, I need a disposition toward systematic analysis—that is, a scientist-like approach toward the target system.

Physical characteristics and other available evidence. With what does the reverse engineer have to work? In answer to this question, there are two distinct types of factors to consider. These may be called nature-of-the-system factors and specimen-situation factors.

Nature-of-the-system factors arise from the structural or functional characteristics of the target system. What are the challenges associated with the technologies and design features of this target system? Nature-of-the-system factors involve questions of size, intricacy, complexity, security, and others.

Size is the first factor to consider. A very large system like a city power grid presents unique size-driven challenges. Similarly, a very small system like an integrated

circuit poses unique issues. Intricacy derives from the number of parts in a given volume as in the city power grid example. Tolerance questions arise from a combination of size and intricacy factors such as those in an integrated circuit. Complexity affects the behavior of the system, and derives from the number of interconnections among components and the non-linearity of the processes involved. Finally, security refers to features that have been incorporated into the design specifically to deter uninvited breaches of the system boundary, including the forays of reverse engineering. Other nature-of-the-system factors include the presence and number of irreversibly assembled components or the presence of system-destroying or self-consuming operational principles such as those found in bombs or other explosive or single-use systems.

Another nature-of-the-system factor that arises from circumstances that are unique to reverse engineering may be termed "divergence in technological advance-level." This refers to a quantification of the relationship between OEM and reverse engineer. Specifically, how do the two engineers compare as pertains their respective knowledge of the existence of scientific and engineering principles. For example, a reverse engineer and an OEM whose educations are similar and contemporary occupy roughly the same technological advance-level.[23] In such cases, it is likely that reverse engineering will uncover some innovations, but it is unlikely that these will be revolutionary. This is the "normal" case of reverse engineering. Most authors in the technical literature are concerned with normal reverse engineering. We may also call this: Case I.

On the other hand, the reverse engineer(s) may be more advanced than the OEM, even considerably so as in the case of the Antikythera mechanism. Here, the process will almost certainly not uncover any technological innovations valuable as such. But it may uncover historical or scientific facts valuable for other reasons. We may call this scenario: Case II reverse engineering. Finally, it is conceivable that a reverse engineer may find herself faced with a target system for which the OEM was technologically more advanced. (for example: a reverse engineer from a pre-technological tribe tackling an

[23] This is the case even if the engineers work in different fields, assuming similar education tracks, and the diffusion of new knowledge (if only the highlights) through the media and continuing education.

iPhone, or the hypothetical human, reverse engineering the hypothetical crashed alien spacecraft). This last scenario is likely to prove most difficult for the reverse engineer but is also most likely to yield revolutionary knowledge. This scenario may be called Case III reverse engineering. Consideration of Case III brings up, once again the possible relevance of Clarke's Third Law in reverse engineering. Figure 5 summarizes the previous discussion.

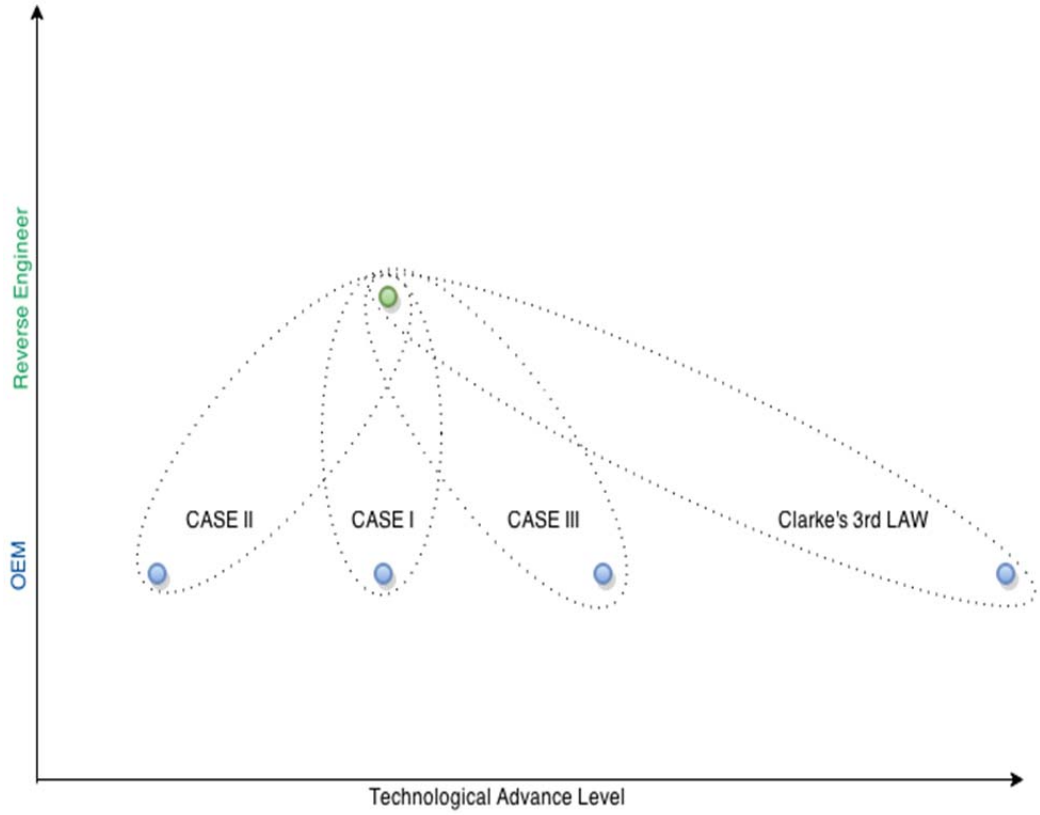

Figure 5. A classification: four cases of reverse engineering

The "distance" in the technological advance level between the original equipment manufacturer and the reverse engineer gives rise to a possible classification of reverse engineering. Case I in which both engineers exist in the same technological advance level (roughly corresponding to the same time-period) is the normal reverse engineering with which most of the technical literature is concerned. Case II in which the original engineer predates (or technologically lags) the reverse engineer is addressed in the literature about reverse engineering and archeology. Case III is not really addressed in the technical literature. Clarke's 3rd Law "*Any sufficiently advanced technology is indistinguishable from magic"* can be considered an extrapolation of Case III. While not addressed in the technical literature, this is a common theme in science fiction.

Specimen-situation factors arise from the particular conditions encountered, independent of the design features of the target system. These involve questions of availability and completeness of the specimen(s) and supporting information. Consider two different reverse engineers tackling the same target system, such as a certain model of scientific calculator. One of the prospective reverse engineers has at his disposal a single calculator, perhaps it happens to have a broken screen and no batteries. His counterpart is at the opposite end of the specimen-situation spectrum. He may have at his disposal a case of 100 brand new calculators, in their original packaging, batteries and user manual included. The higher number and better condition of the samples are likely to make the second reverse engineer's job much easier. The nature-of-the-system factors are constant across both cases. However, the specimen-situation factors are vastly different. Table 2 summarizes the information presented in this section.

Table 2. Reverse engineering factors

Factors	Situation	Implication
Nature of the Process[24]	- Reverse engineering vs. traditional engineering and design	- Need to open system - Need to detect functions and flows that are not explicit - Need to infer function from form
Nature of the System	- Circuit vs. engine - iPhone vs. airplane - Cessna vs. F-16	- Need for system-specific tools - Need for different baseline knowledge - Difference in scale and complexity
Specimen Situation	- One vs. many - Operational vs. broken - Complete vs. partial	- Different approach to tear-down - Different challenges in ascertaining function

All reverse engineering shares some factors, but there are also a number of variables that will distinguish one case from another. The second and third rows show some of these variables within reverse engineering. The first row shows some of the variables that distinguish reverse engineering from other forms of engineering

[24] This is not a third type of factor, but a review of the notion introduced in Chapter I that reverse engineering involves a distinct set of circumstances and implications. The inclusion of *nature of the process* here lets the table serve as a summary of all types unique circumstances arising from reverse engineering.

In summary, physical characteristics and other available evidence encapsulates a large number of variables that amount to a considerable spectrum in the difficulty-level between one reverse engineering project and another. The practical implication is the need to characterize the project in terms of all its relevant factors: I ask what kind of system is this? How big, intricate, complex, and secure is it? And how do these factors affect my goals and impact my budget. Do I think I have a grasp of the operational principles at work within the system? If not, I may need to research the subject. Also, how many specimens do I have? And what is their condition? Will I be able to keep at least one working specimen while I tear the others down? And do I have any other information that I may look into… perhaps packaging information, or user manuals?

What is this for? What is the purpose of the target system? These questions invite ambiguity. Namely, the difference between a purpose and a function is not always perfectly clear.[25] As used here, the difference can be stated as follows: a purpose can only be expressed in terms of a context. In contrast, a function of a system or component can be expressed without recourse to a context. Consider Component A, determined through reverse engineering to be designed to deliver a large amount of electric current for a short period of time. Next, we learn that Component B too, was built to deliver a large amount of electric current for a short period of time. It follows—from being successfully expressed without a context—that delivery of large amounts/short duration bursts of current is a function, not a purpose. Context includes the presence, interconnections and relative locations of neighboring systems. At the system-level, a human user is generally one of the neighboring systems. Interconnections can be physical such as wires, or inferred through cues such as the placement of a handle suggesting interconnection with an operator's hand. In this example, subsequent analysis of the context reveals the respective purposes: Component A was designed to deliver electricity to a starter motor while Component B was designed to deliver electricity to sluggish cattle.

[25] As mentioned in Chapter II, one author (Messler 2013, chap. 7) actually invites even more ambiguity when he attempts to talk about *Role, Purpose, Functionality, and Function* and fails to clarify the distinction between the terms.

While not typical, it is possible to encounter cases of reverse engineering where even system-level purpose is unknown. This was the case with the Antikythera mechanism. But normal reverse engineering usually begins somewhere after the question of purpose has been answered. Reverse engineers do not usually need to inquire about system-level purpose. When tasked to reverse engineer a car, a cell phone, or a machine-gun they already know the target system's purpose through experience, use, literature or TV).

However, as will be discussed later, reverse engineering is an iterative inquiry activity. The process will eventually lead the reverse engineer to ask of each internal component the same set of questions that he asked of the whole system. As the process drills down, it is less likely that first-hand experience and normal exposure will have equipped the reverse engineer with a priori knowledge of the purpose of all the internal components.

There is an important corollary: once the system is completely disassembled, the context for the components—even if meticulously recorded—will have been destroyed. This will adversely impact the ease with which one may ascertain component purpose. In other words, there is an implication that studying the context for the system-level inquiry, and preserving the system in operation in order to successfully pursue the eventual component-level inquiries are important elements of reverse engineering. This suggests that reverse engineering should follow a top-down approach.

The practical implications here are: I must look around the system in order to discover what it is for. What are the other systems that it interacts with? Some of these systems may be obvious, tangibly connected. Other interacting systems may only be suggested by cues. I must then turn the system over… look at it from different angles, at different distances. What is the nature of the interactions between the target system and these others? As I look ahead to the eventual tear-down of the target system, I must remember that this tear-down will affect my ability to determine the purpose of its internal components.

What does it do? As we move forward with asking the questions imposed by a hypothetical task of reverse engineering, note two things. First, the order of the questions is not arbitrary; they are intended to proceed from the general to the specific. This structure that affects the ease with which the answers may be attained. Second, the order of the questions does not preclude the possibility or potential inevitability of a later answer informing and modifying an earlier one.

So it is with system functions. Using the example from the previous section, it can be seen that it would more difficult to discern the purpose of both a starter motor solenoid and a cattle prod if one were unaware of the function that it delivers a burst of electricity to something.

A short theory of technology will be introduced at this point to address the ambiguity in the concept of function. The theory is borrowed and adapted from a slightly different context of technological innovation (Sickafus, 2004). According to Sickafus, there are three important concepts: objects, attributes, and effects. Objects are the bearers of attributes. He describes how objects also interact with each other through attributes. Any such interaction will result in an effect. As Sickafus notes an effect is nothing more than the preservation or alteration of some attribute. Figure 6 shows a graphical representation of an Object-Attribute-Effect relationship where two objects interact with each other. The interaction is not direct but mediated through attributes. Attribute 1.1 is an attribute of Object 1 and Attribute 2.1 is an attribute of Object 2. The contact between attributes results in an effect. This can be summarized in two definitions.

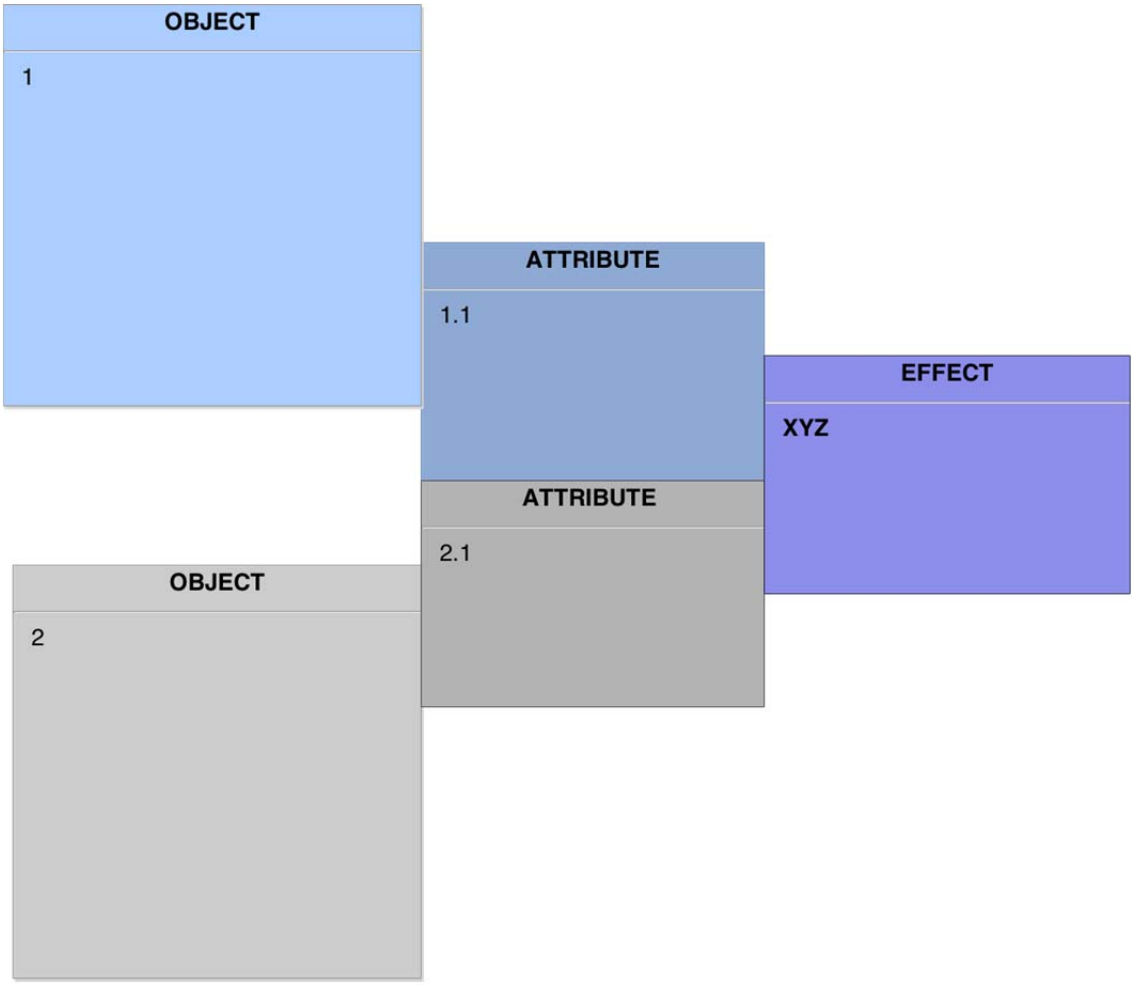

Figure 6. A generic object-attribute-effect diagram

Shown here: Objects interact through attributes in order to cause effects. When the effects are undesirable, they call for design solutions. When the effects are by design, they are functions. Adapted from: E. Sickafus, 2004, "Heuristics for Solving Technical Problems—Theory, Derivation, Application," *Ntelleck*. 33–34.

1. Design: The activity in which objects are arranged relative to each other so as to produce a desirable effect through their interactions.
2. Technological System: A particular arrangement of objects under the guidance of design.

Consider the following example: Object 1 is a battery—its relevant attribute is its voltage. Object 2 is a tungsten filament—its relevant attribute is incandescence (emission of light when heated by a current). In nature, the two objects may interact in some haphazard way of no practical use. But under the guidance of a design, Object 1 and Object 2 can be intentionally arranged in such a way as to alter a third attribute (the

lighting in a room) in a desirable way. This arrangement constitutes a technological system. Note that the arrangement in this case calls for an interface (a wire) that allows the two attributes to be brought into contact. This is important because interfaces often are explicit material parts residing on or near the surface of a system, discoverable through physical inspection. In other words, interfaces are one category of clue that the reverse engineer hopes to find.

To see this concept applied in reverse engineering, consider the following examples:

Imagine the target system is a high-end car tire. Such a tire is designed to carry out functions of traction (static and dynamic), shock absorption, and responsiveness under acceleration (linear and turning). In order to determine the level of performance of the functions, the reverse engineer must test the tire under operating conditions. If the performance of all functions is average, there will be little justification for investing further effort in reverse engineering. On the other hand, if the target system displays particularly good performance in some or all of the known functions or perhaps in some new function, like fuel efficiency, then reverse engineering will be justified.

Thus, in the case of systems with well-known functions the task shifts from finding system functions, to identifying those functions where some innovation or advantage may reside. In either case, whether finding functions or identifying high performing functions, the reverse engineer turns to operational testing.

In order to discover or measure all the system functions, the testing must cover all the use cases that were part of the system's design. These use cases consist of permutations of operational modes. In the case of the tire, it includes braking, accelerating, turning, and going over obstacles in operational conditions. Tests will involve dry and wet pavement, gravel or snow-covered roads, climbing, descending, and so forth.

Consider one more example to highlight the potential usefulness of a having experience as a user/operator. In this scenario the target system is new ink-jet printer. A user experienced with earlier similar printers may know that the ink normally requires a

few seconds to dry after the paper is ejected. Suppose the target system produces dry-ink prints. The function *dries the ink* is evidently an innovation and an important function from the point of view of reverse engineering. But such a function is immediately apparent only to a reverse engineer who is already familiar with earlier similar systems.

In other words, the practical implications of asking what does it do—that is, of finding or evaluating system functions are these: I must discover the modes of operation and the operating conditions the target system has been designed for, and I should operate the system in all (or at least an adequately representative number of) the use scenarios. If the system is broken, incomplete, or otherwise not operational, I will have to resort to cues and inferences to first restore or model it. In each use scenario, my goal is to identify the desirable effects that result from the deliberate arrangement of objects (and their attributes) that make up the system's design. Systems on which the operational scenario involves the destruction of the system, present a special challenge. A familiarity with the target system from a user perspective will be valuable.

How does it do it? This is a question of allocation. That is, given a function, what is the physical agent that enables it? Here it is useful to go back to the earlier discussion of Objects, Attributes and Effects. In the previous section we described the reverse engineer's search for effects-by-design, also known as functions. In this section the search turns to objects with attributes-by-design, also known as interfaces. In a certain sense this is a search for physical locations: where does such and such function reside?

In spite of being objects, interfaces are not necessarily solid or even visible from outside the system boundary. For example, a system-level function for a modern cellphone is wireless data transmission. Usually, the interface for this function is an antenna that is completely internal so is visually and materially inaccessible without breaching, in spite of the fact that the function extends beyond the system boundary. Perhaps it is more accurate to say that in this case, the interface is the wireless signal itself with attributes-by-design like frequency, magnitude, modulation, and so forth. While we do not see the signal in the conventional sense, it is nevertheless "observable" in the scientific sense of the word, if we have the necessary instrumentation and know to use it.

Recall that when a target system performs a certain function within average parameters and through well-understood mechanisms, this function will probably not draw any further attention for reverse engineering. To continue with the example from the previous section, suppose the reverse engineer of a high performing tire, discovers during testing several functions worthy of further inquiry. These could include superior shock absorption and excellent traction in gravel. Next, the reverse engineer drills down toward one function in particular and asks: what object/attributes (interface) in the tire interact with the pavement to yield the desirable effect (function). For example, the function *shock absorption* can be linked to the attribute *radial bulk modulus*. A reverse engineer searching for the object where this attribute resides may explore the tire-wall height, material, and air pressure as possible candidates. All of these contribute to shock absorption, but perhaps it is an unusually low tire pressure, or a new tire-wall material that holds the secret to this particular tire's superior shock absorption. A similar situation is shown in Figure 7 where the object of reverse engineering attention is the tire's traction, but the interface responsible is currently unknown.

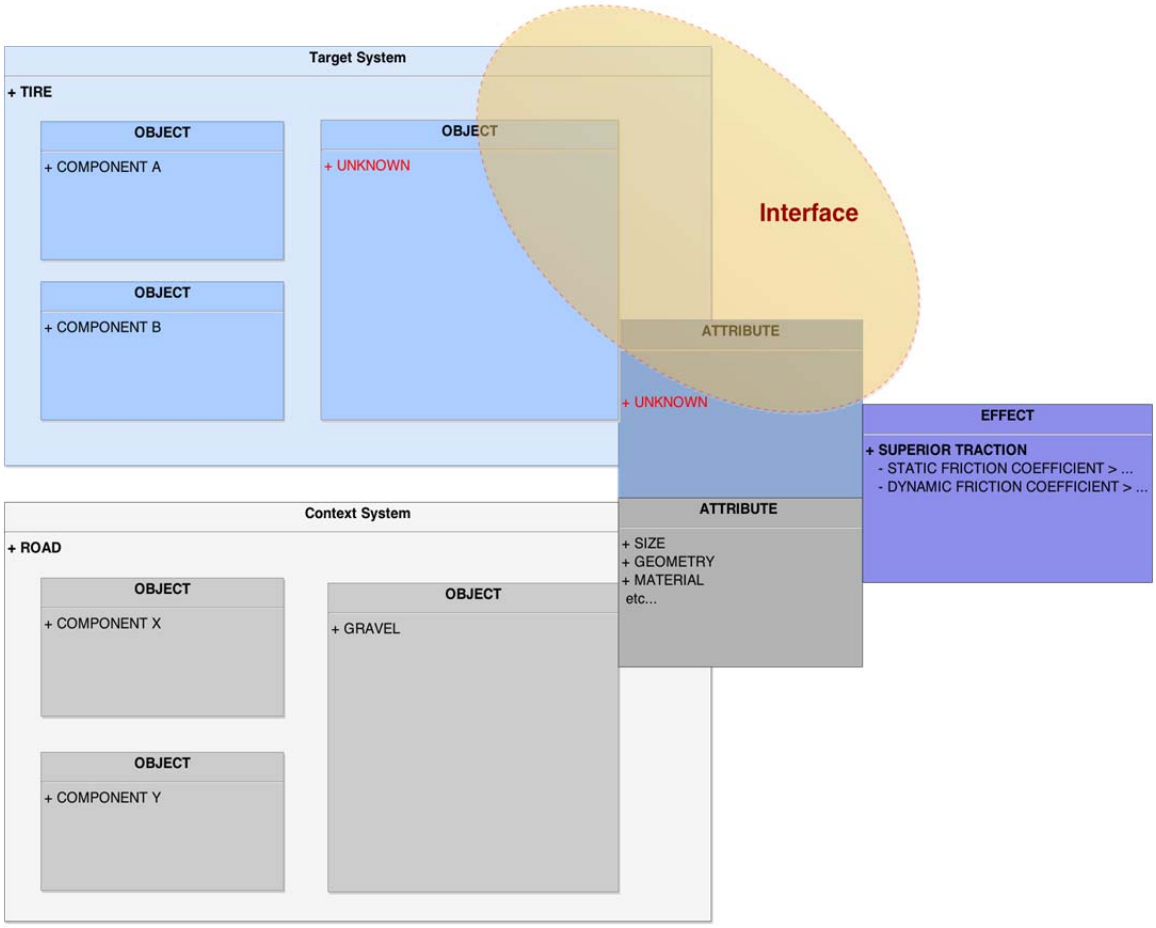

Figure 7. An object-attribute-effect diagram

The figure shows a function of interest (superior traction) that has been revealed through testing. The function results from the by-design interaction between a context object with known attributes (gravel within certain parameters) and an unknown target system object and with unknown attributes. The reverse engineer seeks to identify the unknowns (shown in red). Adapted from: E. Sickafus, 2004, "Heuristics for Solving Technical Problems—Theory, Derivation, Application," *Ntelleck*. 33–34.

The tire is a realistic, but also relatively simple target system. Additional challenges will arise when the target system size, intricacy, complexity or security are increased. For example, complex system behaviors—which can be extremely sensitive to initial conditions—might be very difficult to trace to explicit physical points of origin. Similarly, behaviors that arise from multiple, nonlinear interactions may be difficult to untangle. Another problem arises from the fact that not all attributes (and not all physical

objects) are evident under "normal" examination. An electromagnetic interaction between two components is essentially undetectable, unless the reverse engineer possesses the necessary instrumentation and knows to use it.

Finally, there is a flip side to the "function flows from attribute" assumption. There are often attributes or apparent interfaces that do not serve a function—or that serve a weak subordinate function, in spite of their physical prominence. For example, consider the tailfins often found in cars designed in the 1950s and 60s. These seem to suggest a function related to aerodynamics or stability perhaps because they are suggestive of tails and fins in biological systems. But this is an error.[26] Resources invested by a hypothetical reverse engineer into ascertaining the aerodynamic attributes of the rear-end of a 1950s Cadillac will be inefficiently allocated: the tailfins serve no function.

Thus, the practical implications of asking how does it do it—that is, of allocating all relevant system functions to interfaces made up of physical objects and their attributes are these: Once I am satisfied that I have a complete list of the desirable effects that result from the deliberate arrangement of objects that make up the system (i.e., functions), I turn my attention to these objects and their attributes. That is, each function I have identified should now be traced to a physical interface in the target system that appears to be responsible for carrying out the function. Note that the allocation of function to interface is not likely to be deep, early in the process, as objects and attributes need only be observable and describable from the exterior of the system under consideration. For example, if my target system is a computer, and the function is "display of visual information" the answer to how does it do it? need only be something like "Through a rectangular screen (Object) of such and such color, resolution, and brightness (Attributes)." All functions proceed from objects with attributes. However, I must be aware that the converse is not always true. Not all objects with attributes are there to support a function. Objects that are hidden and yet have system level interactions (for example internal weights in a system with a "balance" or "heft" function, or internal

[26] Admittedly this is a contentious claim. See Appendix F for further discussion of non-functional physical attributes.

antenna in a system with a "communicate wirelessly" function) and attributes that are undetectable except to special instruments (such as radio or infrared emissions, ultrasonic signals, or capacitive reactance) present special challenges.

What is inside it? The actions impelled by this question distinguish reverse engineering from other engineering activities except troubleshooting and maintenance. Engineering tends to be a "constructive" activity—in that it normally moves from ideas to realized physical objects (Mitcham 1994). Reverse engineering is "deconstructive." It begins with the physical object, and then it proceeds through analysis and manipulation of the object toward knowledge. In this process, there comes a point where important questions remain, but no further knowledge can be gained from the system as-is. It is time to take things apart.

Occasionally, a physical system has no hidden or internal components—for example a bicycle, or a parachute. However, most systems are protected from debris and other unwanted intrusions to some extent by a shell or casing that also limits direct visual and material access to the system's inner workings. The protective casing, itself a component, is also a physical boundary that segregates the system from the environment.

In the special case of physical security systems, the boundary with its protective and segregating functions takes on a central role. In many cases, some part of the protective shell or casing also doubles as a chassis or housing with the additional function of providing structural support to the inner components to enable the necessary interactions and functions. Often, the physical boundary plays more than a merely protective and structural role. In these cases, the boundary is integral to the system operation in some deeper way. For example, the block in an engine the glass bulb in a light bulb or a vacuum tube, provide airtight physical boundaries that make possible the internal chemical processes that define those systems. On other occasions the system boundary is not functionally essential but is included in the design for the protection of the operators and the environment from some damaging byproduct of the system operation. The shield encasing a nuclear power plant is a good example of this. Finally, the system boundary may serve none of these functions but be included in the design merely for its aesthetic qualities.

Tear-down is a unique aspect of reverse engineering. The main characteristic of tear-down is the physical breaching of the system boundary[27] followed by disassembly. Some important considerations and special problems that may arise during tear-down can be inferred from thinking about different types of systems in light of the discussion in preceding sections of this work.

The section on system purpose highlighted the importance of preserving the system in good working order, as normal operation constitutes the context for and helps make explicit the purpose of the internal components. This imperative for preservation of function may clash with the need to breach and tear-down the system. The following scenarios present special challenges:

1. A light bulb—It is a system where the boundary is functionally essential. Even if the bulb can be cleanly separated from the cap, the system cannot be opened without destroying its operation except in a special vacuum chamber.
2. A circuit board covered in epoxy—It is a system where the boundary, while not functionally essential, is assembled in such a way that opening it is difficult and disassembly without disturbing the internal arrangement. Tearing it down requires special methods and tools.
3. A 3D printed component with 3-D printed internal subcomponents—It is a system that has been assembled through an additive method of fabrication may present significant challenges because the process allows for manufacture of internal components and boundaries without any seam that may be exploited as a point of entry.

Problematic tear-downs may be alleviated by a specimen situation that allows for the destruction of some specimens and the preservation of others in operational condition. However, when the specimen situation does not allow for it, other courses of action must be considered such as the use of methods, x-rays or ultrasound, that permit interior examination without breaching.

Another problem arises when the system boundary is essential for the safety of the operator as in the case of antipersonnel mines and nuclear reactors. Here the tear-down takes on a particularly risky aspect.

[27] A special case of reverse engineering arises when the target system is a physical security system. In these special cases the tear-down is referred to as "system attack" (Blaze 2004, 17-33).

Finally, the question of breaching the system boundary carries with it an implicit evaluation of the physical actions taken. The sequence of steps ought to be effective and efficient. Effectiveness refers to the extent to which the actions taken do in fact lead to the desired end state: exposure of all internal components and maximum preservation of functionality and interconnections. Efficiency refers to the extent to which the time and resources consumed by the tear-down are minimized. For example, it is possible that the reverse engineer will invest undue time and effort carefully removing old fasteners, where the system could be breached without destruction by prying it open. It is also possible that the actions undertaken to breach the system will result in a significant amount of collateral damage to internal components and possibly cause the irretrievable loss of all system function.

An attentive reader may have noticed that while purportedly discussing the question of what is inside, thus far only the matter of how we get inside has been addressed. This corresponds with the author's belief that the challenges of breaching and subsequent tear-down of the system boundary constitute the most unique aspect of reverse engineering. The scenarios and questions just brought up are less covered anywhere else and are interesting.

Still, this part of the definition requires a closer look: what is inside? The question carries one important implication that has not been mentioned so far: the need for partitioning. For example, suppose an automobile is the target system. One answer to the question of what is inside a car could be as follows: engine, drive train, tires, steering system, framework, and driver. A very different answer could be given in the form of an inventory of parts, disassembled to the maximum extent such as a list itemizing 836 bolts, 403 nuts, 10.5 meters of copper cable. Both answers are true, and each may have an important and distinct role. But from the point of view of reverse engineering, one answer may be more useful than the other. Why?

As mentioned earlier, reverse engineering is a top-down process. This means that the tear-down should not proceed faster than the full functional characterization of the subsystems, components, subcomponents and so forth is mapped. If the question of what is inside is answered with a full parts inventory, this suggests a disassembly that

proceeded ahead of the characterization. The contexts necessary to reach full understanding of the various levels of system were probably destroyed. Therefore, the first answer is the better one. One additional consideration in partitioning is given by the partitioning heuristic that guides the reverse engineer to choose the subcomponents so that they are as independent as possible. That is, the components should exhibit low external complexity and high internal complexity (Maier & Rechtin, 2000, 49).

In summary, the primary practical implication of asking what is inside is that one must first determine how to get inside without wasting resources or unnecessarily losing information. In turn, the ultimate answer will depend upon a number of ancillary questions: Is there a boundary to breach? If so, is the boundary breachable without destroying system functionality? If it is not, does the sample situation permit going forward with an effective tear-down? If not, are there alternatives to tear-down… ways to look inside the system without destroying it? How can the reverse engineer avoid destroying internal components or disrupting functionality? Is the boundary in place to protect the environment? If so, how can the system-sans-boundary be put in a safe condition relative to the reverse engineer and the environment? The second part of what is inside calls forth the need to judiciously partition and label what one finds upon breaching the system boundary.

How does it work? This question may have somewhat different meanings depending on the scenario. For example, if the target system is a very compact automobile, the reverse engineer is likely to be most interested in those objects and attributes that contribute to the vehicle's compactness. In this case the question "how does it work?" may be equivalent to asking: "what clever arrangement of internal components makes such compactness possible?" Perhaps the chassis doubles as a tank, and the seats double as battery compartments. In other words, "how does it work?" may simply stand for "identify the components and their locations."

Sometimes the question requires the reverse engineer to go beyond merely identifying components into tracing flows of matter, energy, or information. For example, the target system may be a computer that operates without overheating in spite of having no fan. In this case "how does it work" will result in a search for an alternative

mechanism of heat dissipation. With the system breached, the reverse engineer will be able to trace the flow of energy (heat)—perhaps with the aid of thermal imaging—in order to answer the question. Perhaps a new material is used as heat sink, an innovative fractal geometry is used in the radiator vanes, or some design feature has been incorporated that enhances natural air circulation.

At other times, the question may refer to a principle of operation that is understood or within grasp but is not yet known. For example, the target system may be a mechanical calculator with gears, levers and cams as its components. None of these are individually beyond the grasp of a savvy engineer, who may even have a working model for how such a system may be constructed. Yet only exposure of the mechanism itself to visual inspection will reveal the actual principle of operation. In a different example, the target system could be something like a stud finder, a device that employs variations in the capacitance of a wall in order to find "studs" hidden under the surface. Here, the reverse engineer could be familiar with capacitance and yet have no idea that this is the operational principle that enables the stud-finder operation. In this case, the exposure of the system's internal parts is not guaranteed to yield knowledge of the operational principle, but it will likely provide cues, such as the discovery of capacitor plates, or perhaps even a revealing inscription referencing μF (the units of capacitance).

Finally, it is also possible that a reverse engineer will be confronted with a target system for which he simply possesses no knowledge of the operational principle. This is more likely to occur in the event of a Case III reverse engineering (see Figure 5 and the related discussion under physical characteristics and other available evidence). For example, a reverse engineer familiar only with vacuum tube technology could be faced with the task of reverse engineering a transistor radio. Similar to a biologist examining an organism, the reverse engineer examining a system that has been laid open about which operational principle he has no knowledge will require his speculation and experimentation. In other words, he will resort to the scientific method. In this scenario, reverse engineering becomes akin to experimental science, in particular to biology.

Regardless of the scenario, the system breach and tear-down play a critical role in the determination of how does it work? A successful system boundary breach can be

defined as follows: A) The internal components are laid out in the "open" for examination, and B) All (or at least a majority) of the system functions have been preserved. This is an ideal situation. A breached-yet-functioning system will allow access to essential observations and measurements either visually, or with the aid of appropriate instruments like multi-meters and oscilloscopes. The ability to "see" the components, interconnections, and flows while the system is under operation is likely to yield previously inaccessible information that will help answer the question of how the system works.

In summary, the practical implications of asking how does it work are these: What type of knowledge am I after: Configuration, or operational principle?[28] Configuration is straightforward: what is where. If I am after an operational principle, what is the expected nature of it? Is it likely to become physically explicit upon opening the system? Or might it involve an innovative application of some basic principle I already understand? Or finally, does this system look like it employs operational principles that are currently beyond my experience? Here again there is a practical consideration of safety. Even in cases where the system boundary was not there to shield the operator (and now the reverse engineer) from harmful radiation or toxic byproducts, it is still probable that along with the casing, I have also removed several safety functions (especially in the presence of electricity or moving parts).

How was it made? This is a question of processes and methods of manufacture. The most common reason for undertaking reverse engineering is the eventual reproduction of the target system. This reproduction may call for identifying and replicating the specific treatments used to achieve the necessary material properties of a part. These could involve unusual or unknown processes that enable unique system characteristics to be built (characteristics like miniaturization, or extreme size, or hard to achieve tolerance). Finally, "how was it made" could also refer to non-trivial assembly procedures where, for example, the sequence of assembly steps may be critical to the proper operation of the system.

---

[28] Both *Standard Configuration* and *Operational Principles* are basic types of engineering knowledge. (Vincenti 1993).

While clearly a part of a complete definition of reverse engineering, the question of "how was it made" is also quite distinct from the rest of the preceding discussion. Up to this point reverse engineering has been concerned with questions about the target system. In contrast, "how was it made" is a question that looks at the manufacturing system. A system that is related but separate from the target system.

Additionally, the question of how a system was made is largely a question for the materials scientist. While reverse engineering may routinely employ materials science, the latter is a well-seasoned science dealing with chemistry and microstructure, and distinct from the reverse engineering process that is largely concerned with functions and operational principles.

For this reason, the questions of how the system was made and what it was made of will not be incorporated into subsequent sections of this work.

Repeat as necessary. The questions: What is the system for? What does the system do? How does the system do it? What is inside the system? How does the system work? This is only the first round of questions in an iterative process. Following the breach and tear-down of the system boundary, the reverse engineer is presented with a new landscape of objects. Once these objects are arranged through judicious partitioning into suitable subsystems, each subsystem will in turn be subjected to the same questions. As the process moves inward, it is likely that examination will require specialization and a multidisciplinary team.

C. CONCLUSION

1. Practical Implications

In preparation for a reverse engineering project, one needs to bring three things: the right tools, good general engineering knowledge, and a scientist-like approach toward the target system. The first challenge is to determine what one is facing. How big, intricate, complex, and secure is the target system? Does the examiner understand the probable operational principle and if not is the reverse engineer familiar with it? What is the specimen situation: how many are there and in what condition are they? Will the engineer be able to keep at least one working specimen? Is there ancillary information?

Now work begins. First, examine the system in order to discover what it is for. With what other systems does it interact? Next, turn the system over (or crawl under and around it) to look at it from different perspectives. Operate it in as many user scenarios as possible. Systems that self-destruct during normal operation present a special challenge. If broken, incomplete, or otherwise not operational, the examiner will have to resort to cues and inferences to restore or model the missing parts of the system. The goal is to identify all system-level functions. After doing that, trace each function to an interface on the boundary of the system. Interfaces that are hidden beneath the surface and yet have system level interactions and attributes that are undetectable except to special instruments present special challenges. Next, determine how to get inside. The goal is to breach the boundary and begin tear-down without destroying any components or system functions in the process and without injury. Make some judgment calls on possible trade-offs. Once inside, judiciously partition items found as part of the effort to figure out how it works. After this partitioning, consider each module in turn, subjecting it to the same battery of questions.

2. Chapter Summary

Based on the discussion presented on this chapter, the following definition is offered: reverse engineering is the activity that takes a human-made system (whole or in part) and attempts—through analysis of its physical characteristics and other available evidence—to provide answers to one or more of the following questions: What is this for? What does it do? How does it do it? What is inside it? How does it work? How was it made?

The definition—like all definitions—addresses the question of what the thing is. The goal of this chapter was to illuminate at least some preliminary considerations on the matter of how these questions are answered.

The source material for this chapter was the synthesis of examples and definitions from the literature review, supplemented by the author's first-hand experience attempting to undertake reverse engineering projects.

IV. AN IMPROVED SYSTEM MODEL FOR TARGET SYSTEMS

> A wagon with spoked wheels carries not only grain or freight from place to place; it carries the brilliant idea of a wagon with spoked wheels from mind to mind.
>
> —Daniel Dennett[29]

A. INTRODUCTION

The previous three chapters have built upon each other to provide a progressively more useful characterization of reverse engineering. The result is an improved definition and a clearer grasp of what reverse engineering entails. This includes a collection of practical considerations. However, the ideas accumulated thus far cannot yet be used to explore the implications of the process, to base the analysis of specific reverse engineering projects, to use as a guide for steering the reverse engineer's efforts, or to predict any particular outcome from reverse engineering. For this we will need *a model of the reverse engineering process*. Such a model will be the end goal of the next chapter. In this chapter, the justification and building blocks for a particular type of model will be established.

B. METHODOLOGY

Heuristics work well when the structure of the technique matches the structure of the problem domain to which it is being applied (Gigerenzer, 1999, 25). In the problem domain of reverse engineering there are two sources of structure. First, there is structure inherent in the target systems. That is, there is some sense in which physical technological systems tend to be internally organized (physically and functionally) in a consistent way (Arthur, 2009; Basalla, 1988; Kelly, 2010). Second, there is a structure inherent in the process. That is, there is some sense in which reverse engineering tends to involve certain steps that follow a consistent order.

Structural information is information that can be conveyed through statements of the following types: Statements conveying physical structure: A is connected to B; A and

[29] Quoted by Gleick (2011, 3).

B are parts of C; A is inside C; C surrounds A; B is between A and D. Statements conveying temporal structure: A follows B; B precedes A; A and C are simultaneous. Statements conveying causal structure: A is contingent upon B (B is either merely necessary, or necessary AND sufficient); B causes C; C and D are independent.

If our objective is to analyze the structure of systems and processes, we might attempt to do this through nouns, verbs and adjectives. Words are how we think. However, we can also think in pictures. The latter is a better way to think about structure (Ferguson, 1994; Tufte, 1997). As humans, we have evolved an ability to process some types of information visually much more efficiently than by reading it or hearing it. Visualization is an approach to the analysis of structural information that plays to this cognitive strength. For example, it takes only a fraction of a second looking at Figure 8, to correctly evaluate the presence of all the physical structure relationships listed at the start of this paragraph.

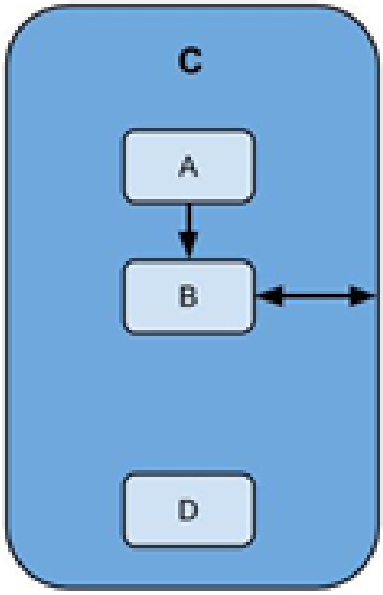

Figure 8. A simple and generic visual model

Several elements of the diagram (such as relative sizes, locations, and connections of A, B, C, and D) are apprehended almost instantly. This is known as preattentive processing.

In fact, in a barely perceptible amount of time a large number of additional relationships can be apprehended. For example, we know simply by looking at or recalling Figure 8, that: There are exactly 3 "components" inside C; A is on top and its relationship to B is one-way; B is in the center and it is connected to C through a two-way relationship. D is in the bottom and it is not connected to the others; D is also closer to the system Boundary, and so forth. This ability to instantly apprehend this type of information through visual channels—referred to in the technical literature as "preattentive processing"—can be a powerful tool for analysis (Treisman 1986).

A model is a symbolic representation of a system that captures all of its essential parts. Depending on the purpose of the model, the representation may be mathematical, verbal, computer-based (often called a simulation) or something else. Visual models are models that use graphic notation (shapes and lines drawn on a two-dimensional medium, generally static) to represent the parts of the system. A visual model encompasses a diagram created through a formal set of rules *as well as* the rules used to create and interpret the diagram. In other words, a visual model *uses* a diagram. Thus, there is a subtle difference between the terms "visual model of a system" and "diagram of a system." Nevertheless, the two terms are often used interchangeably.

How do we visually model reverse engineering? The general scheme is as follows. The model will consist of two overlapping parts: A *target system model* and a *process model*.

The starting point of any reverse engineering project is a physical human-made technological system we refer to as the *target system*. For an illustrative analogy, we may begin by thinking of a target system in reverse engineering as a sort of battleground about which a series of decisions are made, and upon which a series of maneuvers are executed. In the course of the battle, the terrain itself is also altered. Since a model is a symbolic representation of the essential elements of something, we may think of it as a kind of map. If we think of the target system as a sort of battleground, then we can think of the target system model as a map of this battle ground. A map can be used to navigate a terrain. It can also be used as a template for writing things on. In particular, relevant

decisions, maneuvers, and landscape alterations can be recorded upon a map of the battle ground (to be later discussed or analyzed).

Following this analogy, the real-world process of reverse engineering is equivalent to a battle, the actual sequence of events: actions, decisions, and alterations to a system. And the details of a reverse engineering project, like those of a battle, will be forgotten unless they are recorded. With a suitable map and notation, we may record what happened. This is the beginning of a process model. Figure 9 shows the initial development of a notional process model of reverse engineering. The model in the upper part of the figure shows the system "intact" or pre-reverse engineering (this system model represents the target system as it is, not as the reverse engineer believes it to be). In the lower part, a sequence of relevant events has been encoded. The annotations—here in the form of arrows and red stars—mark the locus and progression of action or attention. This encoding constitutes a process model for that particular target system, like a map that records the events of a particular battle.

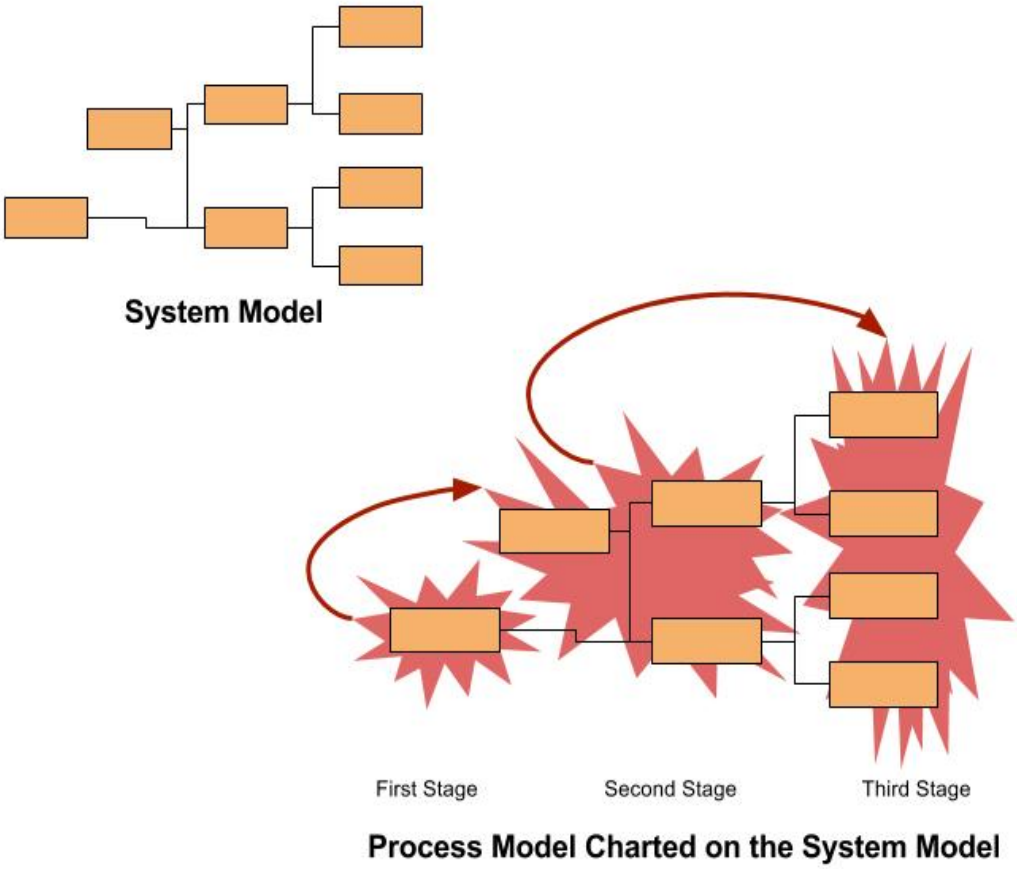

Figure 9. Charting the reverse engineer's progress #1

A generic system model is used to demonstrate the notion of "charting or plotting a process" onto the system model.

A historian may be interested in maps that record particular battles as valuable in themselves. A strategist on the other hand, is interested in the record of a particular battle only insofar as it offers a deeper understanding of *battles in general.* He seeks answers to questions like what is the usual or expected flow of events? What is needed to succeed? What are the occasions for critical decisions? What constitutes a good decision? Where lie the potential strategic errors? How can they be avoided? To this end, the strategist analyses the particular map with the goal of stripping the particulars, and distilling the essence of the battle. Our goal is like the strategist's.

Figure 10 continues the development of a scheme for achieving this goal. In the previous figure, the entire process was charted onto a single model of the system. Here,

by breaking up the process into three time periods or snapshots, the essence of the process is made explicit. We can now see that there are three distinct stages characterized by a locus of action that shifts progressively to the right (in the diagram). The final step, as shown in the lower part of this figure, is to dispense with the particular system model (which has served its purpose), and to represent each stage by a block (suitably named to suggest the nature of the events they contain).

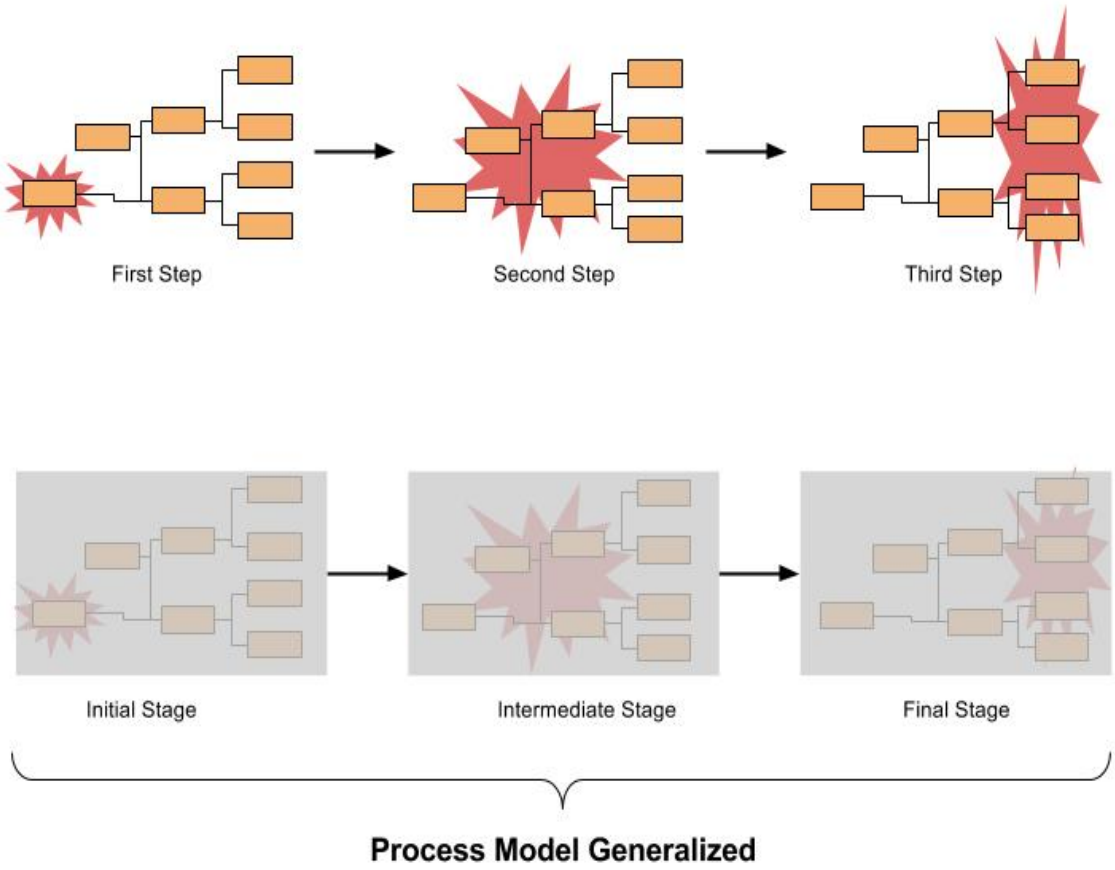

Figure 10. Charting the reverse engineer's progress #2

The same generic system model/process model is further transformed first by spreading the steps over time (analysis) and subsequently generalizing these events

Note that the 8-component system and the 3-stage process shown in these two figures are for illustrative purposes. The blocks have not yet been specified—they could stand for physical components, functional nodes, or some other thing. Likewise the actions have not been defined—the star may represent "examines this component,"

"locates this function," "opens up this thing" or yet something else. It could well be a different action in each stage (but that would probably call for a distinct symbol to stand for each different action). Finally, the correspondence of stages with "action shifts to the right" is also completely arbitrary—we have not yet attempted to establish a sequence of actions. More importantly, we have not yet proposed a general way to represent the structure of systems relative to which a sequence of reverse engineering actions may be described. Therefore, the next step is to determine: what is the right way to visualize a physical technological system.

1. A Target System Model

Not all visual models are created equal. In order to exploit the system model as a tool for thinking visually about the structure of a target system, certain characteristics should be present. First, the model should capture the essential parts of the target system. Second, the model should encode physical information about components, their interconnections and locations (i.e., information of the type: A is connected to B; A and B are parts of C; A is inside C; C surrounds A; B is closer to A than to D). Third, the system model should encode causal information about functions and effects (i.e., information of the type: A is contingent upon B; B causes C; C and D are independent). Fourth, the system model should encode information about functional allocations (i.e., information of the type: D [function] resides in E [component]; F [component] is responsible for B [function]). Fifth, the process of reverse engineering is composed of events in time, thus the system model should support encoding of temporal information (i.e., information of the form: A follows B; A and C are simultaneous). Finally, a symbol convention should support the annotation of such actions, decisions, and modifications as the system may be subjected to during the course of a reverse engineering project.

There are probably dozens of types of diagrams that can be used to represent a system. In the following paragraphs, three types will be reviewed: *functional decomposition diagrams*, *physical structure diagrams*, and *block diagrams based on SysML/UML*. Useful characteristics of each type will be identified and retained in a modified diagram to be used as the basis for subsequent analysis.

2. Functional Decomposition Diagrams

As discussed earlier, an essential characteristic of technological systems is that they are sets of objects organized in such a way that their output is a desirable effect, also known as a function. For design purposes, the system-level function (often the result of an internal chain of lower-level functions and interactions) is the most essential element of a system (Mitcham, 1994, 161–192). For this reason, the first diagram to be considered as a possible basis for a target system model is a type of diagram that was mentioned earlier in this work: the functional decomposition diagram. Figure 3 on Chapter II shows a typical functional decomposition diagram.

As a basis for recording the reverse engineering process, functional decompositions have several problems. Their main problem is that due to the unconstrained focus on functions, the blocks in a functional diagram are by definition not required to correspond with the physical components or internal locations that are important to the reverse engineer. For example, a function that is distributed throughout the system may be shown in a functional decomposition diagram as existing in a single block arbitrarily placed in some corner of the diagram. The converse is also possible, two functions shown at opposite ends of the diagram, may in fact they be physically collocated and performed by the same component. In a functional decomposition diagram, neither the number of blocks nor their location can be assumed to bear a close relation to the physical system. The end result is that a functional diagram is often not readily recognizable as the system it represents. This feature does not bode well for our purposes—reverse engineering cares not just about what the system does, but also about how it does it. This involves questions of configuration and functional allocation that are closely related to physical structure. This suggests the next type of diagram that may be considered.

3. Physical Structure Diagrams

The other major type of symbolic representation of technological systems is a physical structure diagram.[30] Through this type of diagram a "modeler" attempts to faithfully reproduce the physical characteristics of the system and its pieces. This approach to modeling fixes the problem with the functional decomposition: that the model and the system may not resemble each other. In a physical structure diagram, symbols usually have a one-to-one correspondence with the parts they symbolize; and their location in the diagram corresponds to or somehow conveys their location in the system. However, for the purpose of this work, this type of diagram also falls short of usefulness for different reasons.

The primary reason for the failure, is that information about functions and purposes is critical in the reverse engineering process, but particularly difficult to convey in a physical structure diagram. For one thing, this type of diagram contains no information about flows of energy or information. Even if flows can be incorporated into a physical structure diagram, it is not likely that this can be executed in a clear way. It is important to keep in mind the proposed reason for using a visual model of the target system: to facilitate analysis by leveraging our innate ability to reason about structure when using visual channels. As shown in the Figure 11, this type of diagram is informative, but does not lend itself to that kind of visual logic. In fact, using a physical structure diagram we might "miss the forest for the trees." In other words, there is too much information cluttering the page and probably obstructing our perception of "what is going on."

[30] This should not be confused with the structure diagrams of formal systems modeling languages like UML/SysML. These will be discussed in the next section.

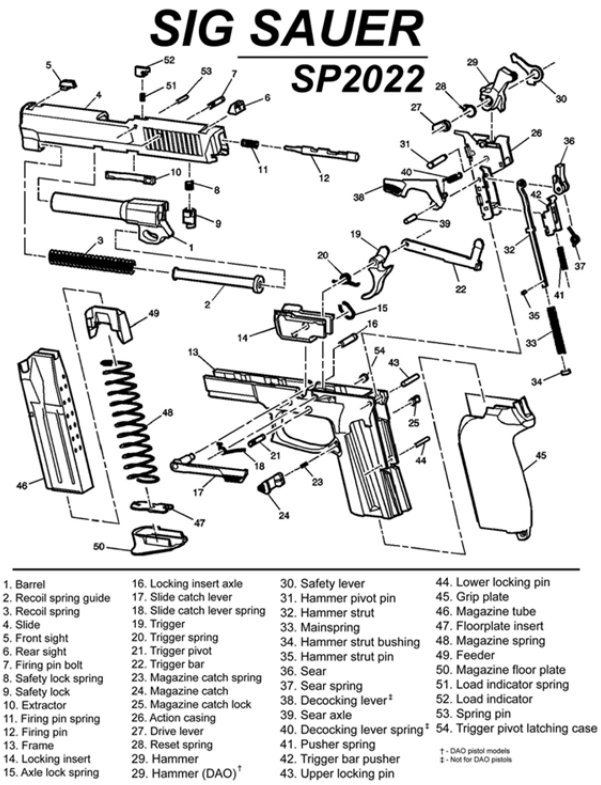

Figure 11. A type of physical structure diagram: the exploded view

Shown here, an exploded view of a firearm. Source: Sig Sauer SP2022 Operating and Safety Instructions, Version 08.04, Sig Sauer Inc. p. 26.

Instead of striving for geometrical accuracy, a structural diagram can attain greater clarity if the modeler dispenses with "unnecessary" detail and embraces a little abstraction. Many non-functional parts (or parts with minor support functions such as *keep the dust out*) may be eliminated from the diagram altogether. Important parts can be rendered in 2-dimensions and reduced to a bare minimum of geometrical accuracy, just enough geometry to suggest their function (a spring becomes a zigzagging line, a threaded component becomes a rectangle with parallel zigzagging edges). Figure 12 shows this approach. It is in essence a cartoon of the system it portrays. From the point of view of encoding system function alongside structure, and its potential use for recording what happens during reverse engineering, this is an improvement over the previous diagram.

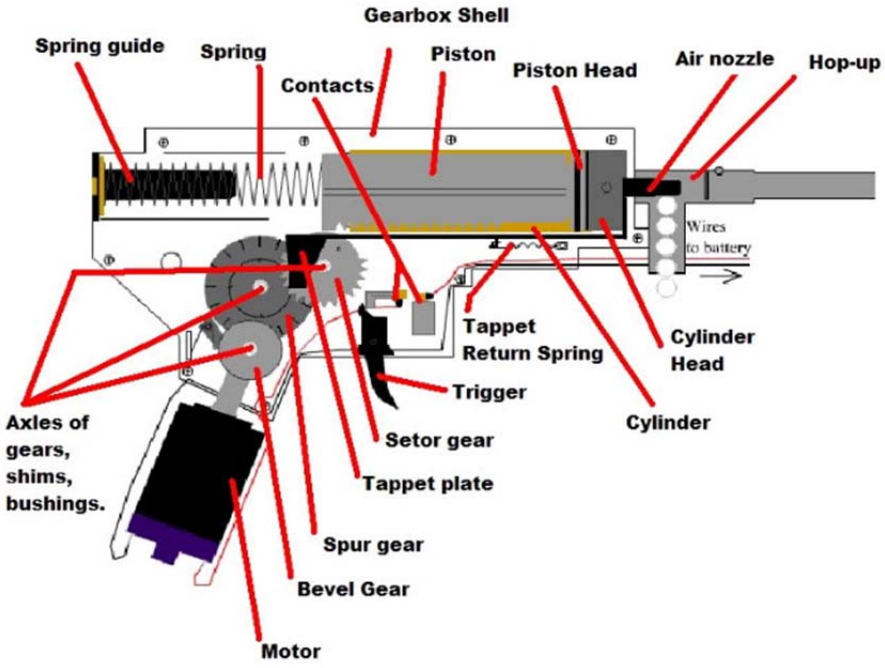

Figure 12. Internal diagram of an Airsoft gun

Matt[2], 2007, "Airsoft Gearbox," Graphic Interface Format (gif) 550x400 pixels looped animation, Available from: Wikimedia Commons, Retrieved from https://commons.wikimedia.org/wiki/File:Gearbox.gif

However, simplified structural diagrams have at least two shortcomings. The first and least important problem is that it is still difficult to make out functions and flows in such a diagram. Parts, shown as they are, are not very informative about what they do, unless you are already familiar with the system being modeled. The original source of Figure 12 is a short animation that shows the operation cycle. This combination of simplified component geometries and animation techniques is a powerful visual explainer of function. The animation takes about 5 seconds, yet it provides a complete explanation of the system function more clearly than a text-based description, in a fraction of the time and without using a single word.

However, the second—and more significant—problem with simplified structural diagrams involves an essential aspect of models that has not been brought up explicitly until now: models are useful because they generalize. Superficially different cases may all be successfully studied through a single model. Modeling two different systems will

undoubtedly result in two distinct models. However, it may be easier to find the common structure in both systems if the modeling approach abstains from using realistic or semi-realistic representations of components.

4. Combination Diagrams

If functional and physical representations fail for different reasons, one possible solution is to combine the two. A side-by-side representation of a target system as both a structure of physical components, and a structure of functional nodes might cover all the bases. Figure 13 shows this approach. It also shows a notional model of the reverse engineering process.

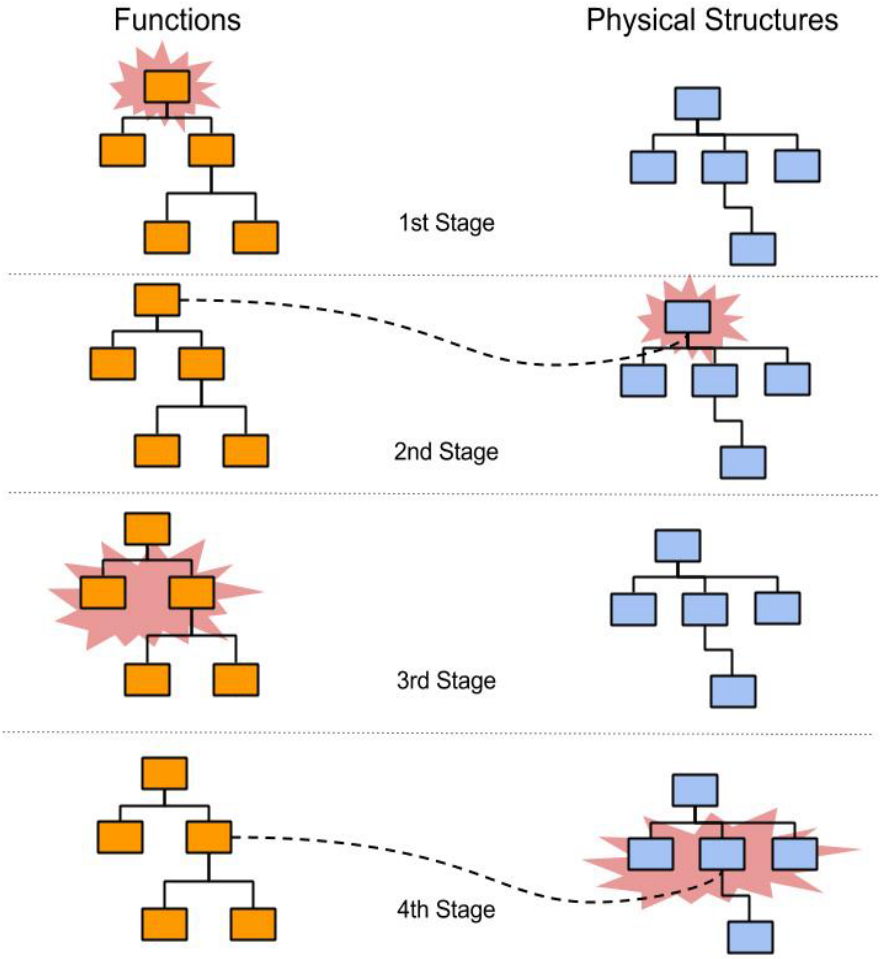

Figure 13. A possible system model

Here a system is shown as two side-by-side models depicting independent physical and functional structures

This composite diagram evolved out of a considered attempt to envision what may actually be "going on" during reverse engineering. As with Figure 10 earlier, this diagram shows a mock process depicted as a sequence of unspecified actions recorded onto a mock system model. Nevertheless, it can be "read" as follows: The 1st stage of reverse engineering consists of identifying the "top-level function." The 2nd stage consists of identifying a component or region responsible for carrying out the top-level function and performing an allocation of function to component (dashed line). The interface-allocation stage involves the identification of "second-level" or supporting functions. This is followed by the allocation of these functions to subcomponents, and so forth. A combination diagram appears to have all the necessary information to serve as a basis for modeling target systems. However, it also seems cumbersome and not very intuitive to think of target systems as being two things at the same time—even if this may be the correct way to think about them (Vaesen 2011). The complexity of this diagram might tend to obscure the patterns we hope to find. The need to capture sufficient information in the model must be balanced against the detrimental effects of adding visual complexity and clutter. Therefore, the approach was rejected when a more suitable representation was found.

5. UML and SysML

As computer hardware performance increased in recent decades, it led to corresponding demands in the complexity of software. In order to handle this complexity programming had to change by acquiring "bigger" building blocks. Assembler language was replaced by macro-assembler language, then by object oriented programming languages. Eventually (for the reasons already discussed in regards to human information processing via visual channels) this led to the introduction in 1995 of Unified Modeling Language (UML). UML is a visual language for software development. That means: it is a list of rules and definitions that can be used in conjunction with drawings of various types of boxes and arrows in order to describe a computer program (Weilkiens 2007, 144–146). UML provides a vast amount of such rules and definitions. Figure 14 shows just a very small fraction of the catalog available to UML users.

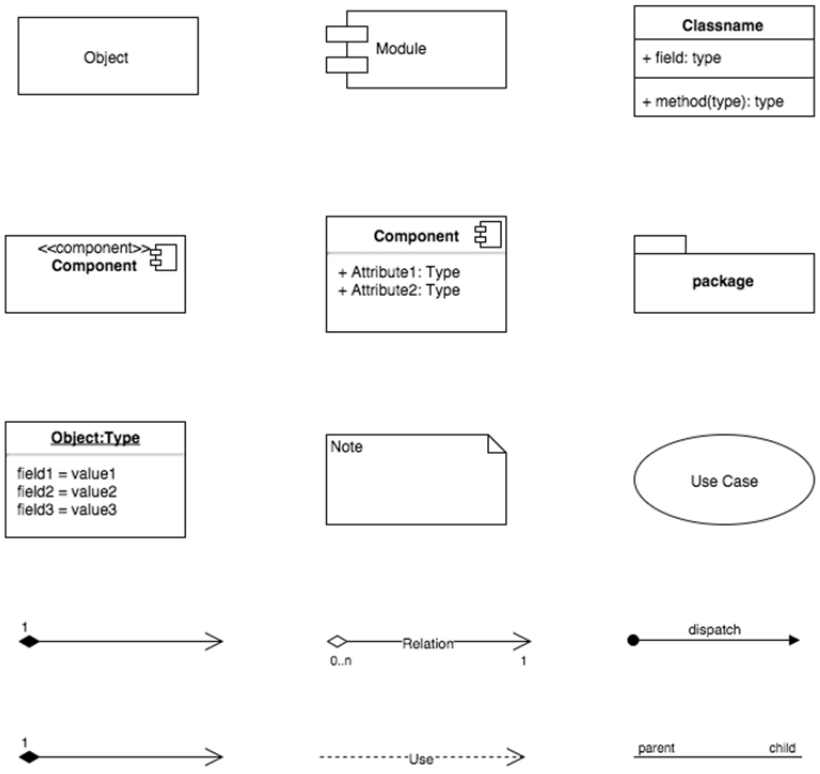

Figure 14. UML symbols

A sampling of UML Symbols

UML has also been used for modeling a variety of structures and behaviors in applications other than software. Systems Engineering is among the disciplines that use UML. In 2001 INCOSE established as its goal the adoption of UML as its standard language. In spite of its flexibility, basic UML is software-centric. For this reason, an extension to UML (an additional set of rules and definitions) known as Systems Modeling Language (SysML) was published and is now considered the standard language for modeling in systems engineering (Weilkiens 2007, 223–225).

Figure 15 provides a taxonomy of UML/SysML visual models. It shows that there are two broad categories: *behavior* and *structure* diagrams. Structure diagrams have some features that may be useful for our purposes. In UML this family of diagrams includes class, component, and object diagrams, in SysML all these terms are merged under the single label of *block*. Two types of structure diagrams from SysML will be reviewed next: *Block Definition Diagram (BDD)*, and *Internal Block Diagram (IBD)*.

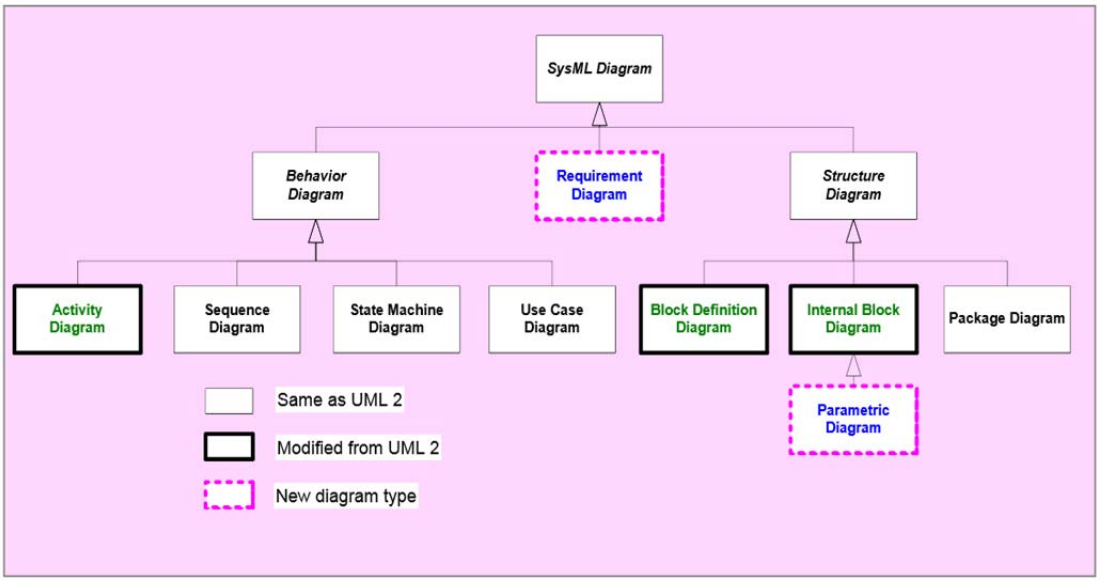

Figure 15. SysML taxonomy

Figure shows two broad diagram categories: Structure and Behavior. Among the Structure diagrams: two types of diagram that have been adapted from UML to SysML in order to provide clearer representations of systems other than software. Source: M. Hause, 2006, "The SysML Modeling Language," Fifth European Systems Engineering Conference, Gloucestershire, UK, September 18–20.

6. SysML—Block Definition Diagram

These diagrams are similar in content and organization to the parts inventories found in user's manuals for systems where assembly is required. Figure 16 shows a block definition diagram (BDD) for a distiller. The blocks represent systems, subsystems or components. The vertical dimension in the model is used to represent hierarchical relationship. In other words, if a block is a composite, then its components will be shown as branches below it. The external (white) block identifies the type of diagram (as BDD) and the name of the system. In this case the top block is the system: a distilling plant. The three blocks under that are the main subsystems: heat exchanger, boiler, and drain valve.

The two blocks under that show the components of the boiler as a furnace and a steam-drum. There are certainly more components than shown, for any of the composite blocks. The level of detail has been kept at a minimum for clarity. The BDD may incorporate information about the attributes of an object. For simplicity, attributes (valve material, valve type, valve size) and their values (bronze, globe, 6") are shown only for one of the blocks. In an actual BDD, every block might specify many attributes.

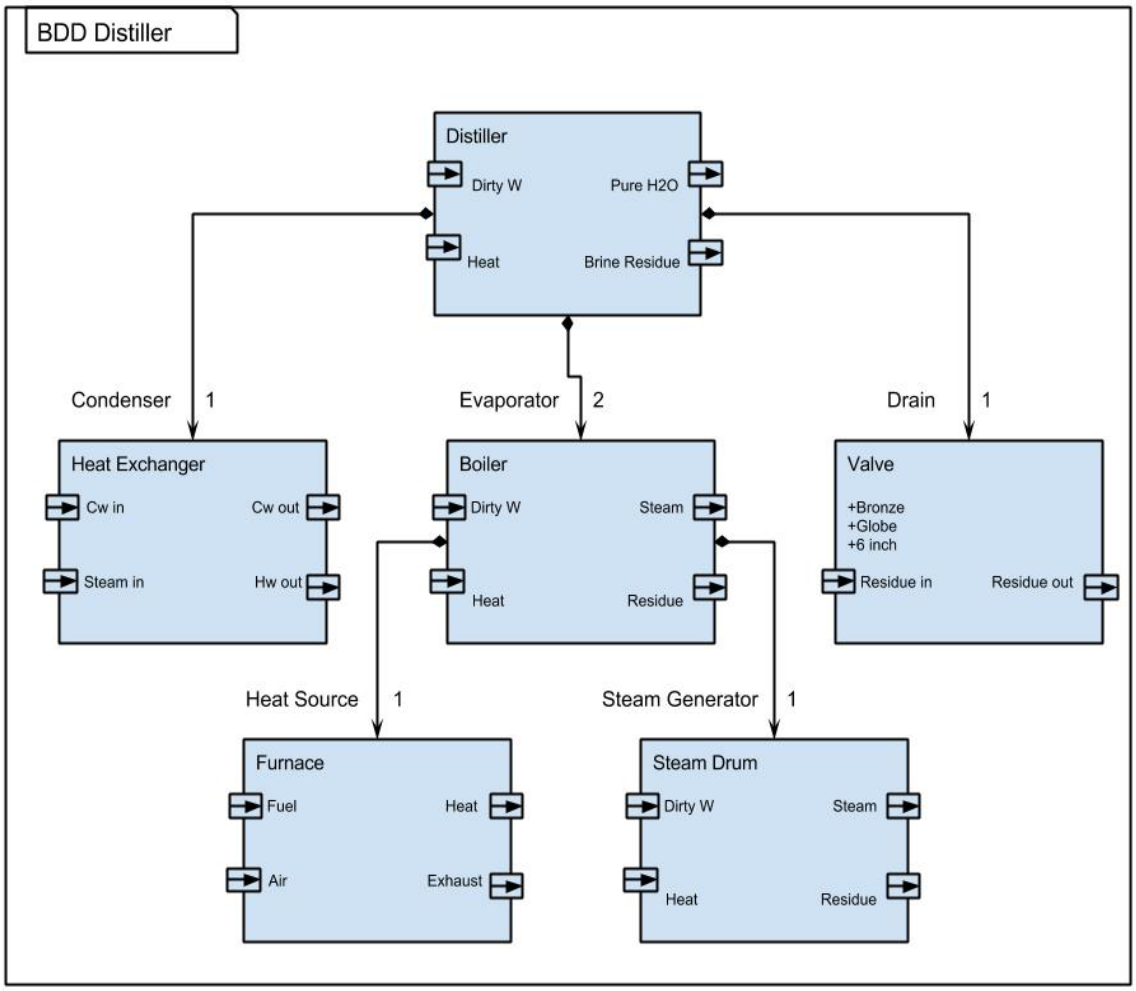

Figure 16. Simplified block definition diagram for distiller

Adapted from: G. Finance, 2010, "SysML Modeling Language explained." OMGSysML.com, Retrieved from http://www.omgsysml.org/SysML_Modelling_Language_explained-finance.pdf

Connecting lines in a BDD indicate the type of relationship that exists between two objects. In this case, the line with a filled-in diamond at the distiller and an arrowhead at the boiler is read as *"distiller has a boiler"* (or, since visual models are not constrained by a reading direction, *"boiler belongs to distiller"*). There are formal symbols to represent a number of other relationships not shown in this example. For instance, if a component belongs to the distiller system but is not a physical part of it, this "weaker" relationship could be represented by using an open diamond at the distiller end.

Multiplicity and role information are shown alongside the relationship lines. In this case there are 2 boilers with the role: evaporator, 1 heat exchanger with the role: condenser, and 1 valve with the role: drain. The boiler in turn has 1 furnace with the role: heat source and 1 steam drum with role steam generator. Finally, flow ports indicate the presence of flows (material, energy, or information) that cross the block boundary (in and/or out).

There are several reasons why a BDD is not suitable as the basis for a target system model. One problem is similar to the one noted earlier in reference to functional decompositions: the diagram symbols do not "map" directly to the components they symbolize. For example, the number and location of any given component is not explicit visually in the diagram (although some of the information is there, its extraction requires reading text-based annotations). Another problem is similar to that of the physical structure diagram: it is difficult if not impossible to explicitly show functional relationships and flows using the BDD. The flow ports signal the existence of a flow in or out of a component, but not its source or its destination. And the connecting lines may be mistaken for flow-paths, but they are not. (Admittedly, showing this type of information was probably never the intent of the designers of the SysML BDD).

7. SysML—Internal Block Diagram

In the internal block diagram (IBD), the hierarchical structure relationships (information like "has a," "belongs to," and so forth) are represented by the placement of the blocks. In other words, component blocks are drawn within composite blocks. Connectors and flow ports are used to indicate *item flows* and are usually labeled (for

example: heat, water, steam). Flows in a composite block are associated with flow ports in component blocks. Flows and flow ports have specified direction components shown (optionally) by arrows. Although the IBD and the BDD use the same number of blocks to depict a given system, the use of block placement to convey hierarchical structure information frees the use of connectors to convey flow information. The resulting system to model mapping is more intuitive and a better fit for our objectives. Figure 17 shows an IBD for the same distiller system (many of the labels of a full diagram have been omitted for clarity).

The labeling of the blocks is also slightly different. In the IBD each block is titled "role: name [multiplicity]." The role and multiplicity in the component label are the same as they were alongside the relationship arrow in the BDD. It is unclear what is gained by this change. The use of a text-based annotation to convey component multiplicity remains a source of visual dissonance between the model and the real-world system (both in the BDD and the IBD). This is a problem given the objective of this dissertation. For example, it would be difficult to visually show something like "at time x, reverse engineer disassembled the number #1 boiler while keeping the #2 boiler fully operational" using either the BDD or the IBD.

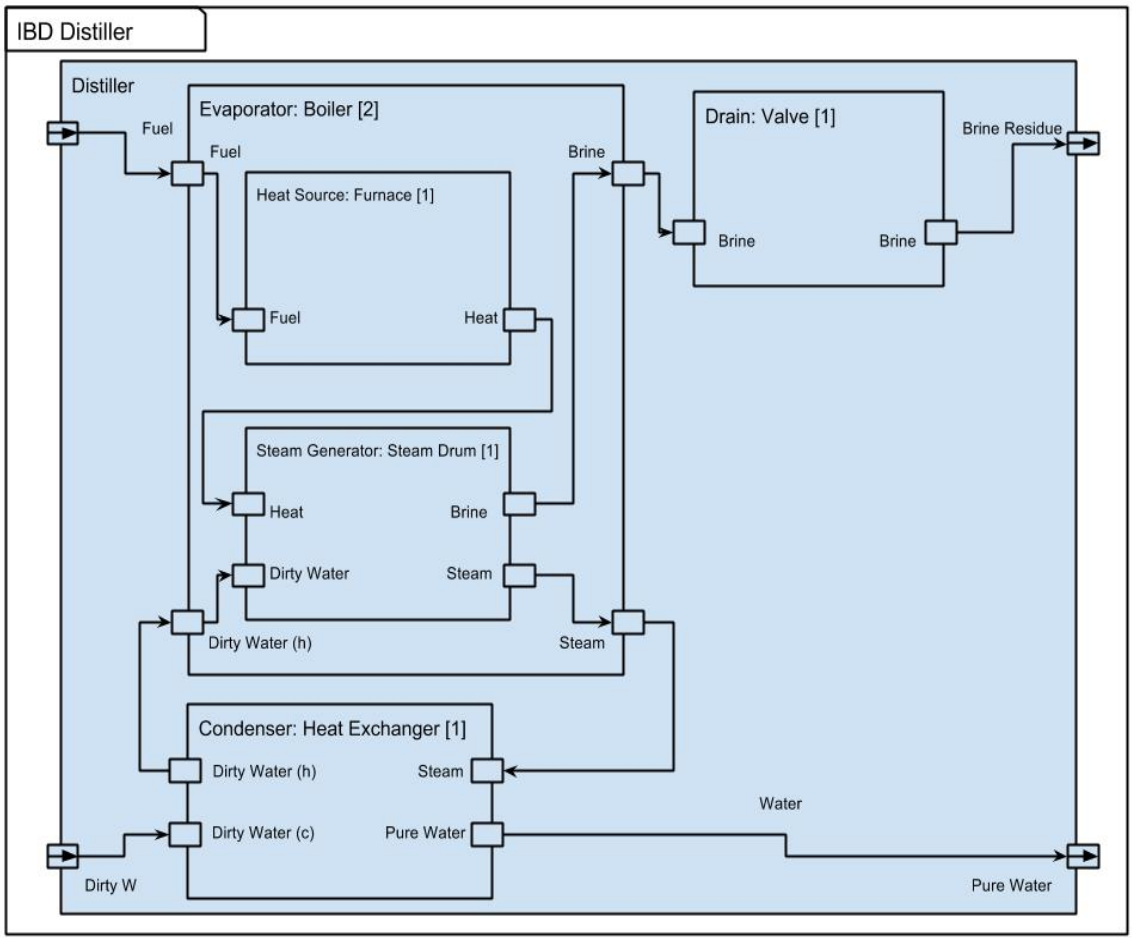

Figure 17. Simplified internal block diagram for distiller

Adapted from: G. Finance, 2010, "SysML Modeling Language explained." OMGSysML.com, Retrieved from http://www.omgsysml.org/SysML_Modelling_Language_explained-finance.pdf

8. Improved System Diagram

The objective of the last several pages was to review three types of diagrams in order to determine their suitability as system models to be used as a basis for a reverse engineering process model. No single diagram type was found adequate. The representations tend to be cluttered, distorted, or incomplete. However, each type of diagram was found to have some useful elements. An improved diagram type is suggested in figure 18. The new diagram incorporates useful elements from the diagram types reviewed, while minimizing elements that were found to be redundant or to cause

distortion. The improved diagram for the distiller is shown in Figure 18. This type diagram may be called a *system diagram for the analysis the reverse engineering*—but from here on it will usually be referred to simply as *the system model*. The subsequent paragraphs outline some of the diagram's features and their justification that fall into two categories: *reducing clutter*, and *adding information relevant to reverse engineers.*

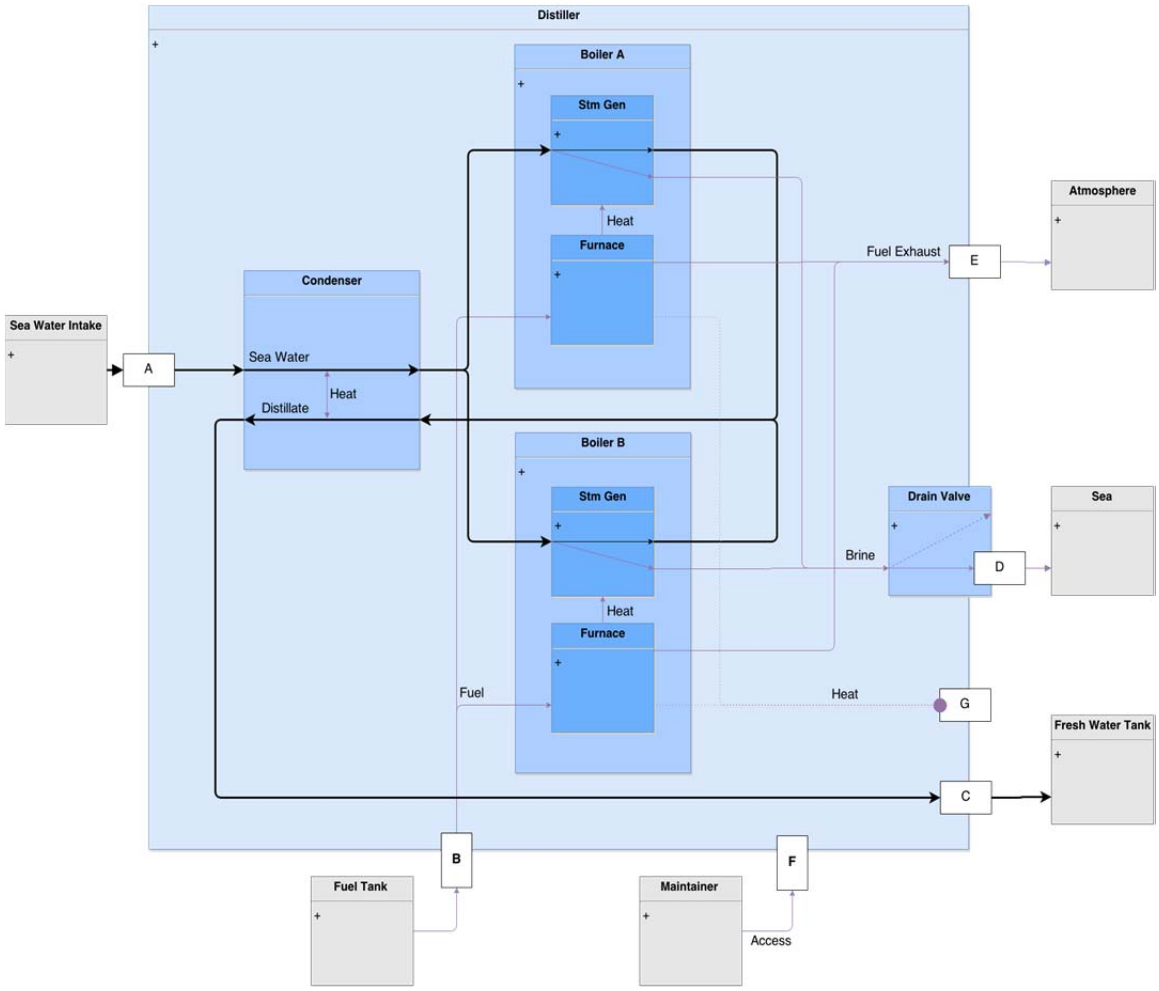

Figure 18. A system model of a distiller

This is similar to other block models but it has been designed to make explicit the sort of information about the system that the reverse engineering process is likely to deal with (Example: the physical location and redundancy of key components are made explicit)

a. Reducing Clutter

Most of the diagrams reviewed tended to be cluttered. The clutter has several sources such as redundant labeling; unclear graphics; careless use of the space; and others. In order to capitalize on the power of visualization, the clutter should be eliminated as much as possible. To accomplish this, a number of new features were incorporated into the system model. These are listed along with justifications:

1. **Dual labels of Role and Name have been substituted by a single label**—In the BDD and IBD, it is unclear why Boiler is a "name" while Evaporator is a "role." Or is it the other way around? Essentially, system components are already typically named after what they do, so why have more words than necessary.

2. **Multiple labels along single flow paths removed**—As a single flow crosses into and out of system boundaries there is no need to identify it multiple times. For example the IBD shows pure water leaving the condenser boundary, it is identified at the flow port, then it is labeled again as a flow item, then again there is a third label at the system-level flow port as the water leaves the distiller. A diagram-reader should have no difficulty identifying a flow represented by a clear and continuous line that has been labeled only once.

3. **Rounded corners added on flows**—This may seem trivial, but lines that represent system or component boundaries and lines that represent flows and interactions can be confused with each other, at least during pre-attentive processing. The use of rounded corners in flow lines makes intuitive sense (pipes, wires, and other conduits tend to have rounded corners). It also helps distinguish flows from boundaries, and non-interacting flow crossovers from bifurcations and merging flow paths.[31]

4. **Line bends and line crossovers minimized**—In spite of the incorporation of rounded flow lines, multiple line crossovers still can make a diagram cluttered. Accordingly these were minimized wherever possible by relocating blocks.

5. **The primary flow is emphasized**—When there are multiple flows involved it improves clarity if the primary flow (the one associated with the system's purpose) is emphasized by a thicker line.

6. **Exterior diagram ID block removed**—One challenge of incorporating visual models into a hard-copy document such as this, is that as the part

[31] There are other conventions for distinguishing these, however, the use of rounded corners has the advantage of being intuitive and being easy to implement.

count increases, the part size must decrease until labels become unreadable. Space is at a premium. Only one type of diagram will be used henceforth. Therefore, there is no need to add an extra box surrounding the system model only to inform the reader what type of diagram this is.

7. **Color was added**—The primary reason for this is not aesthetics. The color enhances the pre-attentive processing by making blocks easier to distinguish from flow lines, and from each other.

b. Adding Information

Reverse engineering has some elements in common with related engineering activities that routinely use system models (design, maintenance, assembly and others). However, reverse engineering is also a unique activity. Much of what is done in reverse engineering is not done by other engineering activities. Accordingly, a diagram that is to be used in modeling reverse engineering has unique representation requirements. To satisfy these requirements a number of features have been incorporated into the system model. These are listed along with justifications:

1. **Context systems**. Neighbor systems have been added (use of a different and subdued color is used to minimize the resulting additional complexity)

2. **Component Internal Flow Characteristics**. Some blocks are modified to incorporate additional information about internal flows. This reflects the reverse engineer's concern with "what is going on inside?"

3. **Flows that Terminate or Transform.** When a flow terminates or is transformed inside a component, that line is shown as terminating. For example, the flow of fuel terminates in the furnace and the flow of exhaust begins there, therefore these are shown as distinct flows.

4. **Flows that Split.** A flow that splits inside a component is shown as splitting inside the block. For example, seawater splits into distillate and brine inside the steam generator.

5. **Valve Functionality**. A flow that is contingent on the position of a valve can be shown in a similar way to the split flow. In the case of a valve with just open/shut one flow path leads through, the other flow path terminates inside the component. For example, brine leaves the distiller through a valve in a normal OPEN configuration. The valve shows an optional configuration SHUT where the brine dead-ends, and does not reach the sea. The same convention can be used for electric switches.

6. **Multiple Flow interactions**. If two flows enter a component and interact indirectly this is shown in the block. For example: seawater and distillate

flows do not mix inside the condenser. They travel counter-parallel to each other and exchange heat—all of this is shown.

7. **An expanded concept of flow—part 1.** Lines that connect components signal the presence of intentional interactions.32 In general, these interactions can be thought of as flows in the traditional sense (matter, energy, or information). However, more subtle interactions may also exist, and need to be represented. For example, under some circumstances a potential user of a hand held system (perhaps a toy or a power tool) may place value upon the characteristic should not feel like it is made of plastic. In response to this implied requirement, the designer may add a piece of metal within the system boundary whose sole function is to convey the desired sense of heft to an otherwise plastic and cheaply made system. Here, a simple piece of metal becomes a functional component. Its weight as a function/flow can be represented as a line originating at the piece of metal, traveling across the system boundary through an interface and interacting with the user.

8. **An expanded concept of flow—part 2.** An opposition to flow is also a category of function that occurs often, and may be represented in a similar way. For example, a dust cover (whose function it is to block the inward flow of debris) or a radiation shield (whose function it is to block the outward flow of neutrons) may be represented by lines that terminate in a square arrowhead. These functions can only be detected when the shield is removed and the previously blocked flow is unblocked. The system model of the distiller shows this notation for heat produced by the furnace, but leaked to the system boundary, instead of the steam generator. Interface G of the system boundary is its insulation. The dotted line traveling to D is used to indicate the leaked heat is unintentional. (A dotted line may also be used to convey an intermittent flow)

9. **Multiplicity is made explicit.** If there are two steam generators in a distiller, it is conceivable that reverse engineer may treat each differently. Perhaps one is disassembled while the other one is kept operational. Accordingly, the actual number of components matches the modeled number.

10. **Spatial Relations.** Relations like “Component A is under Component B,” or “Component C is part of the system boundary” are important in reverse engineering. Accordingly, an effort is made to representing things in their

32 Unintentional effects are by definition not outcomes of the design process, therefore they do not concern the reverse engineer except when an object/attribute is incorporated to block or reduce an unintentional effect. However, their existence suggests there can exist an activity similar to reverse engineering (perhaps even concurrent) that is concerned with identifying non-designed effects, and tracing these to their object/attribute sources. Such an activity may be called "design troubleshooting" if we assume that these un-designed effects are bad.

approximate spatial relation (this effort can never succeed completely given the small size of a page, as well as its having only two dimensions). For example, the diagram preserves the top/bottom relationship of furnace and steam drum.

11. **Interfaces.** In an earlier discussion, interfaces were introduced objects plus attributes incorporated by design to make contact with other objects in order to cause desired effects or functions. Interfaces are shown in the system model as smaller blocks located at system or component boundaries. Though similar and related to flow ports, interface blocks have a more flexible meaning useful in the context of reverse engineering. The interfaces shown in the boiler diagram all convey normal flows, so consider a different example of an autonomous vehicle that has the function self-righting. The interface responsible for this function may be a particular geometry and weight distribution that makes it unstable when upside down. The system model for this vehicle will include an interface box labeled Geometry and Weight (or perhaps two separate boxes).

12. **Input/Output Convention.** Visual models are more likely to serve their purpose of helping grasp structures and patterns, if the diagrams for different systems are as like each other as possible, without sacrificing accuracy. To this end, an effort is made to establish a convention wherein main flows come in from the left and main outputs go out on the right, while auxiliary flows come in and exit through the bottom of the page.

13. **The System Boundary.** The physical system boundary is a critical aspect of reverse engineering. Accordingly, the system model boundary is understood to represent the physical boundary. This has some corollaries: For example, a component that is part of the physical boundary is shown at the model boundary. Likewise, system-level interfaces are also shown at the system model boundary. It is presumed that the system boundary generally restricts material and physical access to the system internals. Accordingly, an interface that is not viewable from the exterior (perhaps a thinner section of the shell designed for a controlled blow-out in the event of overpressure) can be shown as an interface block located just within the boundary. In some instances a system has no physically obstructive system boundary, perhaps in these instances the system block can be drawn in dashed lines, or not at all.

Note: The system model is a tool for the analysis of the reverse engineering process. The process itself involves the reverse engineer gradually developing a mental model of the target system. In this work the system model is considered ground truth. A reverse engineering project is successful to the extent that the mental model in the reverse engineer's mind approaches the system model. In order to preclude confusing the two,

the model in the reverse engineer's mind will usually be referred to as his or her "working model."

9. The System Model and the Reverse Engineering Process

Thus, far, this chapter has shown the development of a *system diagram for the analysis the reverse engineering* (or system model). The final part of this chapter will test whether the system model can be used as the basis for expressing the practical aspects of reverse engineering introduced earlier in this work. The implications are taken verbatim from Chapter III.

Implications: ***I look around the target system in order to discover what it is for***. I look at it from different perspectives. Determine the boundaries of the system. Figure 19 depicts an external system inspection. The yellow highlighted regions are attention-areas, or areas where the attention of the reverse engineer is being directed.[33] The inclusion of context systems suggests the reverse engineer may be tracing flows from the target system back to the originating systems. The numbers and arrows indicate the reverse engineer looked from the seawater inlet perspective first; the fuel supply second, maintenance access third, and ended by inspecting all output flows more or less simultaneously.

[33] Figures 19 through 29 use the same convention. Attention-areas are yellow or orange highlighted areas. In the event these figures are not reproduced in color, attention areas are also easily distinguishable from other elements of the diagram (like components and subcomponents) by their rounded corners.

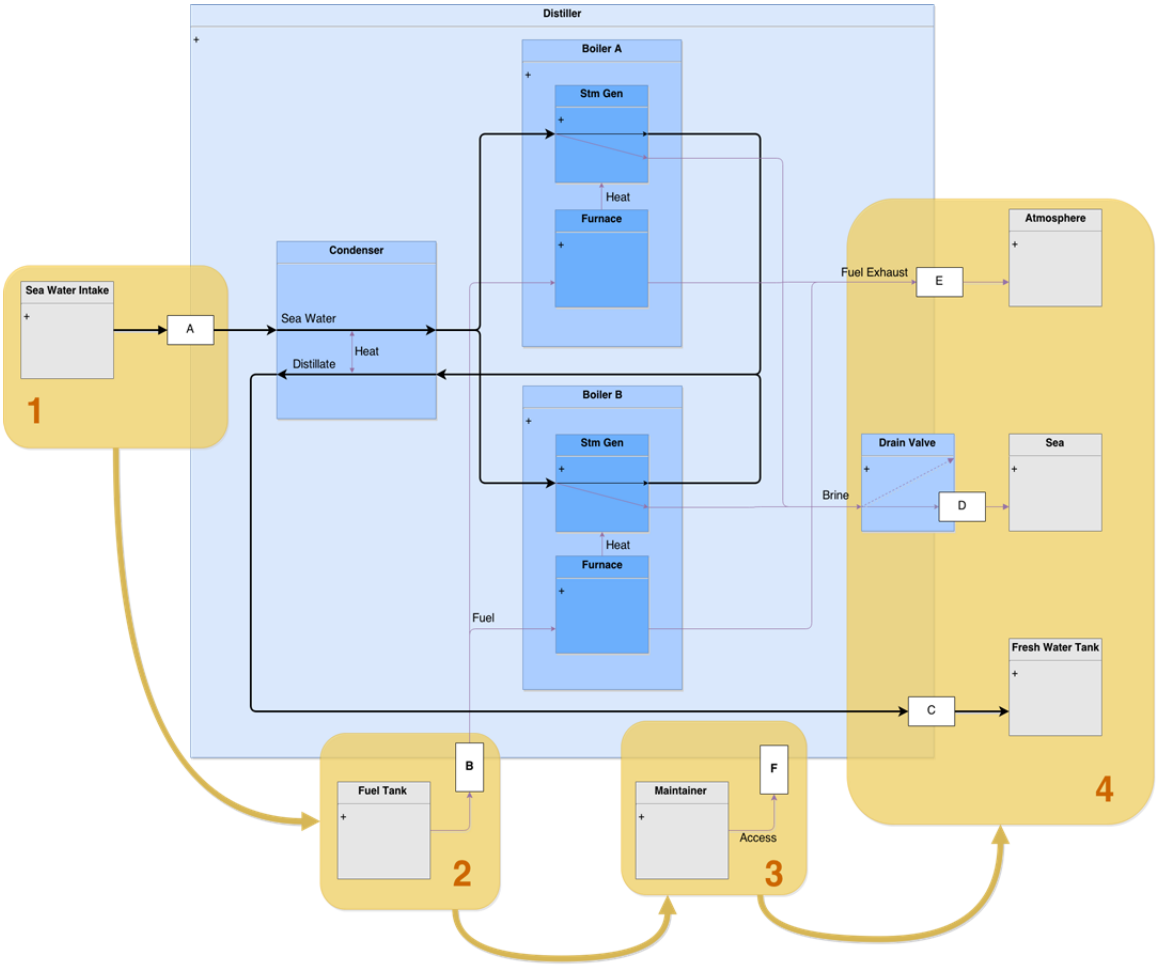

Figure 19. System model used to depict initial context inspection

Attention-areas shown in yellow and with rounded corners show where the attention of the reverse engineer is being directed. The yellow arrows and numbers are used to indicate the shift in the reverse engineer's attention as the target system surface is explored left to right.

Implication***: I identify what other systems the target system interacts with and the nature of the interactions.*** My goal is to learn all system-level functions. Identification is a cognitive action. Accordingly Figure 20 depicts the state of the reverse engineer's knowledge rather than physical steps. Specifically, the green coloring on the context systems and system-level functions or flows suggest that the reverse engineer has correctly answered the questions of what things the target system interacts with and how.

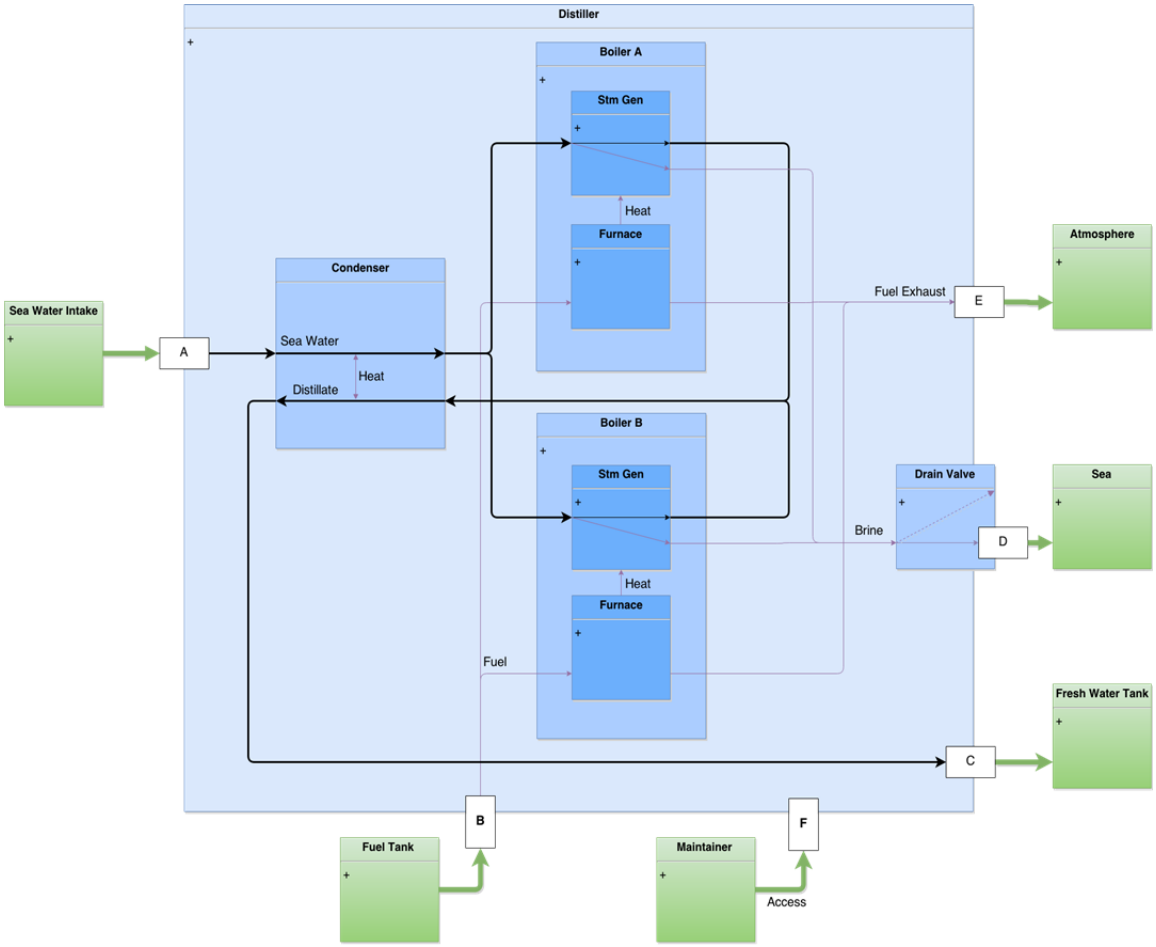

Figure 20. System model used to depict information learned as a result of the initial context inspection

Components or systems are colored green to indicate their correct identification by the reverse engineer. Different shades of blue are used to improve the preattentive processing quality of the diagram (Treisman 1986).

Implication: ***After doing that I trace each function to a physical interface on the boundary of the system.*** Figure 21 once again shows the loci of attention and action for the reverse engineer who is now attempting to answer—at a superficial level—how the various actions are performed. For example: In the last step he ascertained that the system can be accessed for maintenance. In this step he seeks to learns how and where.

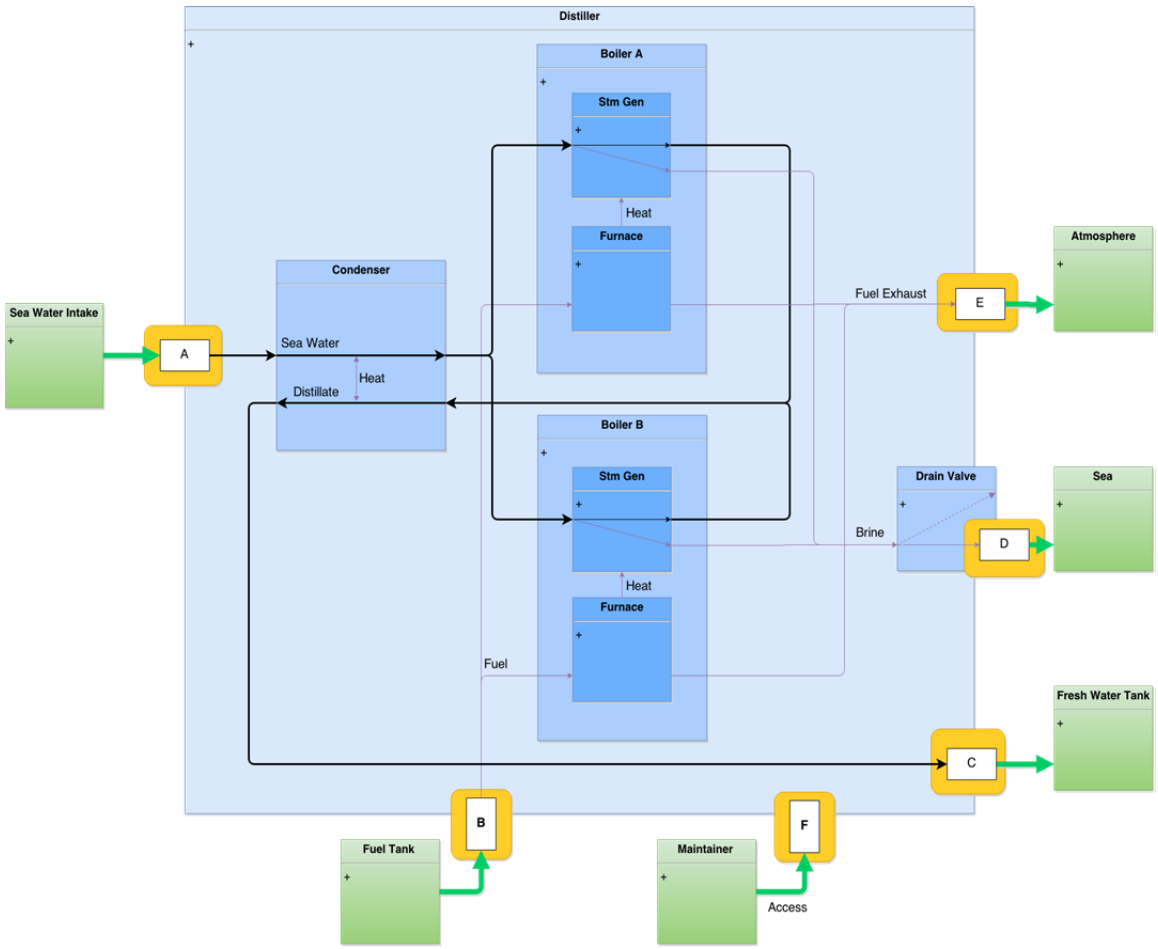

Figure 21. System model used to depict system boundary inspection

The inspection is aimed to identify functional interfaces (the white rectangles found throughout the system boundary). Also shown (by green color) the information that has been learned in previous stages.

Implication: ***Next I must determine how to get inside.*** It is around this point that the reverse engineer has learned all that can be learned as an outside observer. This may be called the *breaching point*. As mentioned earlier, a system boundary often has a function that is uniquely important to the reverse engineer: to keep outside all things that do not belong inside the system (this generally includes the reverse engineer). Accordingly, Figure 22 shows that the reverse engineer's attention has shifted to the system boundary, as an obstacle to be overcome. The diagram also shows the reverse engineer's newly acquired knowledge regarding the interfaces.

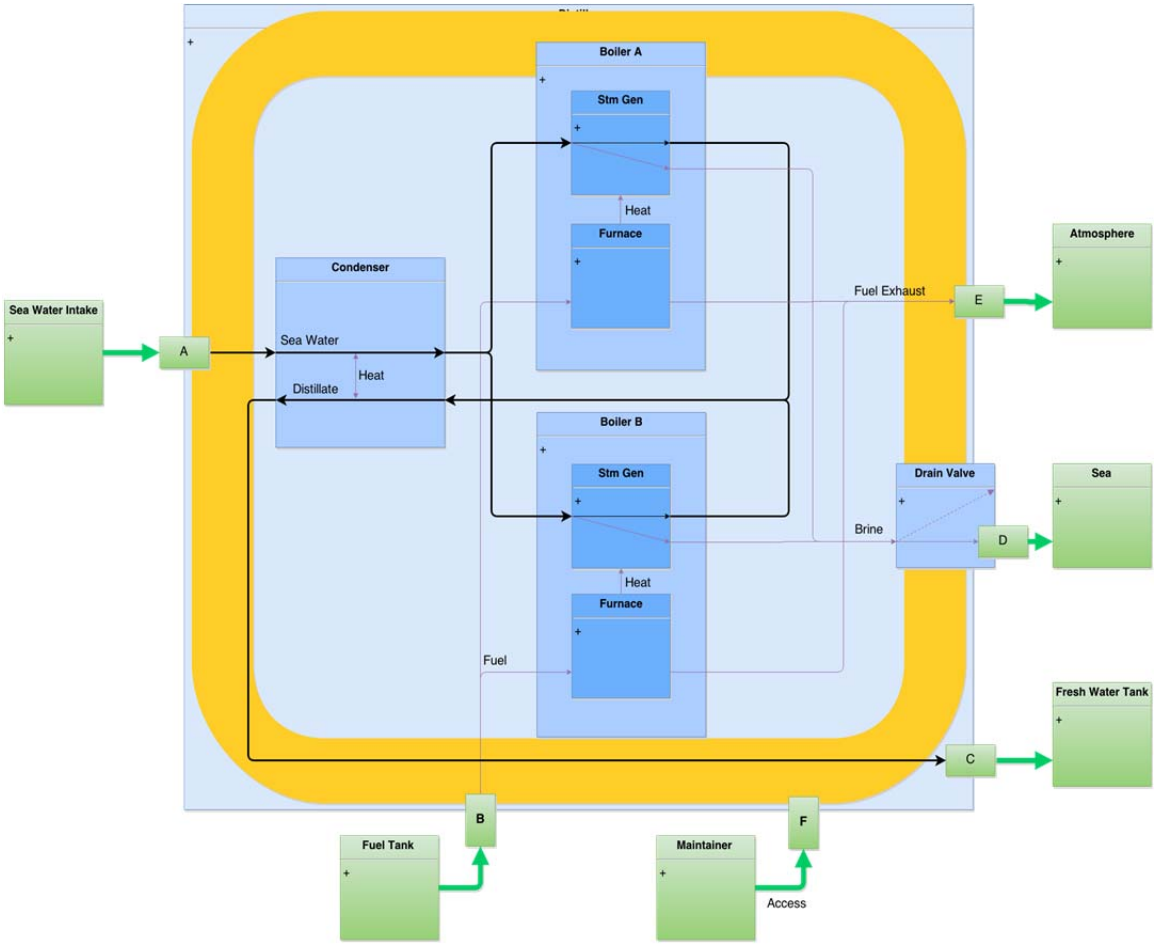

Figure 22. System model used to depict system boundary inspection

The attention-area is moved just inside the system boundary to indicate that this part of the inspection is aimed to identify a point of entry in preparation for breaching the system boundary. Also shown is the information that has been learned in previous stages.

Implication: ***My goal is to breach the system boundary in order to gain unobstructed material and visual access to the internal components.*** This may involve a tear-down.[34] Lastly, Figure 23 depicts successful access. Note the ideal state of affairs at this point is to have the system boundary removed without altering any of the system functionality (at least in a bench setup) this is shown by removing all but the faint outline of the system boundary block. In practice full functionality after breaching will often be impossible or highly undesirable, as in the case of a distiller. In the event the reverse

[34] Tear-down and partitioning are related activities, as used here the first involves a physical separation of the components while the second involves the mental or conceptual separation.

engineer successfully removes the system boundary (including all the insulation) it is not likely that he will want to be around to see the system operate in such condition.

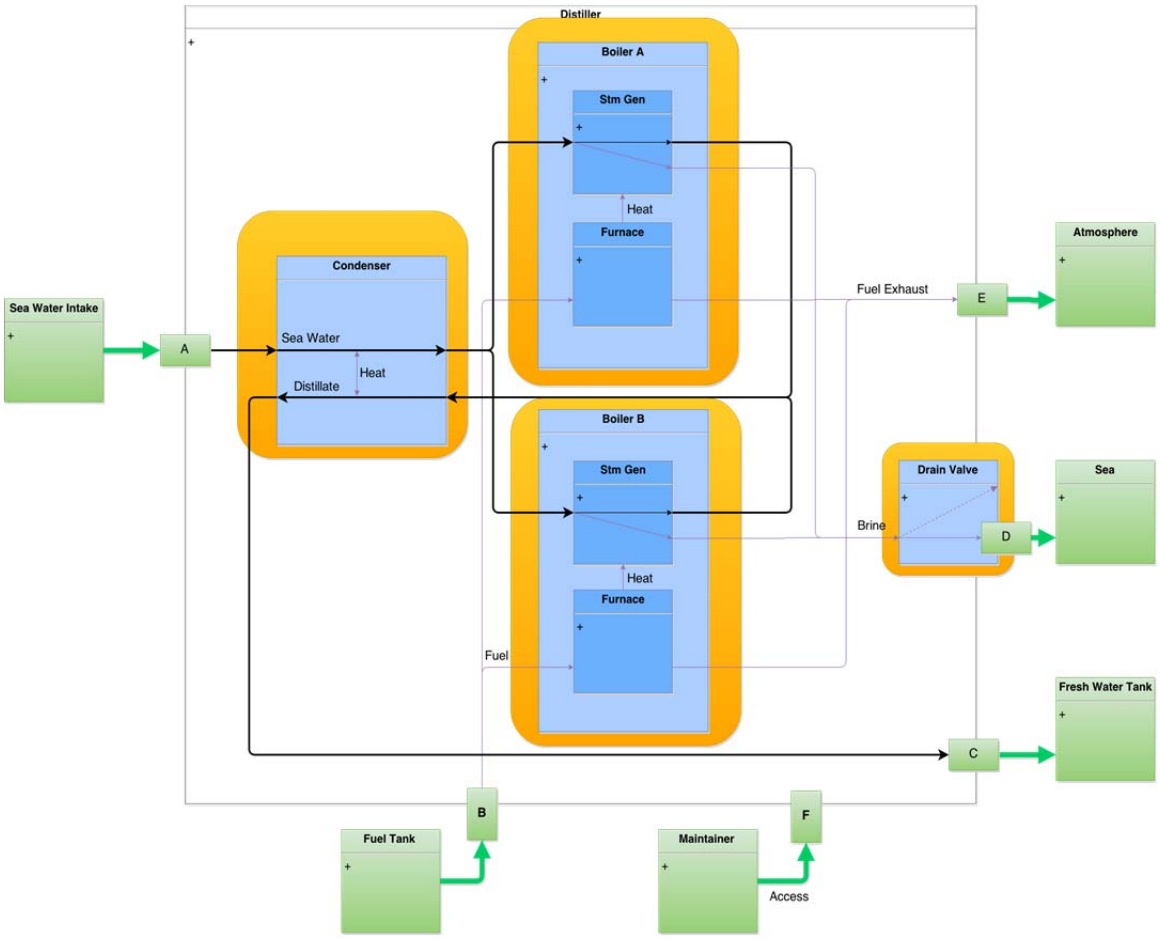

Figure 23. System model used to depict a successful system boundary breach by removing the color inside system boundary

This model also shows the subsequent focus/task of the reverse engineer: subsystem partitioning. Another way to think about this is as the initial context inspection, this time performed at the subsystem level. As before, also shown is the information that has been learned in previous stages.

Implication: ***Finally, each subsystem is in turn subjected to the same process.*** Figure 24 shows the reverse engineer's attention has shifted to one of the boilers. Note that this shift in perspective alters the diagram in new ways. For one thing, interface blocks are now visible at the boundary of the boiler subsystem. Also, the other subsystems within the distiller are now shown as context systems.

The drilling down is likely to bring into focus subsystems of a very different nature from the parent system. For example, reverse engineering of the distiller may uncover electronic monitoring and control subsystems. For large or complex systems consisting of many layers and incorporating diverse operational principles, the "reverse engineer" will probably be a multidisciplinary team that splits up to handle different parts of the system as they are made available. With each drilling down the requisite knowledge of the reverse engineers responsible may be expected to become increasingly specialized. This process mirrors the system design process in some respects.

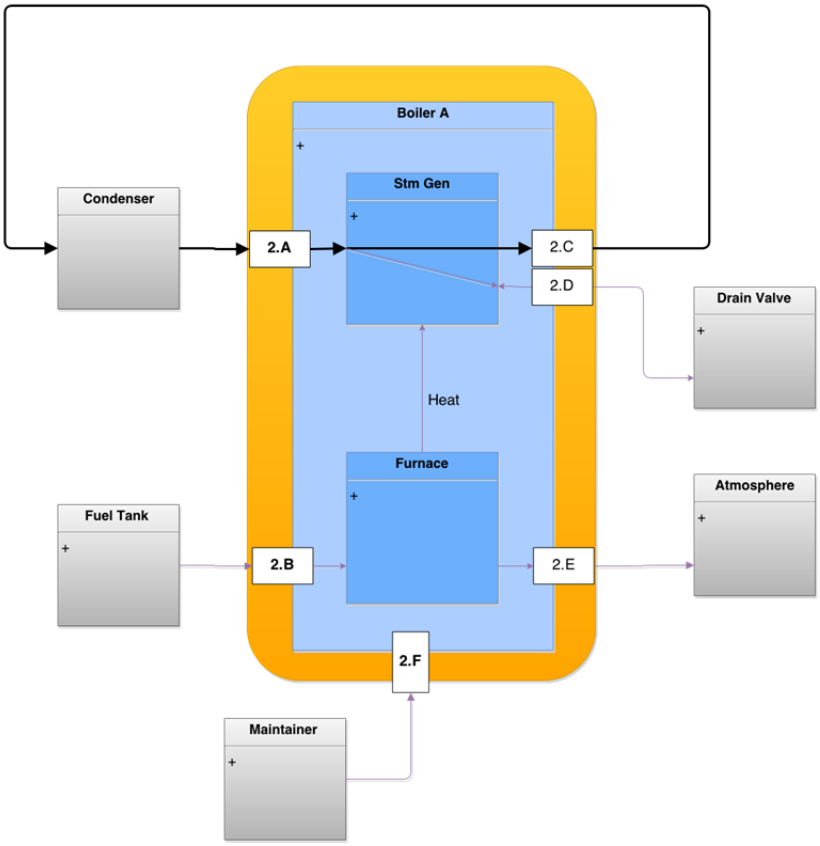

Figure 24. (Sub)system model used to depict post-breach situation

The situation mirrors that shown in Figure 19 except that the reverse engineer is now concerned with a subsystem (or component) as opposed to the target system

C. CONCLUSION

First, the use of visual models or diagrams for the analysis of reverse engineering was justified. After reviewing three types of diagrams, a new type was described in which

useful elements were incorporated and detrimental ones discarded. Then the diagram was tested by mapping some of the practical implications of reverse engineering listed at the end of Chapter III. The resulting notations are shown in Figures 19 through 24. Taken as a whole, these figures constitute an answer (albeit incomplete) to the question of what the reverse engineer does. In other words, this is the first draft of a visual model of reverse engineering. The model has not been generalized to systems other than the one considered in the example. Also, in so far as the model describes reverse engineering actions, it does so only for an ideal process. The types of problems that may arise in a real-world process have not been discussed. As the notation comes into contact with more realistic scenarios and eventually perhaps with real-world projects in the following chapters, it will be refined and expanded to express needs and problems that have not yet been anticipated.

V. A PROPOSED MODEL OF THE REVERSE ENGINEERING PROCESS

> The discovery of a differential gear [within the Antikythera mechanism] was breathtaking. It combined astronomical knowledge, abstract mathematical understanding, and mechanical skill.
>
> —Jo Marchant, *Decoding the Heavens*

A. INTRODUCTION

The objective of this Chapter is to take the information from the previous chapter, and present it in a more visually concise form. This will allow the structure of the process to become more explicit, showing reverse engineering to be an iterative process of **progressively inward focus**. That is, the reverse engineer discovers information by repeatedly pursuing answers to the same set of questions or problems. With each repetition, the focus of the reverse engineer's action and attention moves inward through the system. Information uncovered this way results in an increasingly detailed working model of the target system. The working model forms from the outside in, or what is traditionally referred to as from the top down. This simple idea about reverse engineering will be the starting point for the next chapter in which a new question will be raised: *What could go wrong?*

B. RESULTS

The previous chapter suggests that reverse engineering consists of four stages as follows:

1. Context-Exploration Stage: Define the Boundary

This stage kicks off the process with a thorough external inspection of the system and its surroundings. The aim is to define the target system boundary and then identify all the systems with which it was designed to interact (aka the target system's context). The system's purpose should become clear. Defining a boundary is a subjective decision with important implications for the rest of the project. The word *define* is used deliberately to indicate that there is not a unique solution. The experience of the reverse engineer and the

particular objectives established in advance for a given project will influence the boundary definition. Figure 25 shows the context-exploration stage of the reverse engineering process. The orange shape encompasses the target system and its context, and indicates the region of focus for this stage of the process. Note that the system boundary, the definition of which is the outcome of this stage is different from (and contained inside) target system's context that is the region of focus of this stage.

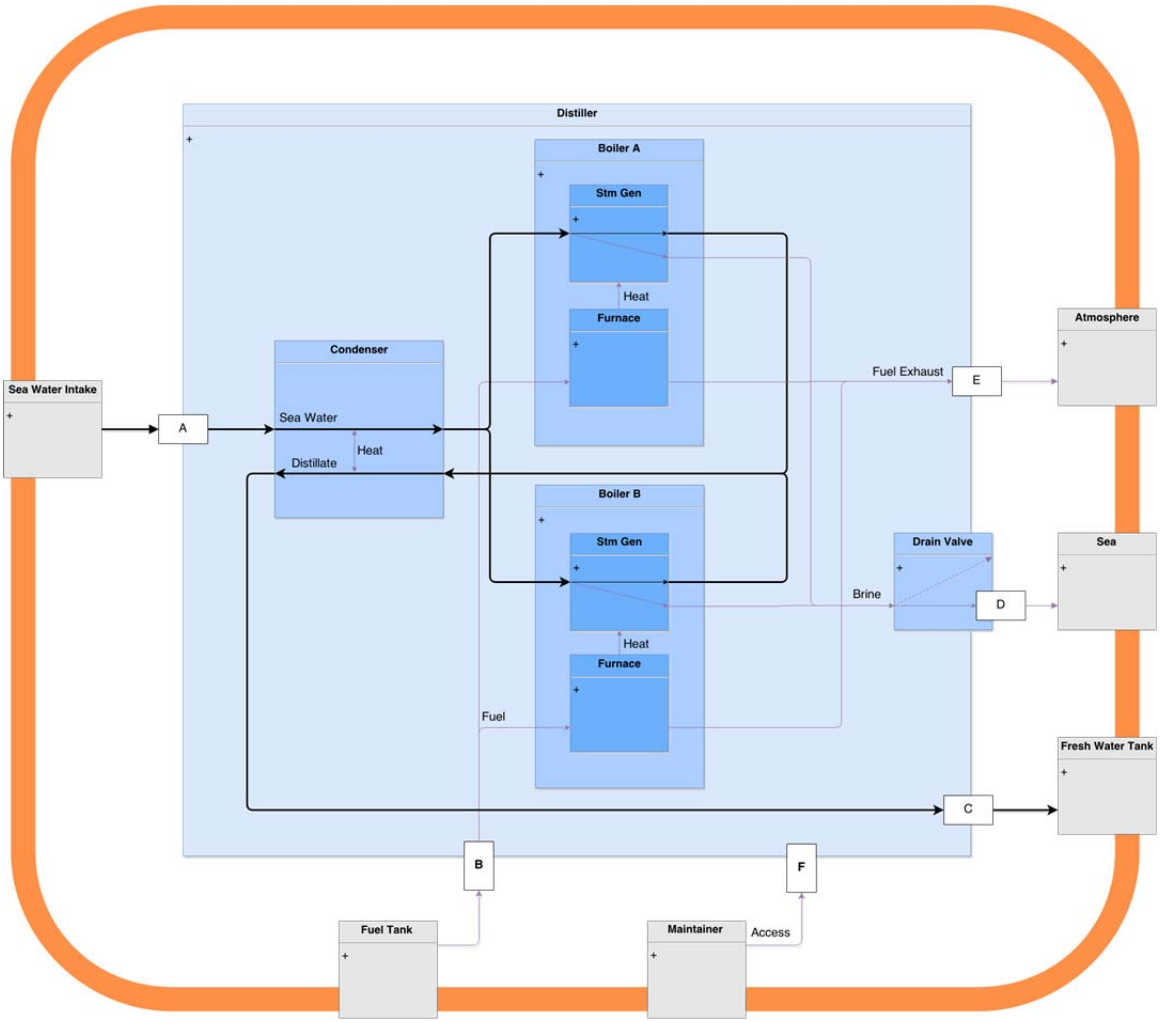

Figure 25. Context-Exploration stage of the reverse engineering process

The situation shown is the same as in Figure 19, but the depiction of the attention-area (the target system's context) has been streamlined to an orange "ring" to reduce clutter and improve preattentive processing. The depiction of the attention-area as a ring also emphasizes the iterative and penetrating nature of the reverse engineering process in figures 26 through 29.

2. Function-Discovery Stage: Identify All Functions

This stage begins after the boundary has been defined. The characterization of the system context achieved in the previous stage should guide the reverse engineer's search for system functions. The most obvious—and generally preferred—way to do this is by operating the system. Operation involves more than powering a system up and watching it run. Operation should cover the operational environments and use cases that were factored in the original design of the system. This may be very difficult to attain in practice, as the reverse engineer has no direct access to the mind of the OEM. Nevertheless, insofar as a finite set of use-cases were considered during the system design, there exists one fully correct answer to the question of *what are all the system functions*. It may be clearer to refer to functions as being of two types: *interactions* and *flows*. Thus, a handle, a button, and a trigger are loci for functional interaction. Whereas an electric wire, a steam pipe, or a boundary between two sides of a heat exchanger are loci for functional flows. This stage ends when the reverse engineer is satisfied that the answer has been reached: all functions have been identified. Figure 26 shows the function-discovery stage of the reverse engineering process. The orange shape—now smaller—indicates a new region of focus that encompasses the target system's interconnections with the other systems in its context. In other words, in this stage the reverse engineer looks at the target system's functions.

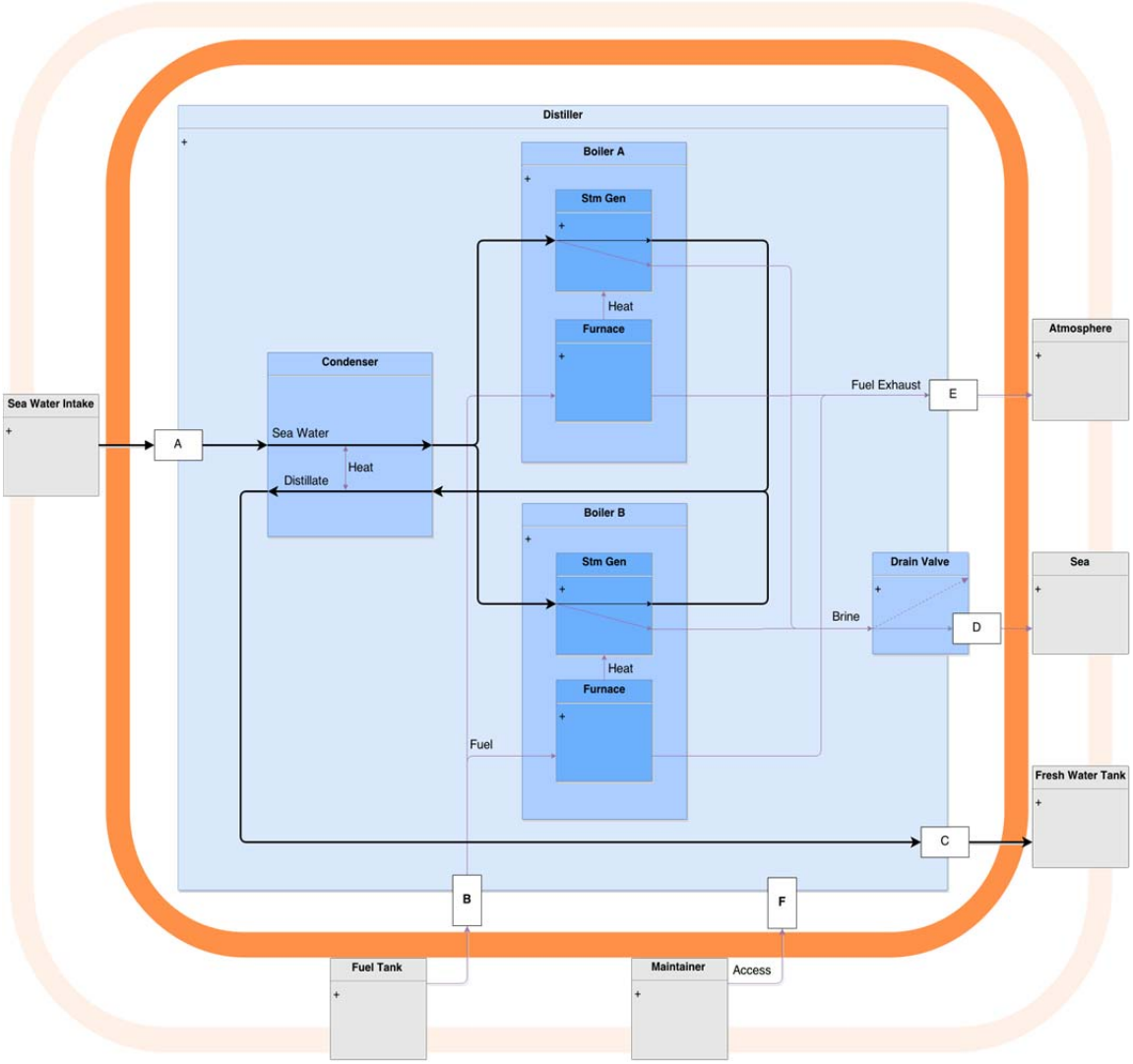

Figure 26. Function-discovery stage of the reverse engineering process

The orange ring highlights the attention-area, in this case: functions/interactions/flow exchanges between the target system and the context systems. The faded ring shows the preceding attention area in order to emphasize the iterative and penetrating nature of the reverse engineering process

3. Interface-Allocation Stage: Allocate Functions to Physical Interfaces

This stage begins after the reverse engineer decides that all functions of interest have been identified. The phrase "of interest" is used intentionally to highlight the reverse engineer's discretion in ignoring some functions. This discretionary exclusion may be applied to functions that are already well understood, or that are otherwise outside the scope of the project as defined in advance or due to constraints in time or other resources.

The objective of this stage is to learn the external mechanisms or physical interfaces by which each function of concern is accomplished. The stage ends when each function has been allocated. Figure 27 shows the interface-allocation stage of the reverse engineering process. The orange shape that shows the region of focus has contracted to correspond with its focus on the physical interfaces at the boundary of the system.

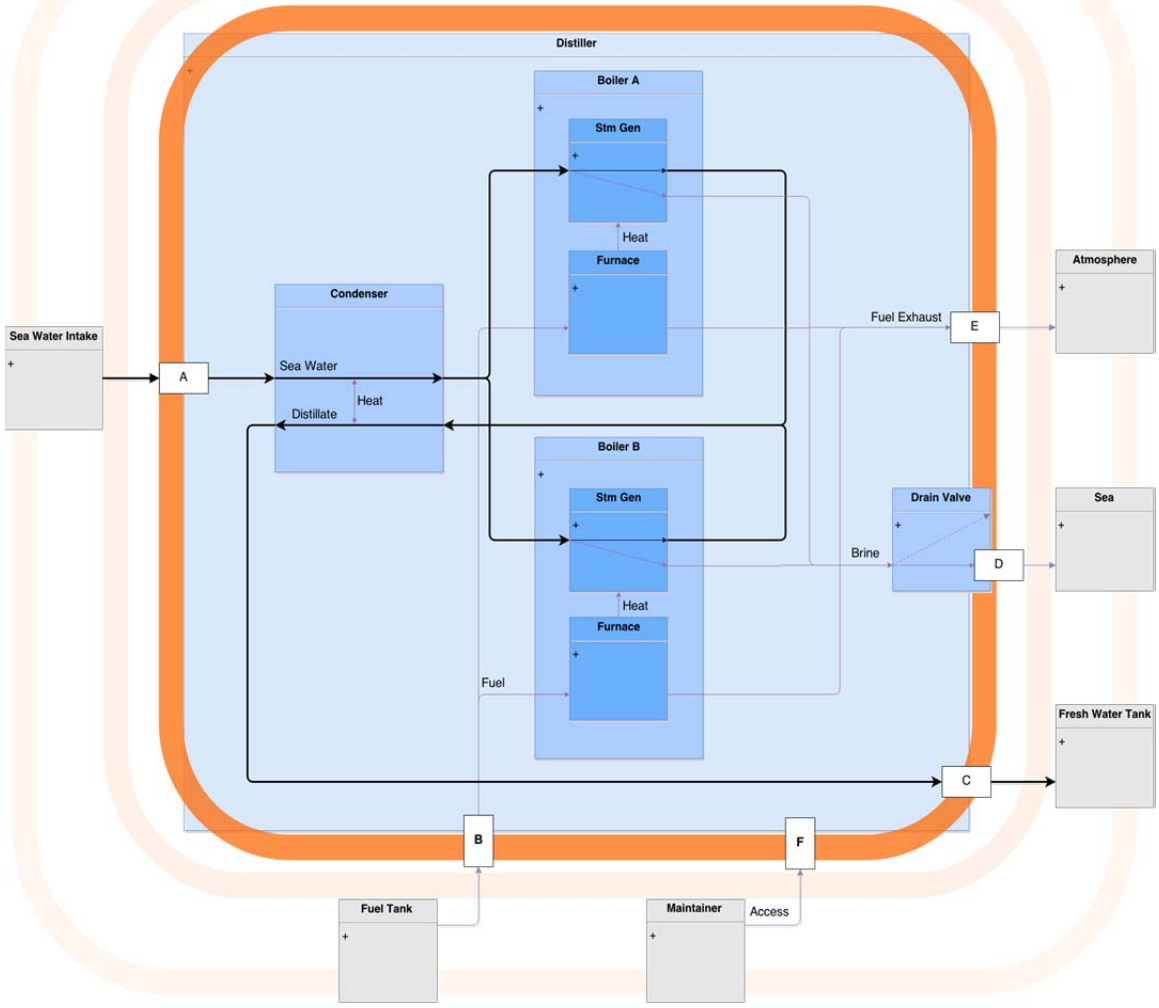

Figure 27. Interface-allocation stage of the reverse engineering process

The orange ring highlights the attention-area—in this case: the physical characteristics of the system boundary that enable the functions previously identified. The faded rings show the preceding attention area in order to emphasize the iterative and penetrating nature of the reverse engineering process

4. Boundary-Breach Stage: Breach the Boundary and Begin Tear-down

This stage begins when the reverse engineer decides that everything that can be learned from the outside has been learned. Once the boundary is breached the system will likely lose its capability to perform some or all of its functions. With less than full functionality, the reverse engineer must resort to cues rather than direct observation as a means to determine functional allocation. For this reason, breach and tear-down should begin only after the system-level functional allocation is believed to be complete. Figure 28 shows the boundary-breach stage of the reverse engineering process: the orange shape has contracted to just inside the physical boundary. This is meant to indicate that the focus of the reverse engineering effort is upon the physical system boundary as an obstacle to be overcome.

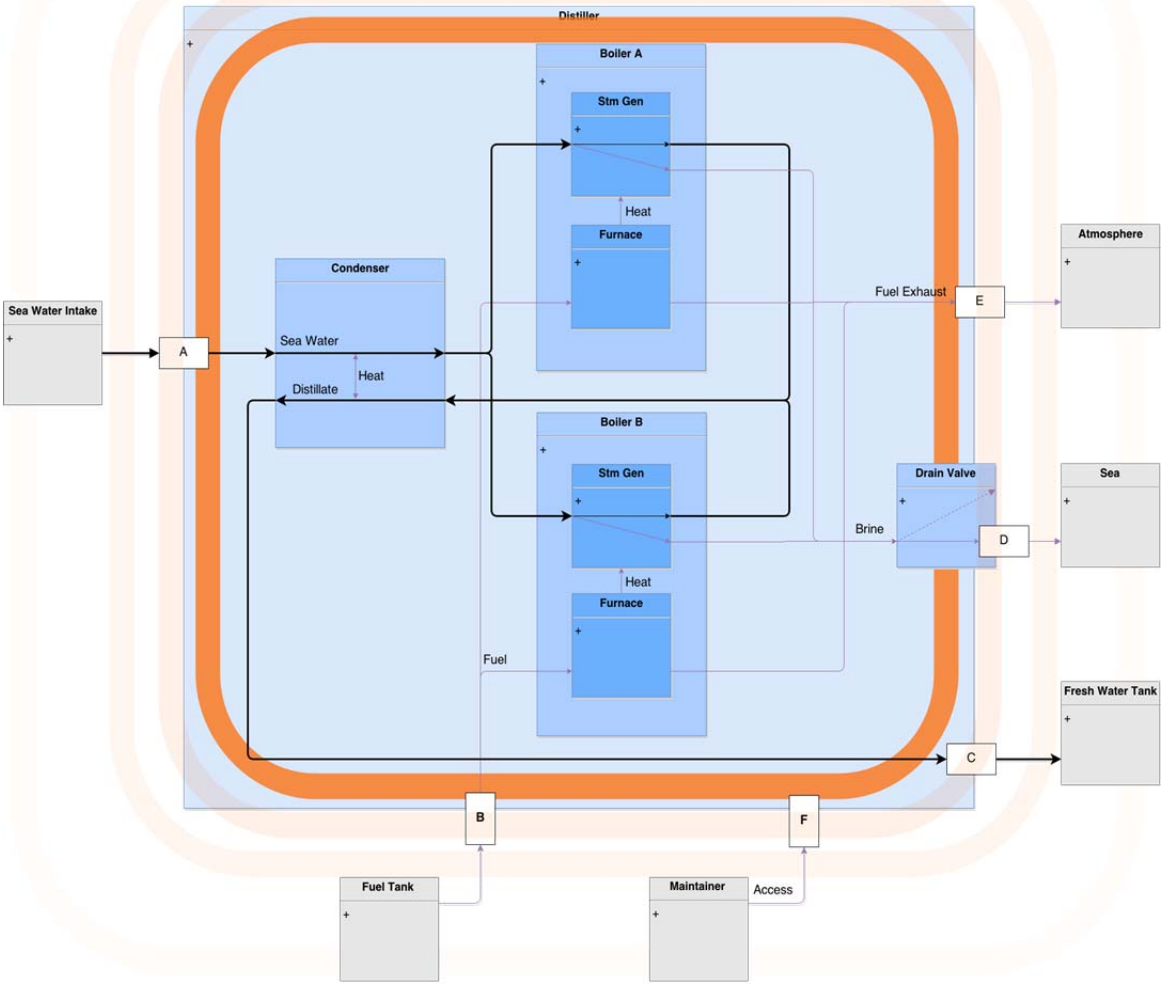

Figure 28. Boundary-breach stage of the reverse engineering process

The orange ring highlights the attention-area—in this case: the system boundary is inspected for a way in (aka a point of entry). The faded rings show the preceding attention area in order to emphasize the iterative and penetrating nature of the reverse engineering process

5. Partition—AKA Context-Exploration Stage Revisited: Define the Boundaries

Breaching the boundary does not necessarily lead straight into the tear-down. In fact, in an ideal scenario the system breach would preserve most system-level functions, and leave intact all the internal subsystems and their interconnections. If the system is of more than moderate complexity, it is likely that what the reverse engineer finds upon

breaching the system will not consist of a neat and clear arrangement of components with obvious functions. Thus, the necessary next step is to partition the internal componentry into modules.[35] A singled out module (subsystem, component, subcomponent, and so forth) now becomes the new "target system," while the rest of the system becomes the context (this may happen sequentially or in parallel for each module). In other words, the partition is in fact a second iteration of the context-exploration stage: Define system boundary. Figure 29 shows the context-exploration stage/second iteration of the reverse engineering process: the orange shape is now focused on the characterization of the internal context relative to the new target (sub)system.

[35] See heuristics for partitioning in Maier and Rechtin, and heuristics for modularization discussed in Chapter II of this dissertation.

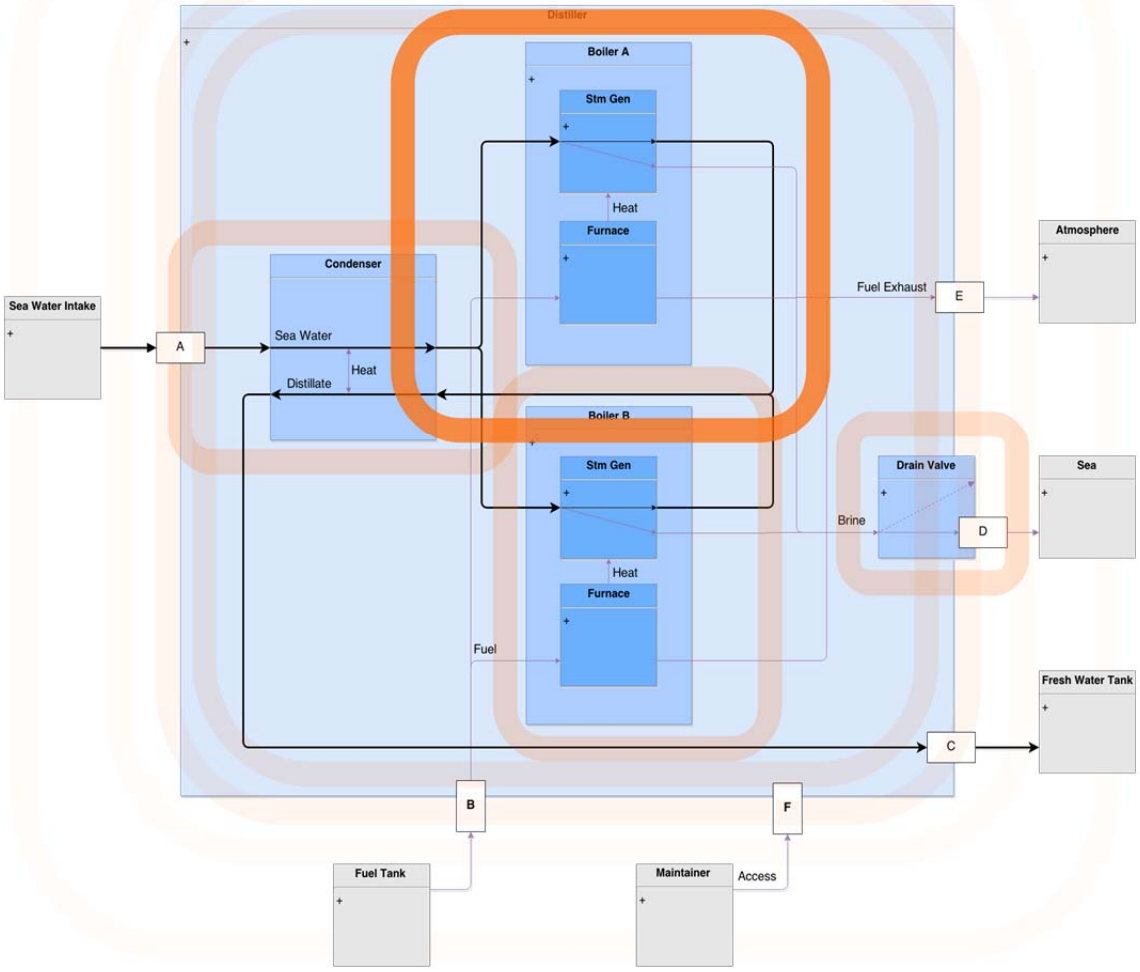

Figure 29. Context-exploration stage (at the next level of structural decomposition)

The orange rings highlight the region of attention/action—in this case: the boundary and context (of the internal subsystems) The faded rings show the preceding attention area in order to emphasize the iterative and penetrating nature of the reverse engineering process

C. CONCLUSION

In the previous chapter, a visual system model was introduced. Its purpose was to serve as a map on which we may chart a variety of shifts in attention and action that we believe occur pursuant to reverse engineering. In this chapter, the various actions and shifts in attention have been condensed into a more succinct picture. This showed reverse engineering to be a process of progressively inward focus comprised of successive

iterations of four stages: define the boundary, identify the functions, allocate the functions to interfaces, and breach the boundary.[36] This four-stage reverse engineering process is summarized in Figure 30. Each stage calls for certain actions (mental or physical) and from each stage the reverse engineer expects to obtain certain types of information. A transition between stages implies that all relevant information available from the preceding stage has been discovered and incorporated into the reverse engineer's working model. The boundary breach and tear-down provide no new information but serve as a gateway to revisiting the context-exploration stage at the next deeper level of structural hierarchy.

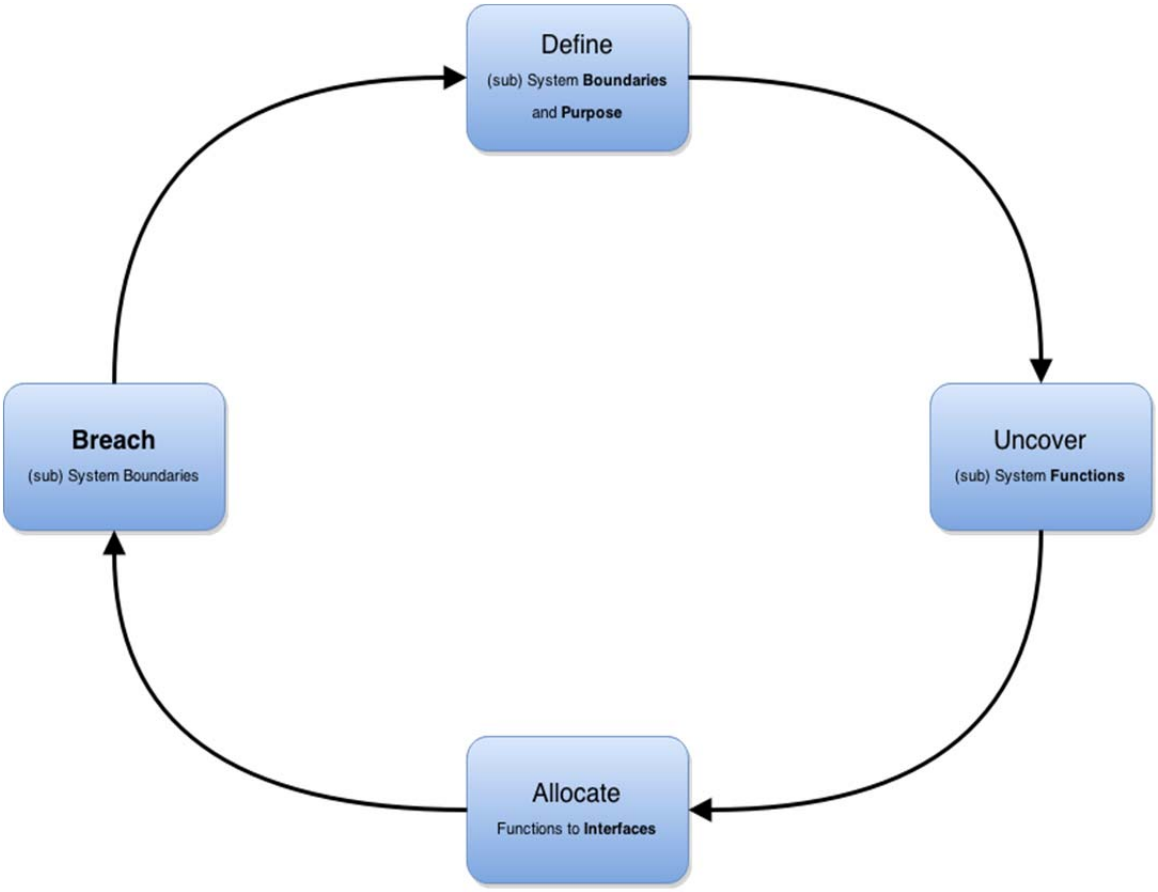

Figure 30. A summary of the reverse engineering process model

[36] **Boundary->Function->Interfaces->Breach** is a process of *stepwise exposure of design* bearing similarities to the model based systems engineering process referred to as "the onion model" described by (Long and Scott 2011).

VI. IMPLICATIONS AND PREDICTIONS

> The inseparability of knowledge and its practical application is in fact the distinguishing characteristic of engineering.
>
> —Walter G. Vincenti, *What Engineers Know and How They Know It*

A. INTRODUCTION

A problem domain is a context in which similar or related problem-solving activities take place. In a sense, it is like a landscape. Mathematics is an example. Whenever we do an addition or solve a differential equation, we traverse some portion of that landscape. Some traverses are easy while others are impossible. As with a physical landscape, each problem domain has some regions that are straightforward and risk-free, and others that are treacherous, like bogs or ravines.[37] When preparing to traverse difficult terrain, we hope there will be road signs and bridges to help us reach our destination. A bridge can carry us across some section of particularly treacherous terrain. A road sign can warn us of troubles ahead, perhaps suggests a better direction. Heuristics are like the road signs and bridges of a problem domain.

One of the stated goals of this research is to find heuristics that are useful in the problem domain of reverse engineering. The terrain analogy suggests a method for doing this. One method for finding bridges and road signs is to drive around until one stumbles into them. A better approach is to use a map in order to spot the difficult terrain in advance. The road signs and bridges, if there are any to be found, are likely to be in the vicinity of difficult terrain. This analogy is equivalent to the earlier explanation that heuristics work when the structure of the technique matches the structure of the problem.

The product of the preceding five chapters is precisely a map of reverse engineering problem domain. In this chapter, that map will be used to locate potentially difficult terrain. Later on this information may be used to search for "bridges and road

[37] A region is not the same as a typical problem, but it may be a part of one. For example in arithmetic a typical problem is the calculation of a square root. A treacherous region may be that of *very large prime numbers*.

signs" (aka heuristics). Along the way, this will produce a number of finds—as is often the case with exploration—that will cause the original map to undergo some revisions.

B. METHODOLOGY

1. Modes of Failure

The reverse engineering model shown in the preceding chapter depicts an ideal process. All the information that could be learned from the system at each step of the way was in fact learned. But that is not necessarily the case in the real world. Things might go wrong. As mentioned, the reverse engineer may get stuck or derailed by the problem domain's equivalent to bogs and ravines. These potential problem regions can now be taken out of their metaphor, and given a more technical name: *modes of failure*.

Reverse engineering is an activity that centers on the discovery and transfer of information. A reverse engineering project is successful to the extent that it uncovers (or recovers) all relevant information from a physical system and transfers it to a working model. This dependence on the transfer of information points to three general types of modes of failure: (1) information may remain *undiscovered* in spite of the reverse engineer's efforts, (2) "information" that is discovered may be *erroneous or false*, and (3) information may be *destroyed or lost* (unintentionally). Figure 31 provides a summary of these three types of modes of failure.

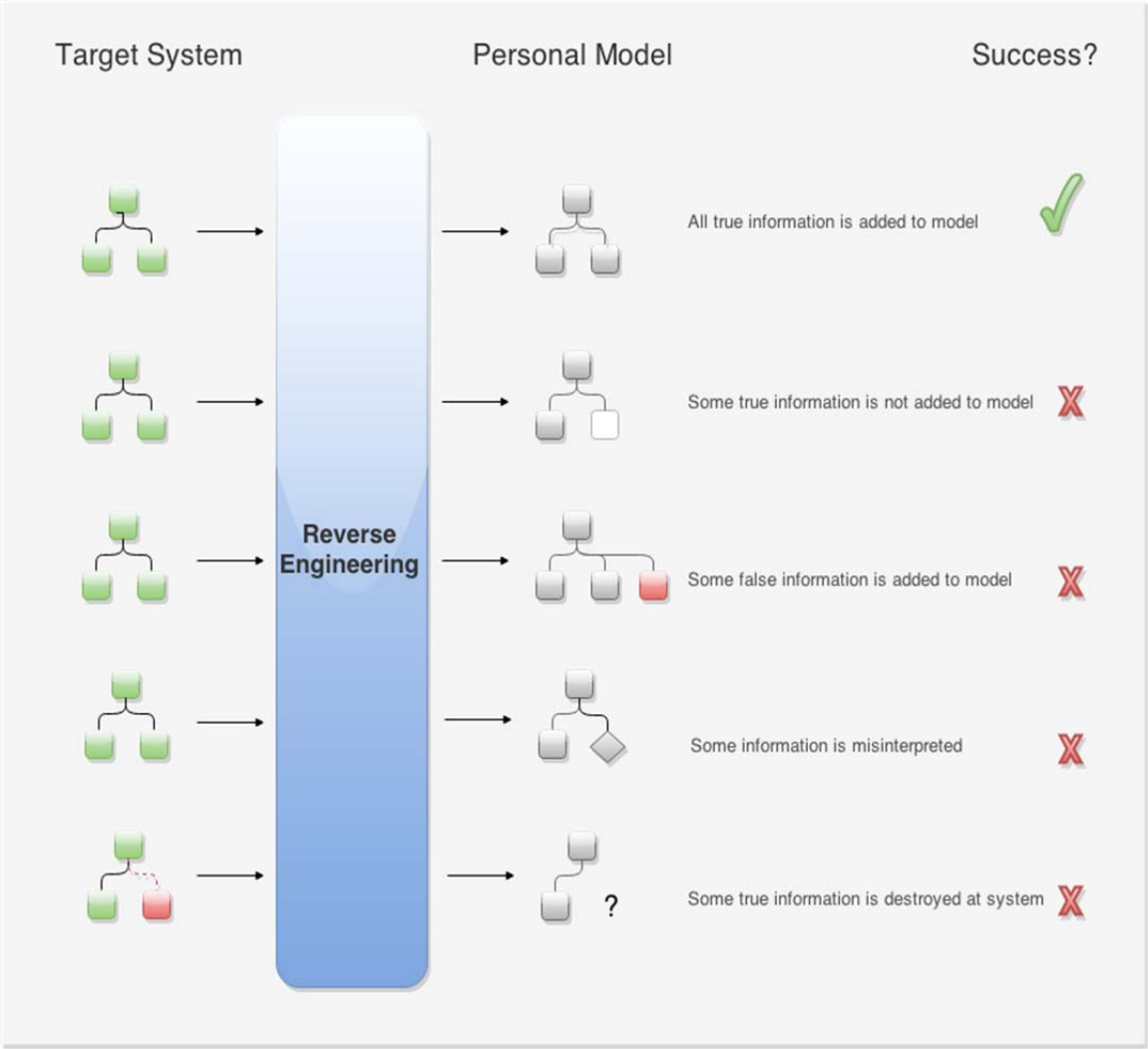

Figure 31. Four types of modes of failure

Shown, the essential types of modes of failure that can be incurred by a process whose success is defined by the capture of complete and accurate information about a system followed by its incorporation into a model

The reverse engineering process has four stages. The transition between each stage and the following one is critical. The first three transitions are predicated upon uncovering some particular type of information before moving on to the next stage. More precisely, they are predicated upon uncovering all the relevant information available at that stage. The fourth transition—from boundary-breach stage to the next iteration of the context-exploration stage—is predicated on the successful opening up of the system.

Each transition between stages can be a locus for one or more modes of failure as follows.

The transition from the context-exploration stage to two can occur before complete or accurate information has been gathered about the target system's context and purpose. The transition from function discovery to interface allocation can occur before complete or accurate information has been gathered about the target system's functions. The transition from interface allocation to boundary breach can occur before a complete or accurate functional allocation has been made. Finally, the transition from boundary breach to context exploration (second iteration) does not require a gathering of information, but carries the potential to impede recovery of or access to information, should components or interconnections become damaged or destroyed during the opening up of the system. All these modes of failure can result in a partial or complete failure of a reverse engineering project. Accordingly these may be termed "hard" modes of failure.

Reverse engineering is constrained by a budget, resources, and time. Thus, a scenario can arise where the information sought is ultimately obtained, and yet some mechanism has interfered with the process such that time or resources have been wasted. We may think of the mechanism responsible for this waste as a "soft" mode of failure.

In the following section the model developed in chapters V will used to ask and visualize the answer to a new question: *What could go wrong?* This will require a few modifications to the target system as well as the introduction of some new elements in the visual notation that was introduced in Chapter IV. These changes and updates will be introduced as they appear.

Modes of Failure From Context Exploration to Function Discovery: Errors in Ascertaining Purpose, Context and Boundary. Figure 32 shows the transition between the context exploration and function discovery stages of the reverse engineering process. The transition requires a complete and accurate characterization of the target system's purpose and context (complete knowledge of what the system is for and what other systems it interacts with). In this process there are two hard modes of failure: (1) The information gathered before moving on to the next stage may be incomplete because

some key aspect of the system's context has been overlooked, or (2) The information may be inaccurate because some erroneous "fact" about the system's context or purpose has been incorporated into the reverse engineer's working model.

Additionally, the system boundary is defined in the context-exploration stage. As discussed, the problem of boundary definition does not have a unique solution. Some definitions of the system boundary will result in reduced efficiency during the subsequent analysis (Maier and Rechtin 2000, 49). Inefficient boundary definition is a soft mode of failure.

To show the hard modes of failure an updated notation has been introduced in Figure 32. The color green was used in Chapter IV to indicate a feature of the target system that has been identified and incorporated into the reverse engineer's working model. Now a similar notation is employed, but with an expanded palette. The color red is introduced to indicate trouble. *Trouble* means that some of the information associated with that feature has been missed, misidentified, or destroyed.

Figure 32 shows "Seawater Intake," "Freshwater Tank," "Fuel Tank," and "Atmosphere" in green blocks. As in the last chapter, this indicates that prior to the transitions from context exploration to function discovery, the reverse engineer incorporated these systems' interaction with the distiller into a working model. However, "Sea" and "Maintainer" are shown with a red outline and blank fill color. The red outline signals trouble while the blank fill suggests the nature of the trouble. The systems in question have not been incorporated into the reverse engineer's working model.

Also, a new block labeled "Cooling Water Tank" has appeared in the diagram. A red outline is once again used to convey trouble. In this case, the block is also filed in red. This indicates that the cooling water tank has been incorporated (erroneously) in the reverse engineer's working model of the distiller. In reality there is no connection between the distiller and the cooling water tank.

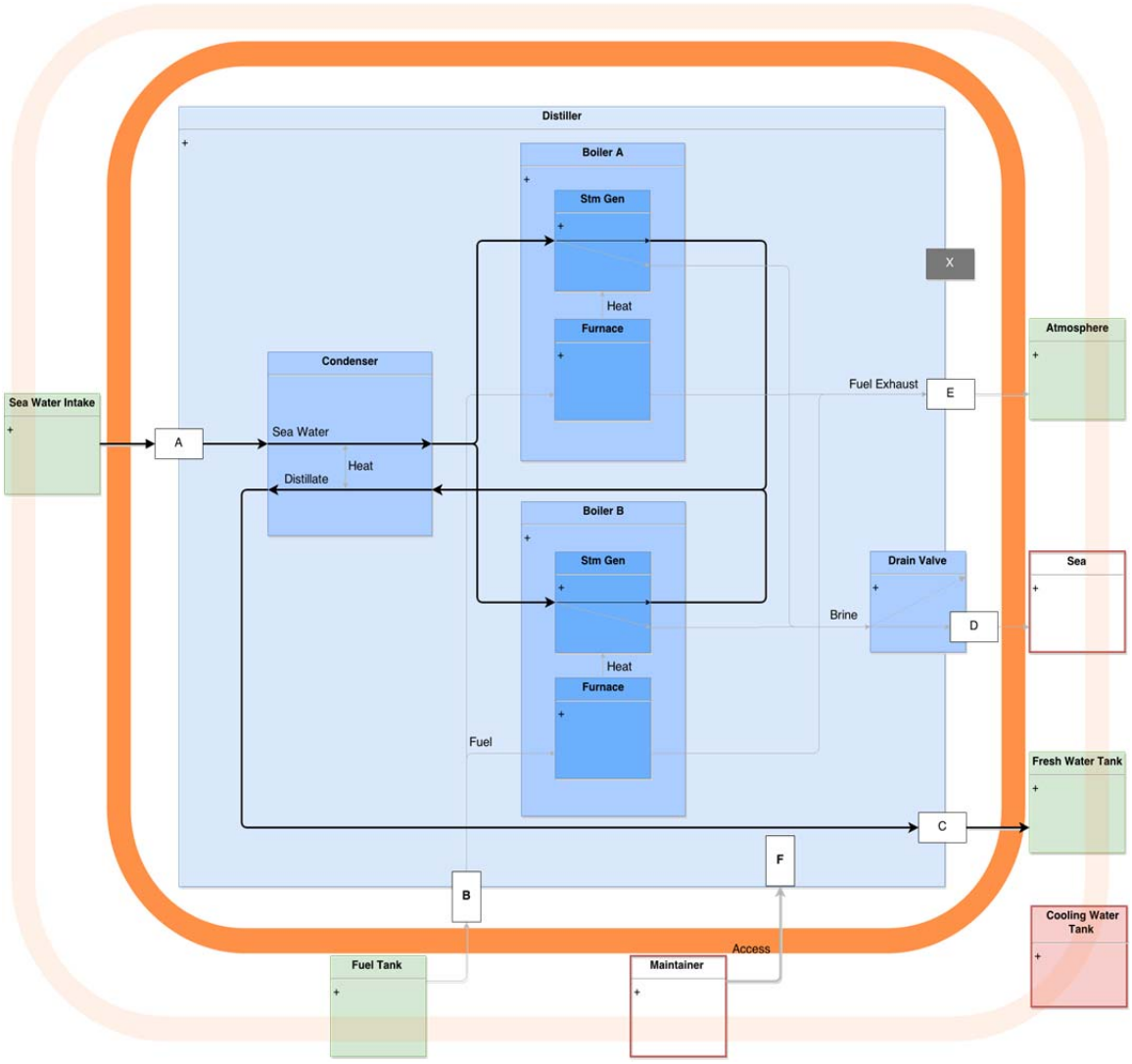

Figure 32. Context-exploration to function-discovery transition

Shown (in red) possible modes of failure

a. Modes of Failure from Function Discovery to Interface Allocation: Missed and Made-Up Functions

Figure 33 shows the transition between the function-discovery and interface-allocation stages of the reverse engineering process. The transition presupposes a complete and accurate characterization of the target system's functions. The system functions that have been correctly incorporated into the working model are shown: arrows that have turned green. The correct identification of a context system is closely

tied to the discovery of the function or interaction linking that context system to the target system.

Note that the target system in this chapter differs in some important ways from the one described in the previous chapter. Attribute F—"Maintenance Hatch"—which was formerly at the system boundary (and therefore easy to discover) has been moved to a new location just inside the boundary. This indicates that the attribute is not explicit at the system boundary. Perhaps it lies hidden under a thick layer of insulation. A hidden or inconspicuous attribute may account for a reverse engineer's failure to fully characterize the context. For example, in this case the reverse engineer has thus far failed to learn that the maintenance person has a place in the model as part of the context, and that there is an interface designed to support that interaction. Note that the presence or absence of a maintenance access hole is interface allocation information, while the identification of *maintenance person* and the function *allows access for maintenance* belong to the context-exploration and function-discovery stages respectively. The implication is that information can travel "upstream" across stages. This feedback process will be discussed in more detail later.

Another physical difference between this target system and the one in the previous chapter is signaled through a new notation element: a small rectangle labeled X and depicted as similar in shape to an attribute, but in "negative" color. This notation is introduced for quasi-interfaces: prominent physical features that appear to bear some function but are in fact non-functional.[38] In the distiller, quasi-interface X gives the impression of being an interface for some type of test equipment. In reality, X could be perhaps a byproduct of the system's assembly process that was accidentally (or for convenience) never removed. This error causes new feedback: "Test Equipment" (context system) and "Testing" (a function) are incorporated erroneously into the reverse engineer's working model. They are shown in red because no such connection exists.

Finally, the erroneously inferred context interaction with a cooling water tank leads to a fruitless search for a non-existing interface that enables a non-existent

[38] This notion was introduced in Chapter III where the example used was the tailfins often added to the rear ends of vehicles in the 1950's 60's. This is also discussed in greater detail in Appendix F.

interaction. This is depicted by the red arrow from the cooling water tank that fails to connect to the system boundary.

There are other related and similar modes of failure that have not been shown in this figure because they do not fit well with the distiller example. For instance, a flow or function may exist and be identified correctly, and yet the nature of the flow as incorporated into the reverse engineer's working model is erroneous. For example, two components of a remote controlled system may be correctly inferred to communicate with each other, however the "Communication" function could be added to the working model as "Radio Transmission," when in fact it is infrared transmission.

There are different possible causes for this mode of failure. One or more functions may have never been activated because a critical context system was not present during the first and function-discovery stage of the reverse engineering process. For example, if a radio controlled vehicle is the target system, and the radio controller is not available. Or a television is the target system, and no television signal is available. Alternatively, the missed function may be present but invisible to the senses and instruments that the reverse engineer has at his disposal. For example, a navigation system might transmit GPS information across a specific radio frequency for a few seconds of every hour. Unless the reverse engineer is looking with the right instrumentation, tuned to the correct frequency, and at the right time, the transmission and therefore the entire function/interaction will be completely invisible.

Another possible mechanism for this mode of failure: time-scale-mismatch. That is, a function or interaction may be present and visible (instrument-wise) and yet escape detection because it occurs too fast, or too slow for the reverse engineer to perceive.

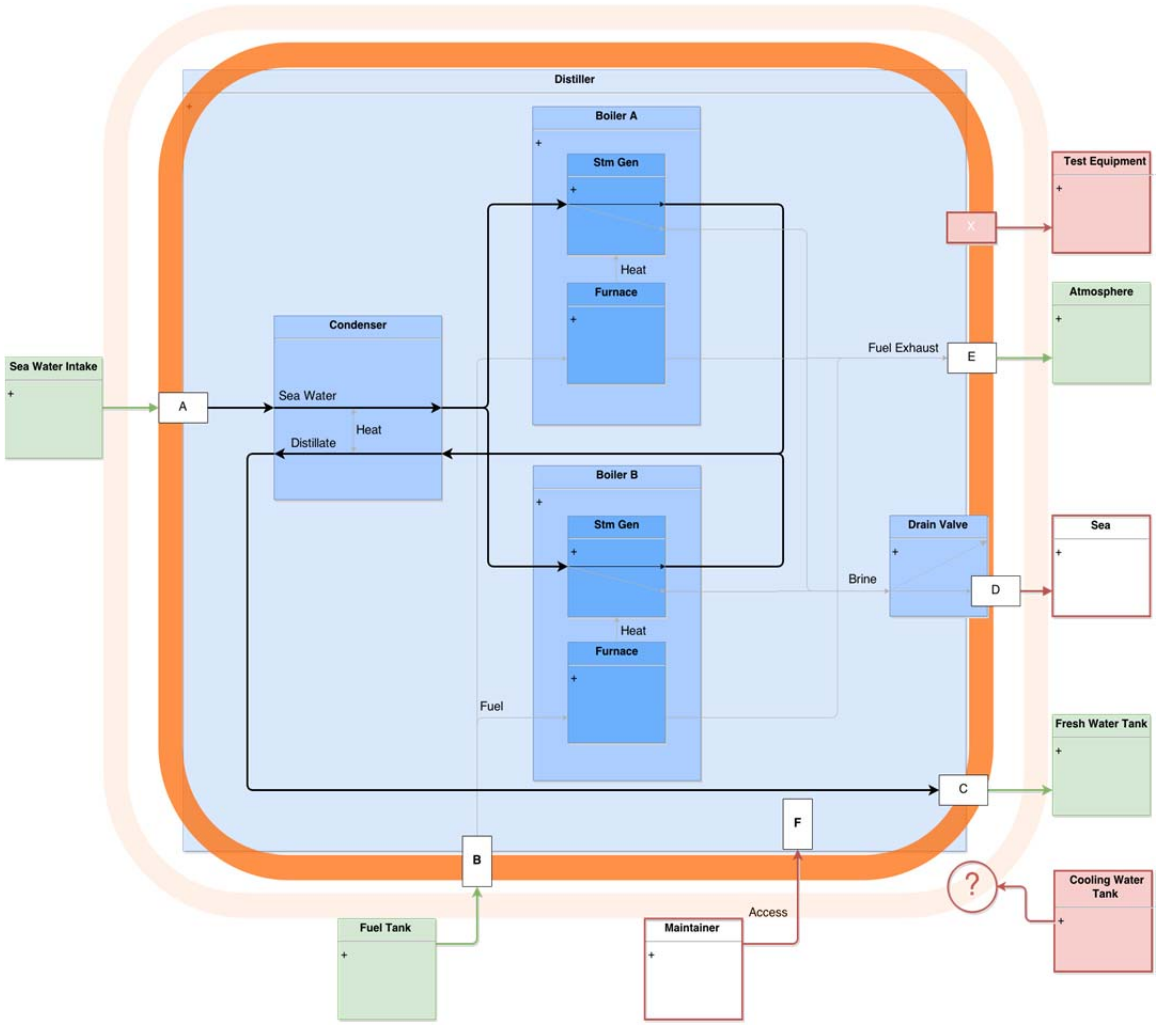

Figure 33. Function discovery to interface allocation transition

Shown (in red) possible modes of failure

b. Modes of Failure from Interface Allocation to Boundary Breach: Overlooked and Non-functional "Interfaces"

Figure 34 shows the transition between the interface-allocation and boundary-breach stages of the systems engineering process. The transition is predicated upon a complete and accurate functional allocation.

In the example, the fact that Attribute F remains undiscovered has the repercussion that the system function "Provides Access for Maintainer" has not yet been discovered. Given that the objective of the boundary-breach stage is to gain access to the system's internal components, this is a critical piece of misinformation. It may be more

realistic to assume that a thorough search for a non-destructive means to breach the system would eventually turn up the access hatch. Regardless, for illustrative purposes this is not the case.

The last transition introduced the concept of feedback: discoveries can prompt a revision of the information that was obtained—or failed to be obtained—earlier in the process. Two new feedback scenarios appear now. Each case has a different impact on the status of the reverse engineering project.

First, as might be expected the close inspection of the system boundary in search of a point of entry may bring the reverse engineer to discover subtle attributes that were missed earlier. These discoveries may in turn lead him to infer the existence of functions he had not considered yet. In this case, the inspection of the system boundary turns up a valve. This discovery is enough to cause a revision of the working model. Clearly, the valve is an interface that must be accounted for. Furthermore, the flow from the valve can be traced and the working model is revised to incorporate the SEA into the context. However, the reverse engineer then goes on to ascribe an erroneous function to the valve. Perhaps he concludes that it is a pressure relief valve instead of a brine discharge. Thus, the block representing the valve goes from red border/blank fill (should be but is not part of the reverse engineer's working model), to red border/red fill (is a part of the reverse engineer's working model—but not with ascribed role).

The second case of feedback is slightly different. The thorough inspection of the physical system boundary has made it clear that there is no interface between the target system and the cooling water system. The erroneously presumed interaction and context system are therefore eliminated from the reverse engineer's working model. This is depicted by the change in the colors of the affected block as follows: green border/blank fill (component is correctly no longer part of the working model). A new type of arrow (curved dotted line) pointing from a location near the system boundary to the cooling water tank is used to depict the role of feedback in this development. The direction of the arrow indicates the direction of the inference, and its distinct shape is meant to preclude it from being confused with a system interaction. The information gained by inspecting the system boundary resulted in updated knowledge regarding the system's interaction with

the cold water system (i.e., there was none). In this case feedback has an error-correcting or stabilizing effect.

But, not all previous errors are fixed by feedback. One piece of misinformation from earlier stages still lingers, with consequences. The discovery—during interface allocation—of the Quasi-interface X continues to be associated with an erroneous function and context.

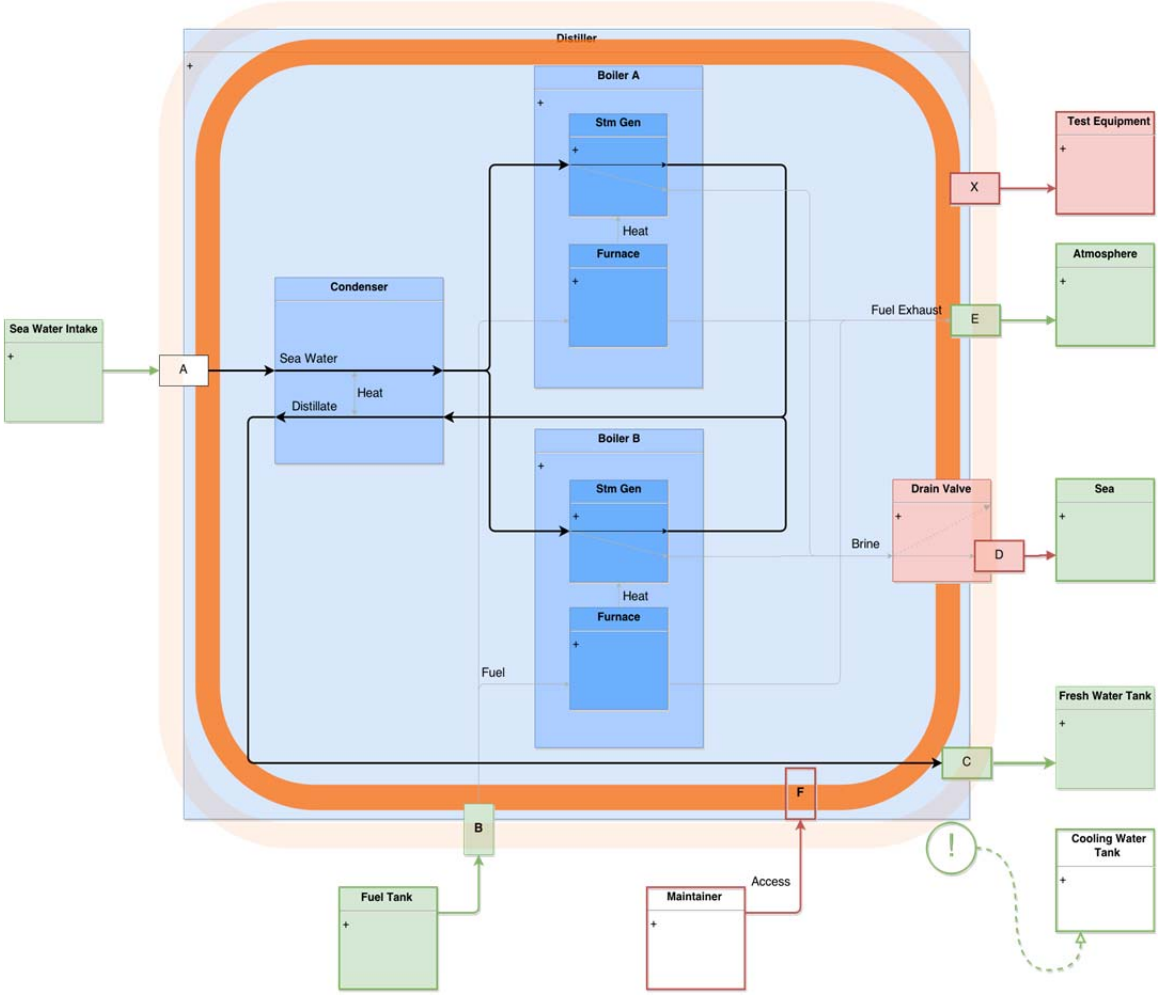

Figure 34. Interface-allocation stage to boundary-breach stage transition

Shown (in red) possible modes of failure

c. Modes of Failure from Boundary Breach to Context Exploration: Breaking Things before We Understand Them

Finally, the transition from boundary breach to the context-exploration stage (next iteration) does not introduce new information, but it does carry the potential for destroying information before it is gained into the working model, by destroying components or interconnections. If information is destroyed in the target system it will be impossible to retrieve it later, unless the process involves multiple specimens. The annotations on the system model show several things have taken place during this transition.

First, the system breach is characterized by the reverse engineer's selection of a point of entry. Generally, something has to be removed from the system boundary in order to gain access the majority of system components. In smaller target systems, it is common for the entire boundary (a protective casing, shell, or skin) to be removed. This could be indicated by using dashed lines for the removed system boundary.

For larger systems, the point of entry is probably going to fall short of the entire system boundary, and it will not necessarily be unique. This means that determining the best (most efficient and effective) point of entry offers another possible soft mode of failure: the selection and pursuit of a non-optimal point of entry: one that results in slower or limited access.

The reverse engineer working on the distiller has a bigger problem. Failure to locate the maintenance hatch has led to the selection of the seawater intake as a point of entry (the ship is in dry-dock!). The consequence is that the seawater flow through the system has been destroyed or disrupted. This is a problem because the system is not available for further operational testing while it is in this condition. If the valve or piping were damaged in the process, the situation is irreversible and from this point onward, any conclusions about system function will have to be drawn from inference rather than direct observation.

This no-flow-possible condition is indicated in the system model by the dashed line. The accidental or intentional breaking of any internal component or interconnection

prior to or during the reverse engineering process can be shown via a dashed line. The color of the dashed line can be used to indicate whether the destruction was justified. In some instances—for example, when non-reversible assembly procedures have been used—controlled destruction will be a necessary element of the system breach sooner or later. Such inevitable uses of destructive steps may be indicated by dashed green lines or outlines. However, in many instances an assembly process is not irreversible, but merely appears to be. For example, a simple concealed push-tab used in conjunction with tight fitting components poses a kind of physical puzzle. In these cases the requirement for locating a point of entry challenges the imagination and experience of the reverse engineer. If the puzzle is not solved, it will eventually drive the reverse engineer to use destructive force to gain access. Such avoidable destruction constitutes a mode of failure, and may be indicated by a dashed red line.

On the other hand, it is also conceivable that non-destructive breaching of the system boundary is sometimes not only possible, but that there even exist several options. This flexibility appears to be a good thing. However, it introduces the possibility that a less than optimal point of entry is selected. A less than optimal point of entry may be one that results in unnecessary consumption of resources (for example by yielding only slow, or restricted access) or perhaps one that is riskier in terms the potential for accidental destruction or disruption of internal objects and functions. In other words, the boundary-breach to context-exploration transition introduces an additional soft mode of failure: inefficient or unnecessarily dangerous breaching.

In the distiller example, internal inspection coupled with feedback has resulted in a revised and correct view of the interaction between the distiller and maintainer (alas, too late to be taken advantage of during the boundary-breach stage). The internal inspection has also failed to reveal any internal connections associated with the quasi-interface. Therefore, the erroneously held interface with test equipment is removed from the reverse engineer's working model.

Also, internal inspection (finally) reveals the nature of the valve to be a brine discharge valve; this is shown by changing the valve block color to green. That the reverse engineer may have failed to know this up to this point is of course unlikely.

Nevertheless, it illustrates an important point: internal inspection of the system will probably correct misinformation held prior to breaching the system.

Finally, Figure 35 also shows incorrect partitioning. While it would be ideal that each subsystem be clearly delineated, self-contained, and minimally connected with other subsystems, we can hardly expect that to be the case. Space constraints and other design factors are likely to cause subsystems to be tightly packed, overlapping, and entangled. In light of this, Figure 35 shows another possible "soft" mode of failure: incorrect partitioning of the subsystems. This is not necessarily a big problem. After all, system partitioning (like system definition) is subjective. Correct partitioning carries the implicit objectives of facilitating the efficient allocation of expertise and resources, and the reduction of flows/functions that are in need for analysis.[39] Incorrect partitioning by definition introduces ineffectiveness and inefficiency into the process.

In this case the orange focus area shows that the reverse engineer believes that "the boiler" should be treated as a single unit. But incorrect partitioning does not imply a false model of the system, merely a less efficient model. In the context of reverse engineering, the recognition that there are two systems that mirror each other is valuable because it improves the sample situation. It allows the simultaneous tear-down of one component while keeping the other in "operational" condition.

Note that the soft mode of failure "incorrect partitioning" has already been encountered. That is because the process has gone full circle and entered its second iteration. The subsystem partition "Boiler" is an outcome of the reverse engineer's assessment of the context found inside the distiller. This partition will be followed by the characterization of the boiler's functions, then the allocation of these functions to objects and attributes. Ultimately it will lead to breaching of the boiler.

[39] Number two of Maier and Rechtin's "most widely applicable heuristics" is "In partitioning, choose the elements so that they are as independent as possible; that is, elements with low external complexity and high internal complexity. " (Maier and Rechtin 2000, 49)

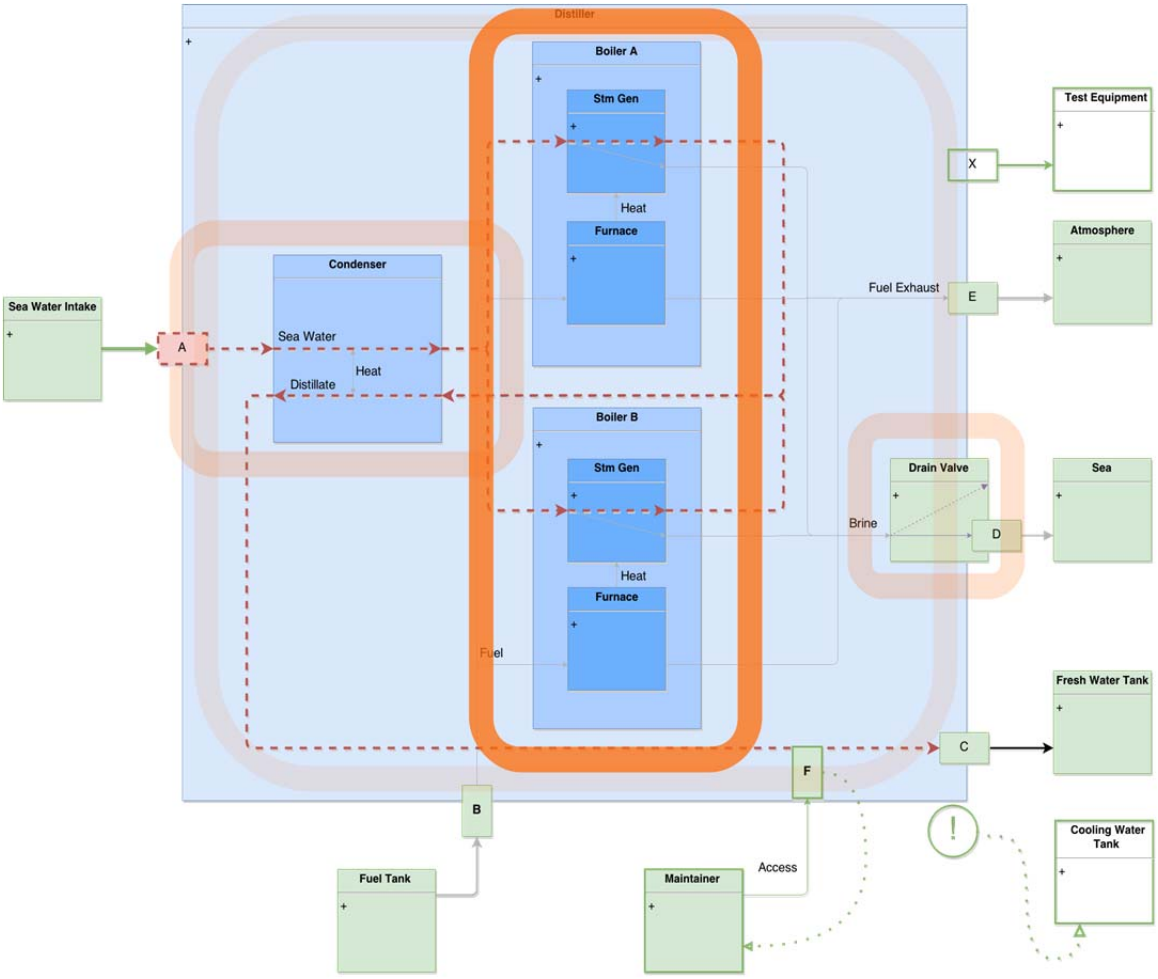

Figure 35. Boundary-breach to context-exploration stage transition

Shown (in red) possible modes of failure

C. CONCLUSION

When we claim to understand some aspect of the world, what we are saying is in fact that we possess a model that can account for what we see—and for what we will see. In that sense, modeling is synonymous with the scientific method. Figure 36 shows this process.

A scientific model is descriptive. In other words, if the data was judiciously selected, precisely measured, and rigorously analyzed, then the model will describe the real world process. This may allow some prediction and inference. In engineering a model can sometimes be more than descriptive: it may be prescriptive. In other words, a

scientific model generally describes the world as it is (according to the best available analysis of the most recent data). But in engineering—which is a human activity—a model may describe a process not as it is, but as it should be. This is the case with the process model for reverse engineering.

Almost from the start, we acknowledged that the real-world process of reverse engineering will stray from the reverse engineering model suggested in Chapter IV. In fact, we take this very divergence between model and reality to be our object of study, as we suppose that therein we may search for heuristics. In this chapter we looked more closely at those possible points of divergence—we have called them modes of failure.

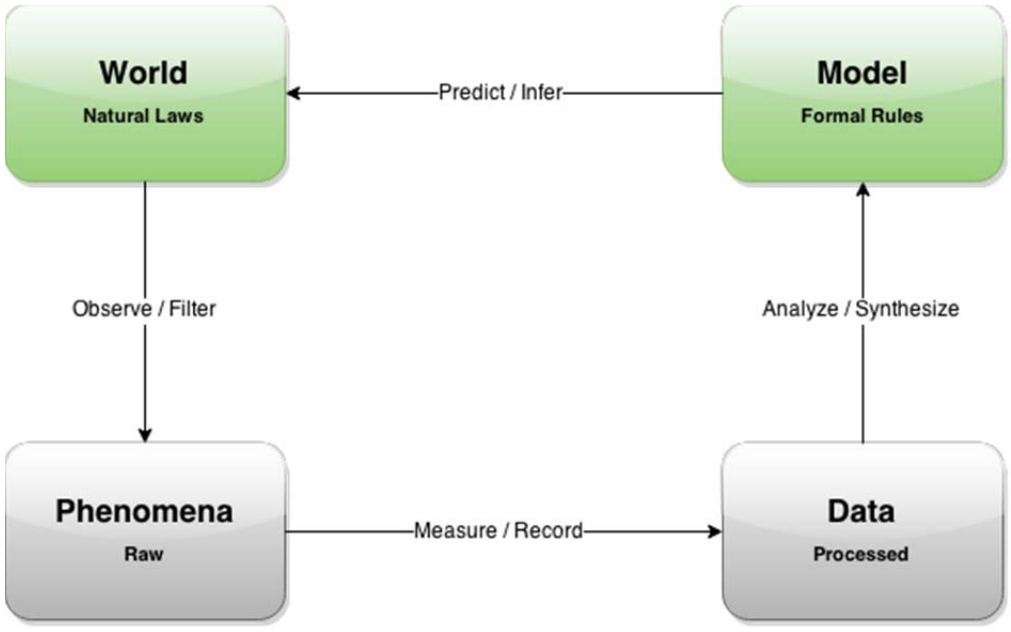

Figure 36. The role of modeling in science

Understanding some aspect of the world is synonymous with possessing a model that can account for phenomena (what we see). Modeling is synonymous with the scientific method.

Conclusions drawn from this chapter call for a revision of the process model presented in Figure 30. In that process, it was suggested that information obtained in one stage of reverse engineering will prime the search for specific, related information in the next stage. The discovery of a context system in the context-exploration stage primes the reverse engineer to search for the nature of the interaction, which in turn primes him for

the search for an attribute or object responsible, and so forth. That model it is now replaced by the model shown in Figure 37. It incorporates feedback. Information not only primes the search for the next stage, but it also informs and revises previous stages.

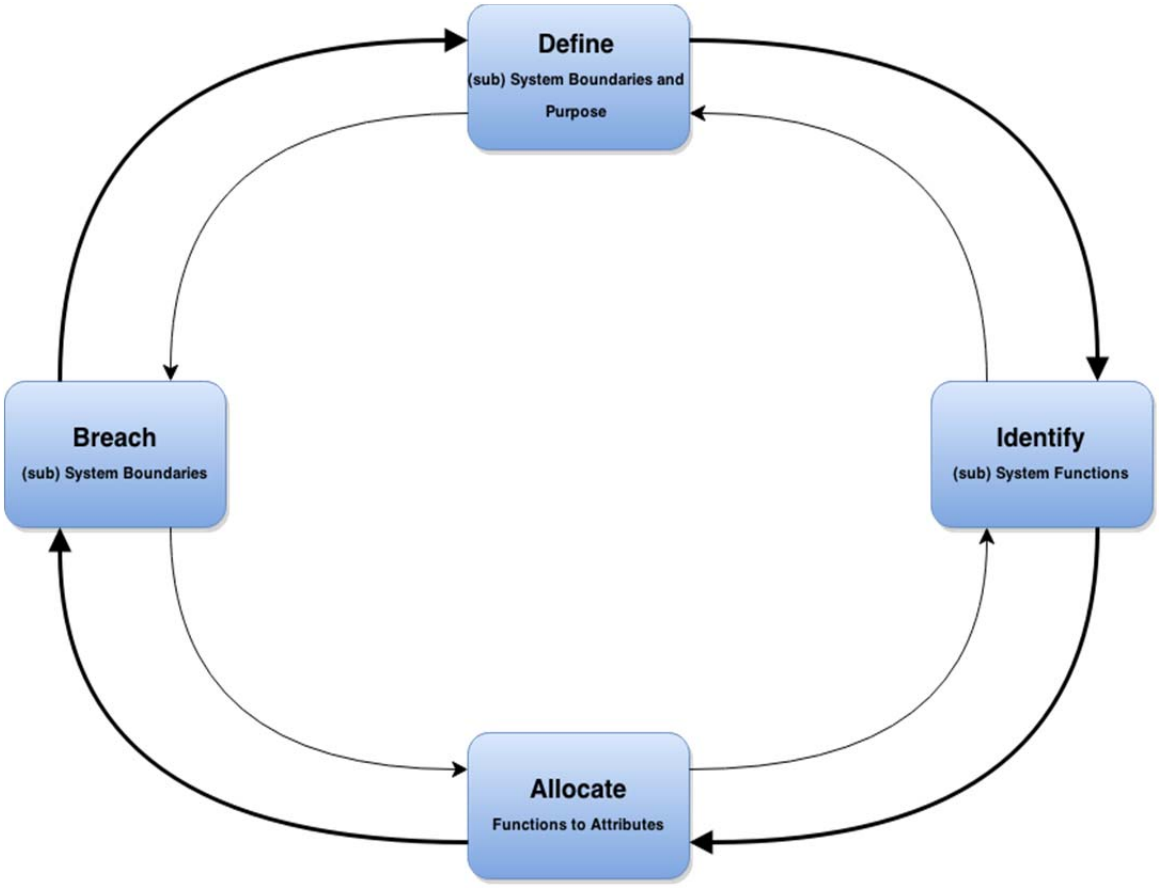

Figure 37. Updated reverse engineering process showing feedback

Shown, a summary of the reverse engineering process model modified by the heuristic: Let each stage feed back to the previous stage.

Another important conclusion: The iterative nature of the process and the connected nature of the system mean that when a failure mode occurs in one stage, the failure will propagate into other stages.

(1) Chapter Summary

In reverse engineering, a target system is like an encoded message. The reverse engineer is successful to the extent that she unpacks or decodes all the information contained in the message. Given this definition of success, there are three paths to failure: the information may stay *un-decoded*; it may be *erroneously decoded*, or it may be unintentionally *destroyed before decoding can take place*. In addition to these three "direct" paths to failure, there are indirect paths: actions that result in wasted resources, or that increases the probability of failure. In the previous chapter, reverse engineering was presented as consisting of four stages, iterated at each level of a progressive tear-down. In the present chapter, it has been proposed that it is at the transitions between stages that failures can occur (as the information must be transferred from one stage to the next). The mode whereby these failures can occur is specific to each transition. These have been termed **modes of failure.** The modes of failure where information may be left behind, erroneously incorporated, or accidentally destroyed are termed hard modes of failure. The ones where inefficiencies or unnecessary risks are introduced are termed soft modes of failure. All modes of failure are shown in Table 3.

Table 3. Modes of failure

Context Exploration to Function Discovery	1. Incomplete characterization of system context/purpose
	2. Inaccurate characterization of system context/purpose
	3. Inefficient definition of System boundary (Soft)
Function Discovery to Interface Allocation	4. Inaccurate characterization of system function (feed-forward from inaccurate context)
	5. Inaccurate characterization of system function (feedback from quasi-interface)
	6. Incomplete characterization of system function (hidden context)
	7. Incomplete characterization of system function (hidden object/attribute)

Interface Allocation to Boundary Breach	8. Inaccurate allocation of system functions (to a quasi-interface)
	9. Inaccurate allocation of system functions (to a real attribute)
	10. Incomplete allocation of system functions (due to hidden attribute)
Boundary Breach to Context Exploration	11. Unessential destruction of system objects/attributes
	12. Unessential disruption of subsystem functions/interconnections
	13.a Inefficient point of entry selection due to resources or time use (Soft)
	13.b Inefficient point of entry selection due to unnecessary risk introduction (Soft)

The reverse engineering process hinges on the discovery and utilization of information. This suggests modes of failure: circumstances when the information is either incompletely or inaccurately passed along between one stage of the process, and the next. Several of these modes of failure were later validated in the case studies described in Appendix B and Appendix C

The information above is summarized visually in Figure 38. Note that the occurrence of a hard mode of failure does not mean the reverse engineering project has failed, only that it has fallen short of its full potential with regards to the information that may have been obtained. Incurring a hard mode of failure will result in a corresponding shortcoming of the reverse engineering project only if the mode of failure remains in place until the end. However, the process of feedback will tend to correct errors before the end of the project .

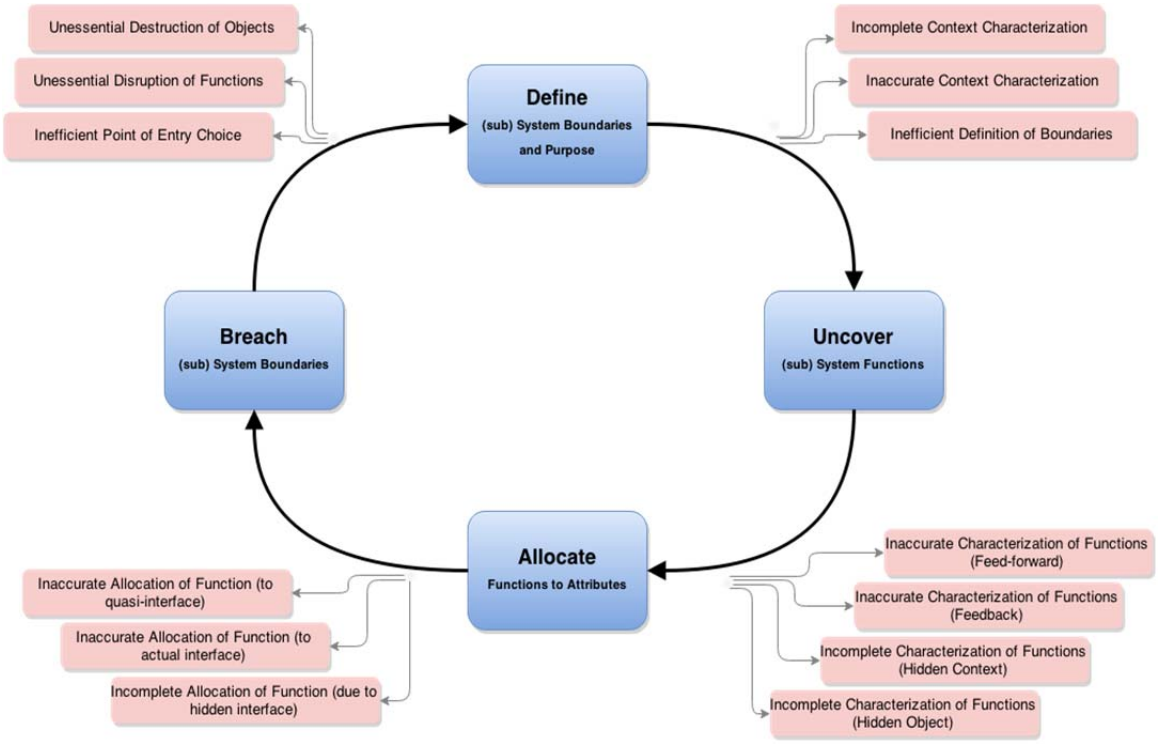

Figure 38. The reverse engineering process model showing modes of failure

The modes of failure are shown adjacent to the transition in which they may arise

VII. CASE STUDIES

> The eyes and fingers—the bare fingers—are the two principal inlets to trustworthy knowledge in all the materials and operations which the engineer has to deal with.
>
> —James Nasmyth[40]

A. INTRODUCTION

It has been argued that a model of reverse engineering may be used as a language to think and communicate about reverse engineering. It has also been suggested that a model can be a vantage point from which real world cases of reverse engineering can be evaluated. Finally, it has been claimed that a model is a map that can be used as a tool in a search for heuristics. Such a map has been developed in the preceding chapters. However, an untested map is a collection of symbols of questionable validity. It becomes valuable only after is has been validated against that portion of real world it purports to represent. The objective of this chapter is to provide some validation of (and possibly some improvement upon) the proposed model for the process of reverse engineering.

B. METHODOLOGY

As suggested above, validation is a process whereby a model is held up against reality in order to confirm its accuracy and usefulness. The validity of the proposed model of reverse engineering will be evaluated in terms of how the model does several things.

Does the model accurately describe reality? In using reverse engineering to discover the functions and operational principles of a system—does the real world process follow the modeled process? Did the stages and transitions between stages take place as predicted? Were inter-stage modes of failure encountered, and did they match the ones predicted by the model?

---

[40] Ferguson, Eugene S. 1994. *Engineering and the Mind's Eye*. The MIT Press. P. 50

Does the model work as a prescription for reality? This question arises because the model describes a human activity that can be patterned after the model whether this is a good idea or not. That is, a model of reverse engineering is like other models used to describe a learning process. Models of "how do we learn X" are de facto models for "how should we study X." In using reverse engineering to discover the functions and operational principles of a system: Does it lead to discovery? Would closer adherence to the model lead to more fruitful reverse engineering? Did the actual experience of reverse engineering suggest any changes or revisions to the model? If so, what are these changes?

Does the model scale? Can the model be used to guide and/or understand reverse engineering of wide range of systems regardless of size, complexity, or other relevant variables? This question cannot be answered in this dissertation given the small amount of case studies and their relative simplicity. However, the variation in complexity between the projects may perhaps be used as a basis for extrapolating. Did moderate variations in complexity result in practical differences in the application of the model? If so, how might these differences change when the target system changes in complexity by several orders of magnitude? And is it possible that the model will remain useful or relevant in such circumstances?

What heuristics were discovered? One of the motivations for creating a model was that it might be useful as an aid in identifying heuristics. The researcher analyzing a case study of reverse engineering in light of the model may say something like "The reverse engineer just transitioned between the function-discovery and interface-allocation stages—did he incur any of the potential modes of failure expected for this transition? If not, why? If yes, what could have helped avoid this?" The incidence or near miss of a mode of failure can be used to identify steps that might have led (or did in fact lead) to its avoidance. A second method to identify a heuristic through the case studies (though not necessarily with the aid of the model) is to search the case study for significant discoveries—moments when the reverse engineer learns something that is both new and particularly illuminating to the problem at hand. If these discoveries can be identified, then the sequence of actions that lead to them—if they can be found and expressed in terms that make them generally applicable—are heuristics. Finally, a third approach was

taken to find heuristics in the case studies. This method consists simply of asking: In this case study, what was particularly difficult? What bit of information or what physical step involved more effort than anticipated? Answering these questions may lead to a better understanding of the problem landscape: what must the reverse engineer prepare for? This preparation in itself may be considered a heuristic.

Throughout the remainder of this chapter each of the case studies will be considered, and the proposed model of reverse engineering will be subjected to these questions. The author performed the reverse engineering activities in Cases I–IV. Case V is based on the historical and ongoing effort to reverse engineer an ancient artifact that has come to be known as the Antikythera Mechanism. In Cases VI–X the author used a formal executable language (Monterey Phoenix) to create a virtual representation of his reverse engineering model and then analyzed the event traces generated by the language.

1. Case Study I (Foaming Pump)

A soap bottle with a foaming pump was selected as a low complexity reverse engineering project. The general purpose and operation of the system were well known to the reverse engineer: when hand-pumped, the bottle takes liquid soap from a reservoir and mixes it with air in order to deliver foam. However, in spite of possessing thorough acquaintance with the operation of this type of system (and occasionally even wondering about it), the reverse engineer feigned no hypothesis that may explain how the "foaming" was accomplished. The specimen situation included two bottles, giving the reverse engineer the option to keep one specimen intact while tearing down the other. The system was presumed to involve only a small number of moving parts and a simple single operational principle. In the end, this assumption held true, and yet this project proved to be the most difficult of the four hands-on case studies. The target systems are shown in Figure 39. The narrative of the reverse engineering process is found in Appendix B.I

Figure 39. Case study I target system

Foaming Pump Soap Bottle: in original packaging (1), with labels removed to show mechanism of interest (2).

a. Model vs. Reality

The context-exploration stage of the model involves finding the purpose of the target system, obtaining a full characterization of its context, and defining its boundary in order to focus the activities that will follow. Familiarity and use reduce the possibility of incurring modes of failure in the transition between the context-exploration and function discovery stages. This describes the situation encountered. The purpose and context were fully and accurately characterized from the outset (Appendix B.I.1-2). The system boundary definition was also complete, although it was subsequently revised: the bottle (the reservoir of soap) was initially considered as possibly playing a role in the system function. This was not the case (an experiment confirmed this). The bottle was therefore placed outside the target system boundary.

The function-discovery stage of the model involves the full and accurate characterization of target system functions or cross-boundary flows. This was a relatively simple task and no modes of failure were incurred. There is a single functional interaction (the operator pushes on the pump) and three fairly obvious fluid flows (soap, air, and foam). One possibility for an erroneous characterization of a function arose when the possibility was considered that the bottle might act as a pressure vessel. However, this suggested a simple experiment, and the experiment eliminated that possibility. In this case study, the transition from the function-discovery to the interface-allocation stage incurred no modes of failure.

The interface-allocation stage of the model involves the full allocation of functions from the function-discovery stage to physical attributes or interfaces. This is where this case study became more interesting. The components that support interactions with operator, soap-suction, and foam-discharge are all clearly identifiable. But while it was obvious that the system consumes air, it was not equally obvious where this air was consumed through—air could enter the system through the nozzle, through the threaded interface with the bottle, or through the gap between the moving components of the pump. The final correct determination of the atmosphere—foaming pump interface was not attained until later on, after boundary breach and tear-down permitted close inspection of the question.

The boundary-breach stage of the model involves the location and exploitation of a good way to get into the system. This called for the minor destruction of a "sleeve" component whose only function was to serve as an interface between the pump and the bottle (i.e., the location of the threads). Due to the close physical interaction between the various components, it was feared that the mode of failure "disruption of functions" would be unavoidable. However, the system could easily be reassembled every time it was necessary to test a hypothesis that the exploration of the component had suggested. Therefore, no modes of failure were incurred as the project went from the boundary-breach stage (breach and tear-down) to the second iteration of the context-exploration stage (consider the context of the internal components, infer their purpose, and define their respective boundaries). The end result of the tear-down is shown in Figure 40.

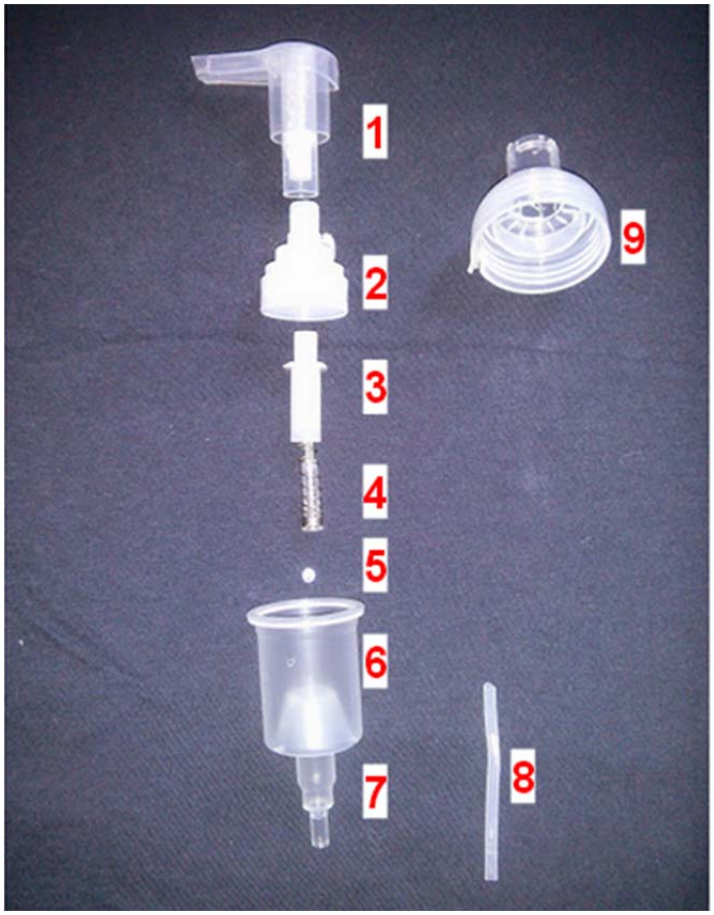

Figure 40. Foaming pump soap bottle fully disassembled

Nozzle and foaming chamber (1), ready chamber (2), soap pump piston, valve, and chamber 'b' (3), valve stem (4), small glass ball [part of one-way ball-and-spring valve] (5), air chamber 'a' (6), air chamber 'b' and soap chamber 'a'(7), and suction tube (8)

At this point the boundaries between subsequent stages became blurred. Looking at this case study for the first time after its completion, it seemed that the process described in the model completely dissolved at this point. What followed was a series of experiments during which purpose, boundary, function, and physical interfaces were all more or less simultaneously "teased out" a little at a time. The experiments involved the repeated assembly and disassembly of the components, and the replacement of different flows through different sections of the mechanism.

As the experiments drew to an end a complete characterization of the system's entire operational principle was at hand, it is interesting to note that the model re-

emerged. That is, the picture of the system became modularized (i.e., context-exploration stage). Specifically, the system was broken down into a series of chambers each with a distinct function. It became clear that the final challenge was to identify the features and interfaces (valves and orifices) that allowed these chambers to "communicate" with each other.

Some additional interesting results from this case study are:

1. Temporal modularization—In pursuit of the operational principle, the system was modularized not just physically, but also temporally. That is, it became necessary (or at least convenient) to think of the system at two distinct times: while pushing the pump handle, and while releasing it.

2. Diagramming—The presence of a diagram was used as a proxy for understanding. If the reverse engineer could not draw a diagram of the full system, then he did not yet understand the full system. Interestingly this process of drawing-making made explicit the modularization that had taken place in the reverse engineer's mind. That is, there came a point near the end of the project, where there were 4 internally consistent drawings, and the challenge became how to merge them.

3. Jumping Ahead through transparency—In reality, the second iteration of the context-exploration stage was already underway during the first steps of this project. The reason is that the system's transparency allowed the reverse engineer to visually inspect the internal context and flows before the breach allowed their physical inspection.

4. Iterations—Due to its relative simplicity, this case study did not call for multiple iterations of the four stages. There was a single component that called for second iteration of boundary-breach stage.

Following each case study, a system diagram was created in order to assist in the subsequent analysis of the process. Figure 41 shows the system diagram for the foaming pump.

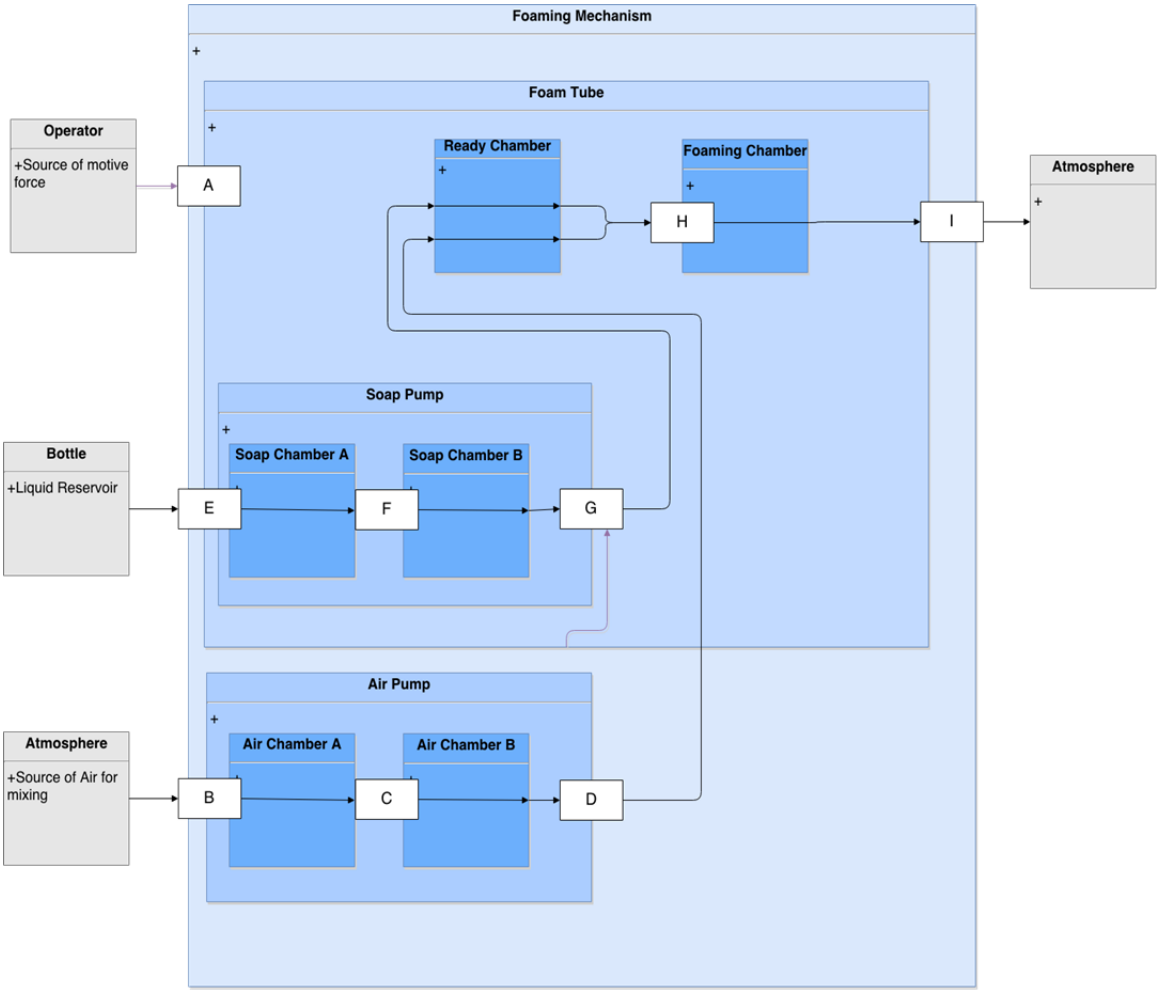

Figure 41. Foaming pump system diagram

The modest "technology" involved is not a good indicator of the difficulty of the project. This system was the hardest of all to "figure out" requiring multiple models and experiments until a satisfactory explanation was arrived at. Labeled interfaces: A—Pump Head; B—Air Gap; C—Check Valve (Flap); D—Air Gap; E—Check Valve (Ball-and-Spring); F—Free Flow; G—Stem Actuated Valve; H—Orifice; I—Nozzle

b. Major Difficulties.

In order to understand the operational principle it was helpful to consider the two flows (air and soap) separately. It was also helpful to consider each flow during two distinct stages of the system operation (push and release). This resulted in the relatively clear grasp of four distinct flows (air/push, air/release, soap/push, and soap/release). The most difficult part was the project consisted in integrating these four flows into a single coherent picture. As a corollary to this, it was particularly difficult to resist the urge to

claim complete understanding of the operational principle when in fact this understanding had not been reached.[41]

c. Heuristics[42]

The following heuristics were observed during this case study:

- H: Remove and Operate to find the function or relevance of a component [B.I.8]
- H: Break down a question into its parts [B.I.11]
- H: Experiment to see more clearly—You can do this by:
  - Using repetition [B.I.15]
  - Removing opacity [B.I.13]
  - Enhancing contrast [B.I.23]
  - Changing the speed [B.I.15]
  - Disassemble-reassemble-repeat [B.I.28 through the end]
- H: Make a hypothesis
- H: Experiment to test hypotheses
- H: Make a drawing
- H: Break the problem into distinct flows
- H: Break the problem into distinct events
- H: You do not really understand the system until you can bring it all together (this was easy to forget)
- H: Unless you can draw the whole thing, you do not understand the whole thing
- H: When you think you are done you usually are not

[41] A noted shortcoming of the case study analysis is the lack of independent validation of the system model which, in the first 4 case studies is nothing more than the final version of the author's working model. For this reason, 5 additional case studies were incorporated using process simulation in an executable formal language known as Monterey Phoenix.

[42] The alphanumeric information at the end of each heuristic is a reference to the its source in the case studies as documented in the appendices. For example **[B.I.8]** means that the source of this heuristic is found in Appendix B, Case study I, Step 8.

2. Case Study II (Medium-size Robot with Sensors)

The Hexbug Original was selected as an example of a medium complexity reverse engineering project. Figure 42 shows the system in its original packaging and during the static inspection. The problem solver was completely unfamiliar with the design features or operational principles incorporated into the system, with the exception of those features that were explicitly stated as part of the system's package name: *Hexbug Original: The Robotic Creature That Reacts to Touch and Sound*. The design was presumed to incorporate a moderate number of moving parts and two or more operational principles. In the end, this project was the simplest of the four hands-on case studies. The narrative of the reverse engineering process is found in Appendix B.II

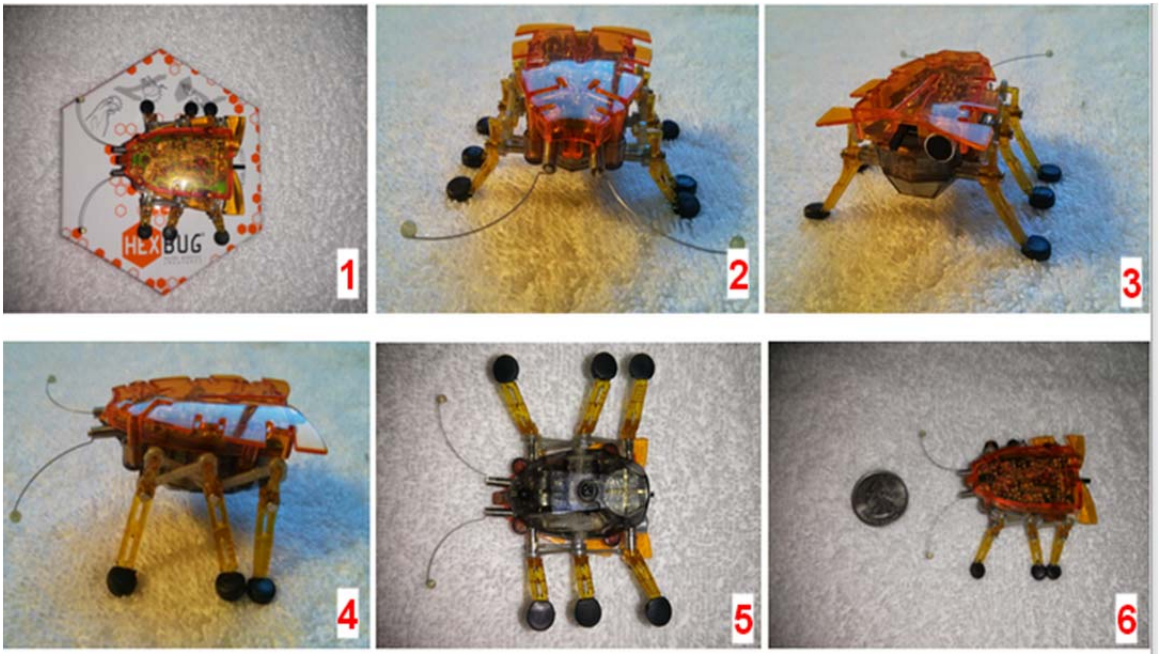

Figure 42. Case study II target system

Hexbug Original (with Obstacle and Sound Sensors) Prior to Project: in original packaging (1) front view showing antenna [switches] (2), rear view showing sound activated sensor [switch] (2), side view showing inter-leg linkages (4), bottom view [some interesting features are visible through semi-transparent shell, including asymmetric arrangement of gears], size reference [a quarter] (6)

a. Model vs. Reality

Per the context-exploration stage of the model, system purpose, context, and boundary were completely and accurately characterized. This characterization was based on the system's appearance, the functional description included in the system's name, and the pre-existing knowledge of the fact that the system is a toy. No modes of failure were incurred as the project transitioned to the function-discovery stage.

The function-discovery stage involved a series of inspections both powered and off. The target system functions were all discovered, however there was an error. The system responds to physical obstacles (detected by the antenna) by backing up, turning, and heading in a different direction. It also responds to loud noises (detected by a small sensor) with a similar maneuver. The error lied in ignoring the fact that there was a small difference between the maneuvers triggered by the two different stimuli. It is important to note that the difference was observed, but no further attention was given to it because it was small, and probably because it did not fit with the mental image the reverse engineer had at the time of the how the system operated. As the project transitioned to the interface-allocation stage, the mode of failure "inaccurate characterization of function" had been incurred.

The interface-allocation stage was fairly straight forward and did not occupy a distinct period of time from the function-discovery stage. As the project transitioned from interface allocation to boundary breach, the earlier mode of failure was carried forward in the form of a new mode of failure "inaccurate allocation of function."

The boundary-breach stage began at the most obvious location: the battery compartment. As this POE was a dead end, it could be questioned whether pursuing it was a mode of failure "inefficient point of entry choice." However, as the only alternative was to go directly into destructive disassembly, trying the battery compartment first was probably the best course of action, even when ultimately unproductive. After this, the requirement of the boundary-breach stage drove a destructive disassembly. This could have led to the mode of failure "unessential destruction of objects." However, close inspection coupled with the application of controlled force (short of destructive), helped

identify the minimal amount of destruction necessary for breaching. As soon as the system was opened, a small gear fell out of it—incurring the mode of failure "unessential disruption of function" (temporarily, while the correct placement of the small part was determined and restored).

Following the first system breach there remained two components within the technical capability of the reverse engineer for further analysis: a small cylinder in the middle of the axle connecting left and right legs, the circuit board.

A second iteration of the model could be applied to the small cylinder. The cylinder's context consisted of the shafts that connected to it on both sides. The cylinder's function was determined through testing to be the transmission of torque in one direction of rotation but not in the other (i.e., a clutch). The functional allocation was trivial: the points where the shaft coupled with the cylinder. A second iteration breach was undertaken which revealed the operational principle behind the clutch. The mode of failure "unessential disruption of function" lurked during the breach due to the tiny size of the components involved and the presence of a spring. However, the earlier incident with the small gear falling out primed the reverse engineer to be extra cautious, and no parts were lost. Figure 43 shows the system in various stages of the breach and tear-down.

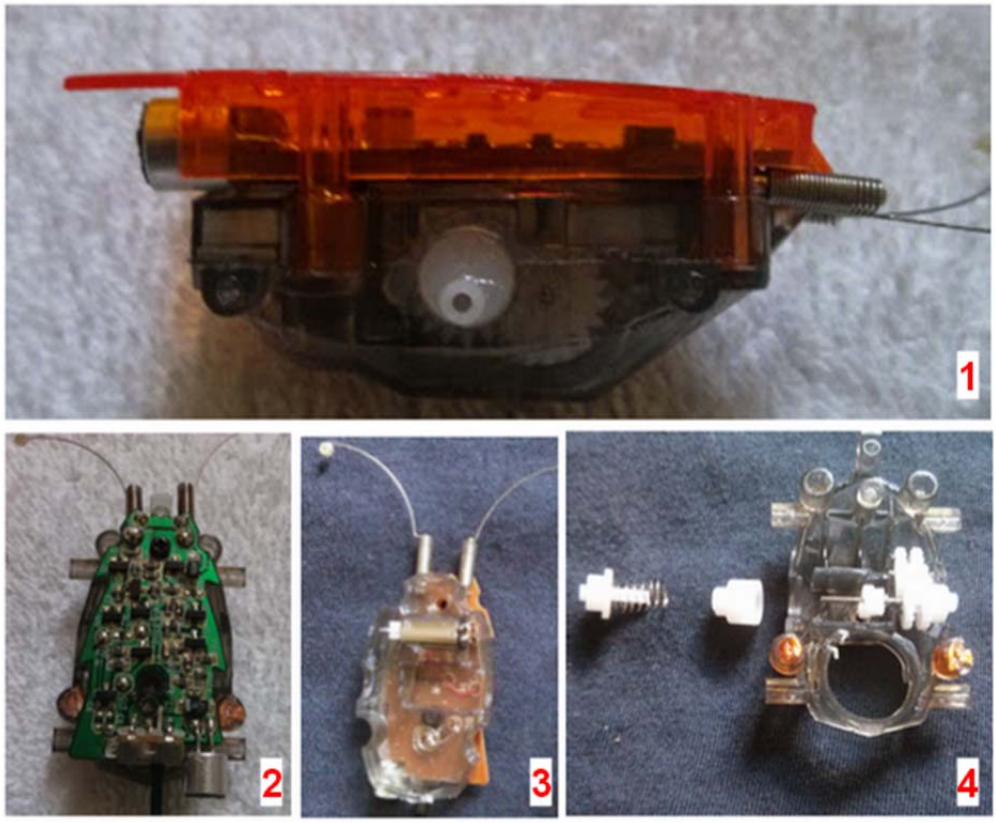

Figure 43. Hexbug Original (with obstacle and sound sensors) in different stages of disassembly

With only legs removed, the cam shaft powering the center leg [only] is seen clearly (1), top view minus the carapace shows the circuit board, the shape is clearly intended to achieve a distinct aesthetic effect (2), under the carapace, showing the motor (3), bottom part of the body including the gears and the clutch [to the left side] responsible for uncoupling the left-side legs when executing the "obstacle avoidance" maneuver

A second iteration of the model could also be applied to the circuit board, but only in a limited sense (It was within the technical capability of the reverse engineer—who is not an electronic engineer—only to trace the inputs and outputs to different locations on the circuit board). It was at this point that the mode of failure incurred at the start of the project was finally noticed: the antenna and the sound-sensor provide inputs to similar but distinct IC components. This fact once discovered made sense of the previously ignored fact, and resulted in feedback up the stages to update the characterization of system functions. Figure 44 shows the system diagram for the target system.

Some additional interesting results from this case study are:

1. Since the system is semi-transparent, the initial inspection led to the discovery of some interesting internal features: the asymmetric arrangement of propulsion-related components, and the unusual shape of the circuit board.

2. The inspection (function discovery and interface allocation) went beyond mere observation to include prodding and pulling of components (legs) where these actions might yield information. The operational principle controlling the gait of the robot was completely characterized before the system was turned on. The model does not address the possibility that some operational principles can be fully understood without the need for a breach of the system

3. The primary question in the reverse engineer's mind going into the tear-down was informed by an incorrect belief of the operational principle this mistaken model was amended as soon as the system was breached (a case of feedback). The actual operational principle was much simpler than the one originally suspected.

4. Once the correct operational principle had been deduced by inspection, some challenges were encountered in attempting to confirm the deduction by observing the system in operation with the operational principle exposed.

5. A number of characteristics of the system's behavior (a clicking sound, and the small shuffling movement of the left legs during the "evasive maneuver") were present from the outset, and yet were not actually observed until after a mechanism was discovered that accounted for their presence (the clutch).

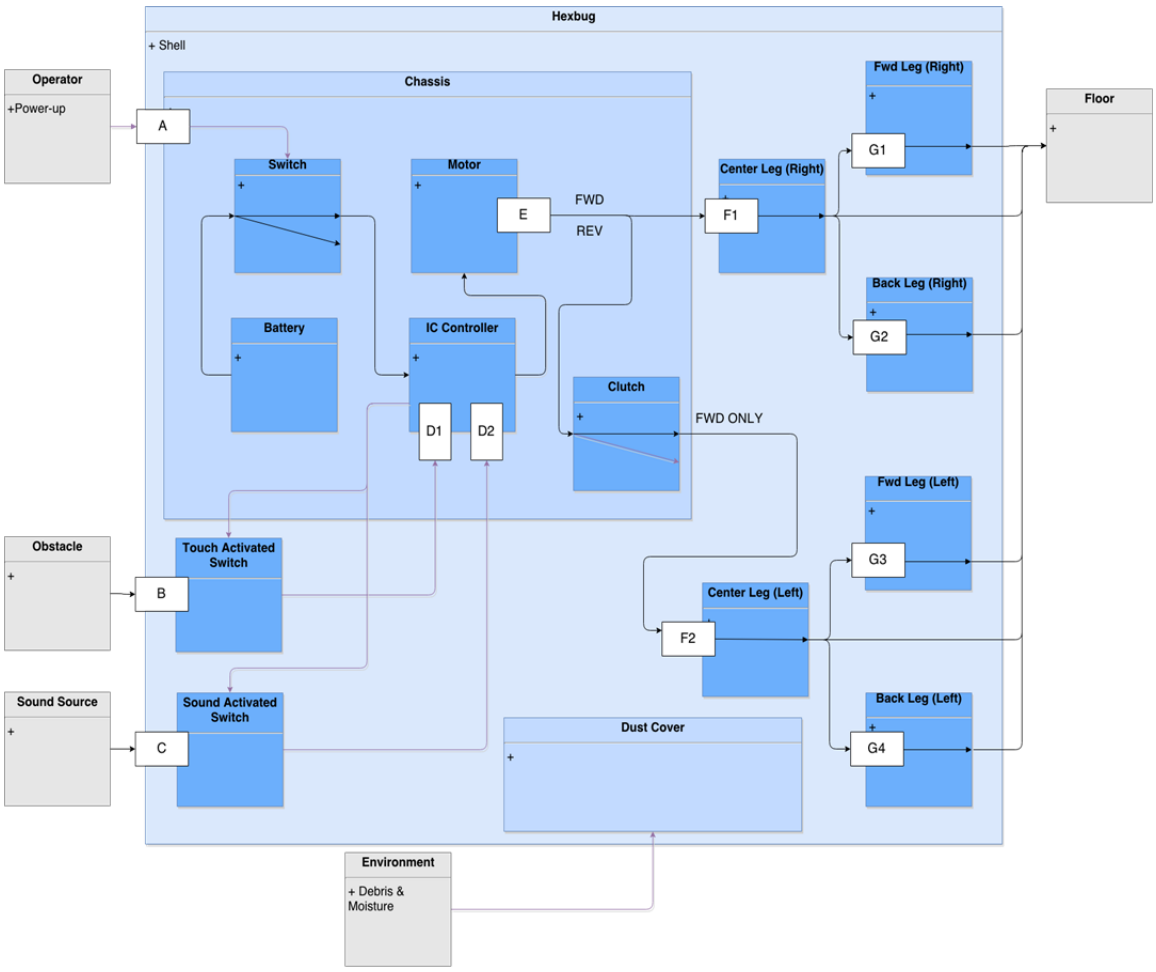

Figure 44. System diagram for Hexbug Original (with obstacle and sound sensors).

> While there are many parts, the repetition of functions and largely symmetric arrangement made the discovery of all operational principles fairly simple. The largest mystery was "What is the mechanism responsible for the obstacle evasion maneuver?" The answer [a reversal of the motor acting in conjunction with a clutch] was obvious once the system was disassembled. Labeled interfaces: A—Lever; B—Antenna (switch); C—Sound Detector (switch); D—J476 (Analog->Digital); E—Output Shaft; F—Camshaft; G—Linkages

b. Major Difficulties

This was a relatively easy project. The only aspect of it that presented a challenge was the effort to effect the breach and tear-down while preserving the ability to rebuild the system later. The reason for the difficulty was the use of non-reversible assembly methods (i.e., epoxy). A second and minor difficulty was encountered in the form of very small parts that fell out of place during the disassembly. With the parts out of place,

operational testing would be inconclusive and restoration to operation would be impossible. Therefore, a relatively large amount of the effort was dedicated to ensuring the small parts were all in place before going any further. The challenge was presented by the twofold need to manipulate a tiny part with precision, and the need to ascertain its proper placement based on cues and testing.

c. Heuristics

The following heuristics were observed during this case study:

- H: Certain things may be harder to notice once the system is running: see what you can figure out before you operate the system (Appendix B.II.11)
- H: Pull, probe and poke to elicit information (Appendix B.II.8 and others)
- H: Small details that do not make sense probably reflect a disproportionately large error in your working model (Appendix B.II.44)
- H: Look for Seams. If you do not see any, remove something. Look again. (Appendix B.II.23)
- H: When attempting to breach a system it is often useful to apply force up to but just below the point of breaking, this helps expose seams. (Appendix B.II.26)
- H: Just because some destruction is inevitable it does not mean that the system cannot be reassembled and returned to full operation (Appendix B.II.29 and 45)
- H: Breaching the system can (often does) in the loss of information (i.e., tiny or spring-loaded components)
- H: Use all your senses to see (Appendix B.II.43)
- H: Take photographs during disassembly—they may prove useful during assembly (Appendix B.II.45)
- H: After you are done, reassemble the system and observe it again—you will see new things (Appendix B.II.46)

3. Case Study III (Toy Gun)

The Nerf Firestrike was selected as an example of a low complexity reverse engineering project. Figure 45 shows the gun prior to the start of the project. The author had no direct knowledge of the operational principles incorporated into this system, but

had a good idea of what these may be, from playing reverse engineer as a child. However, the author was intrigued by the claim in the packaging label that the system was capable of ranges in excess of 75 feet. This performance suggested that the operational principle might incorporate more advanced technology than previously encountered. Also, the system is advertised as having a "precision light beam" sight. This secondary system would provide an opportunity for analyzing interfaces and characteristics other than purely mechanical ones. In the end the operational principles involved were not much more complex than originally envisioned, and yet the project did include a number of interesting puzzles the answers to which were not immediately apparent. The narrative of the reverse engineering process is found in Appendix B.III.

Figure 45. Case study III target system

Nerf "Firestrike" Prior to Project. Shown in original packaging front (1) and back (2). Outside the packaging with three included "bullets" (3) and left side showing point of entry (4).

a. Model vs. Reality

As in previous case studies, the context-exploration stage was almost perfunctory. Familiarity and use obviate the need to spend any time discovering the purpose, context, or boundary of a system. This reinforces the value (previously discussed in Chapter III) of approaching a system as a user prior attempting to reverse engineer it. The transition from context-exploration stage to function-discovery stage incurred no modes of failure.

Similarly, the function-discovery stage requirement to characterize of the system functions is apparently simplified by familiarity with the system. However, this same familiarity can have a negative effect. There were three obvious functional interactions between the user and the gun: cocking (loading the system with potential energy the discharge of which results in the firing), aiming (operating the sight), and shooting (operating the trigger). There was also a functional flow between the gun and the environment (ideally, the target): the flow of ammunition. However, in this case study, the target system also had an unexpected function. As the project transitioned from function-discovery to interface-allocation, the mode of failure "incomplete characterization of system functions" was incurred. The function in question is a safety feature that prevents the gun from discharging a high-speed burst of air (potentially into someone's eyes) when dry fired.[43] Figure 46 shows the interfaces for the various system functions.

[43] While the operational principles behind this function were fully exposed by the reverse engineering process, the purpose of the function as a "safety feature" is only a guess. This points to a limitation of reverse engineering as a means to discern purpose.

Figure 46. Nerf "Firestrike" front and back views

Front view shows barrel, sight system lens, and read ammo storage for two bullets (1). Rear view shows charging handle (2) and "iron" sights/accessory rail.

The importance of determining the use cases that went into the design of a target system was introduced in Chapter III. The gun offers three use cases: carry, dry-fire, and fire. The interface-allocation stage calls for the inspection of each use case to arrive at a full characterization of the interfaces. While exploring the muzzle during dry fire the mode of failure "incomplete characterization of system function" was corrected through feedback. The transition from interface allocation to boundary breach did not incur any additional modes of failure. It is important to note that at the end of the interface-allocation stage, the inspection of the system had raised more questions than it had answered. In particular, the muzzle was unexpectedly "closed" and it gave no clues as to the mechanism that impels the ammunition.

The boundary-breach stage appeared simple with an obvious and large point of entry. The receiver (i.e., the body of the gun) consists of two shells held together with

screws. This type of construction seemed ideal, as it would give access to the entire internal system, without requiring any damage to the system boundary.

It is noted that there was a near-miss modes of failure: "unessential destruction of objects" almost occurred when the reverse engineer attempted to breach the system but lacked the correct tool. Several screw heads were almost stripped. The damaged screws could turn the simple disassembly into an irreversible destructive one. At the very least, it could have resulted in a delay and expenditure of unnecessary effort. Instead, the project was paused until the correct tools were on hand. The second forestalled mode of failure was "unessential disruption of functions" which could have resulted had any springs fallen out during the system breach. Springs present a particular challenge for a number of reasons. For example, springs can sometimes be "loaded." This sometimes results in their flying out a considerable distance, possibly getting lost, upon careless breaching. Springs also tend to be small and easy to lose. Finally, the place of a spring within a mechanism is not always evident once the spring has flown out.

After the boundary-breach stage the effort was focused on answering the questions raised during the interface-allocation stage: How is the ammunition impelled given that there is no discernible movement of air or parts during a dry fire? The answer was easy enough to find following the full system disassembly: the rear end of the barrel was a valve that was opened when the ammunition was loaded. Following the full disassembly there were no remaining components with any internal complexity that might hide a function. There was no need for a second iteration of the four stage process.

Some additional interesting results from this case study are:

1. After the project started it became apparent that the system required batteries and none were included in the packaging. The project was put on pause until batteries were procured and installed. (Appendix B.III.2)

2. The performance of the light beam sight was interesting. Taken at first to be little more than a red flashlight, the "sight picture" was unexpected. For this reason some parameters of the sight's performance were recorded for later analysis of the subsystem (see Figure 47).

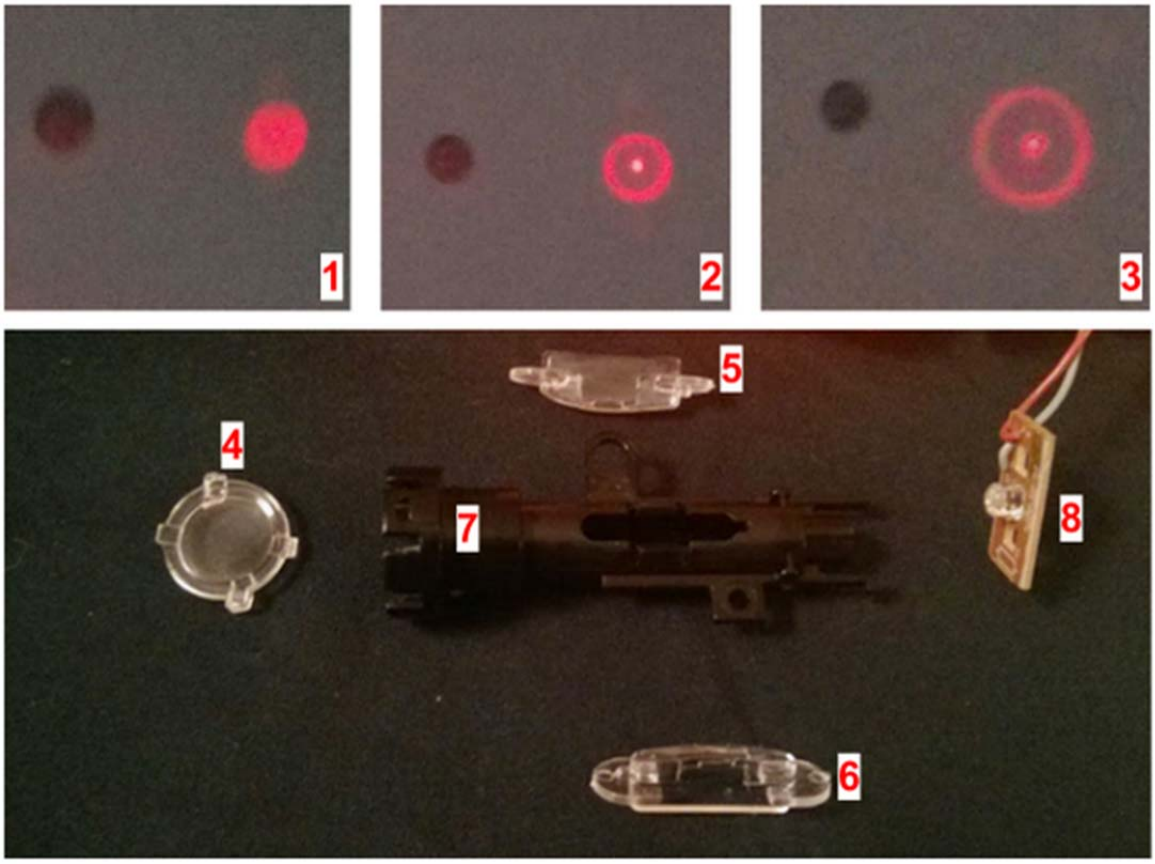

Figure 47. Sight system parts and operational test

The sight was tested at various distances. 5-feet (1), 10-feet (2), 15-feet (3). Sight system shown fully disassembled: beam-shaping lens (4), side "lenses" [apparently decorative] (5,6), optical tube (7), and LED (8)

Here again drawings were used several times. The making of these diagrams is not described or prescribed by the model, but by now it was routinely incorporated into the projects as a way to document what the reverse engineer knew at a given time.[44]The fully disassembled system is shown in Figure 48.

[44] The importance of "drawing a picture" is also highlighted as an important aspect of problem solving or engineering in a number of the works reviewed for this dissertation (Polya 1973), (Ferguson 1974), and others.

Figure 48. Nerf "Firestrike" fully disassembled

Parts as follows: barrel (1), air valve (2), cylinder (3), piston and piston head (4), sear (5), charging handle (6), barrel support [apparently decorative] (7), light sight system (8), trigger (9), sight light actuator (10), right side of receiver (11), battery compartment cover (12), barrel support rails [apparently decorative] (13), bullet (14)

It is interesting that a complete and accurate understanding of the system functions and operational principle did not result in a conclusive answer to a question that emerged at the end of the process: what is the purpose of the air valve? Another way to say this: a function can definitively be explained "down" to its operational principle, but it does not follow that it can be explained "up" to its purpose. The system diagram for the Firestrike is shown in Figure 49.

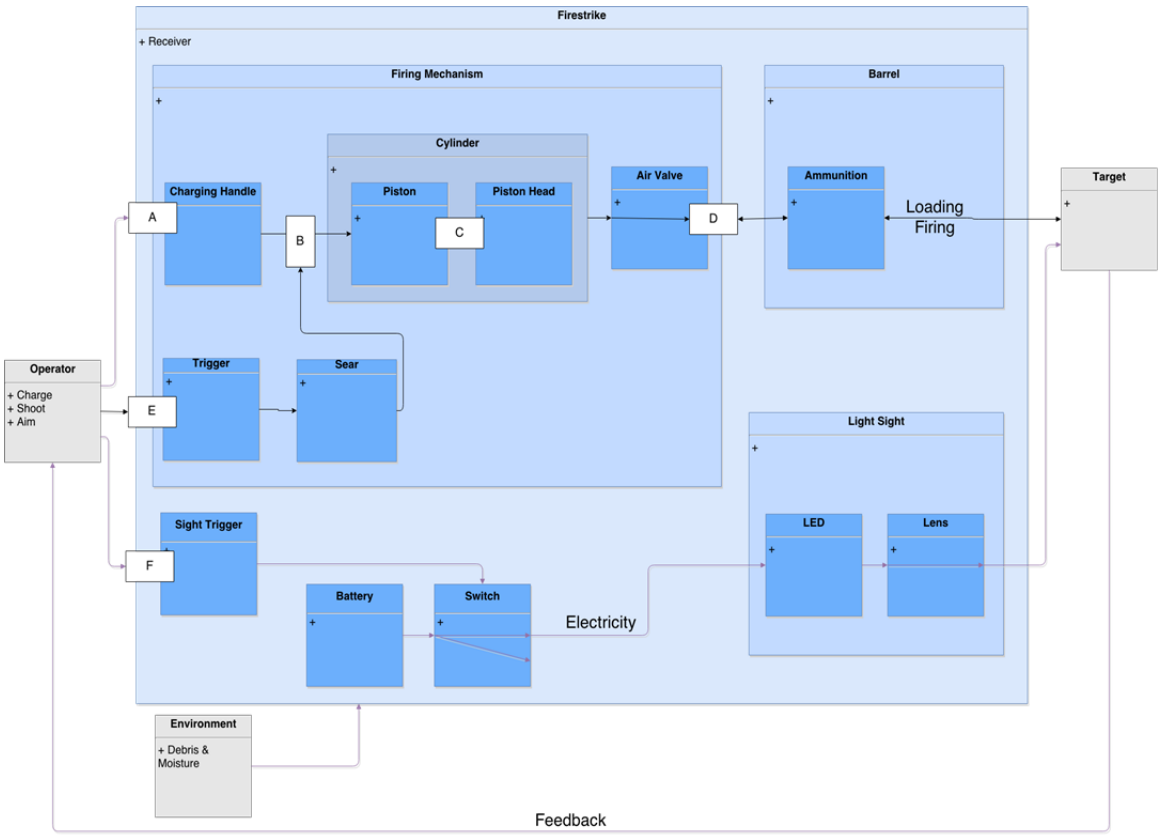

Figure 49. System diagram for the Nerf "Firestrike"

Given the low price and small size, the performance of this system was impressive. For example, the sight system although using normal red light, produces a sight picture (on the target) that is small and intense even at distances over 20 feet. Additionally, the muzzle speed feels high, and this is supported by a very straight flight path [Given indoor conditions, the bullet drops approximately 6" in 25' of horizontal travel]. At distances of 10' +/- 5 the bullet hits within one inch of the sight dot. Labeled Interfaces: A—Handle; B—Sear Notch; C—Direct coupled; D—Valve; E—Trigger; F—Sight Actuator

b. Major Difficulties

Based on the overall time spent on the different aspects of this project, the "major difficulty" encountered is easy to determine. A disproportionate amount of time was spent in procuring the right tools (i.e., correct size screwdriver) and supporting necessary material that is part of the fully operational system context (i.e., batteries). This difficulty seems very prosaic, but perhaps it is worth taking note of. The lack of foresight incurred two unplanned trips and probably doubled the amount of time spent on the project. The presence of small springs could have led to a second challenge like the one encountered

for the previous project (small parts flying out of place), however this was anticipated and forestalled.

c. Heuristics

The following heuristics were observed during this case study:

- H: Have the necessary supplementary material—get it (Appendix B.III.2)
- H: Identify each state or mode of operation for the system (Appendix B.III.1)
- H: The, deliberately observe the system during each state or mode of operation (Appendix B.III.6-15)
- H: Before opening the system, make a diagram of what you expect to find (Appendix B.III.13)
- H: If you find something unexpected, revise your drawing (Appendix B.III.13)
- H: Sometimes you have to gather new questions before you can get any good answers (Appendix B.III.6-15)
- H: Have the right tools (Appendix B.III.18)

4. Case Study IV (Small-size Toy Robot)

The Hexbug Nano was selected as an example of a low to medium complexity reverse engineering project. The author had no knowledge of the design features, performance characteristics, or operational principles incorporated into the system. The author was intrigued by the fact that the system appears to have a relatively high volume of sales (based on significant shelf space occupied at the retailer by this system and a number of available versions and accessories). The system is shown in its original state in Figure 50. In the end the system was mechanically simple (corresponding with the low price), yet its behavior was surprisingly complex, robust, and entertaining (corresponding with its popularity). In spite of the mechanical simplicity, the behavioral complexity made for a very interesting project. The narrative of the reverse engineering process is found in Appendix B.IV.

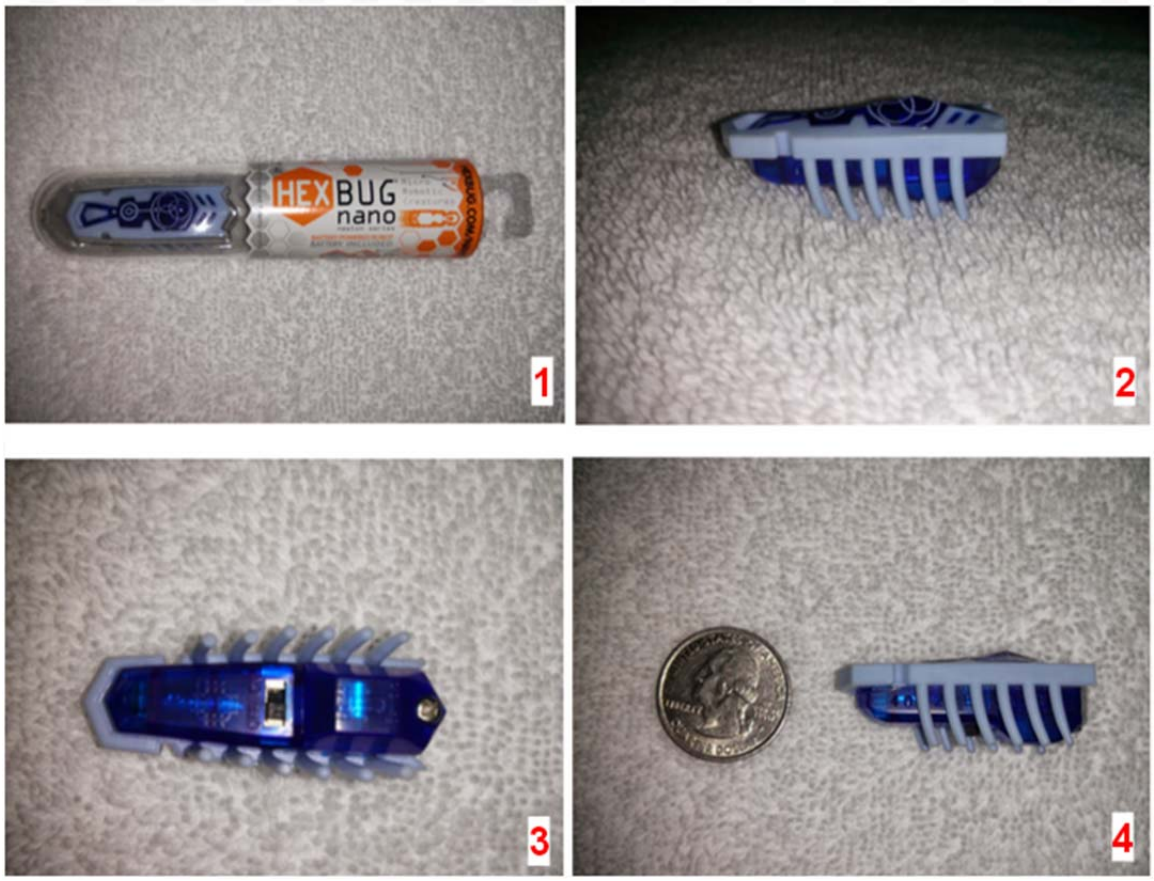

Figure 50. Case study IV target system

Hexbug Nano Prior to Project. in original packaging [the approximate size and shape of a test-tube] (1), side view (2), bottom view (3), size reference [a quarter] (4)

a. Model vs. Reality

The context-exploration stage of reverse engineering consists of identifying the systems with which the target interacts. The first and third case studies (foam pump and toy gun) involved systems that achieve their full functionality through their interaction with a user. The second case study (the robot with sensors) was an autonomous system. By definition, the only user interaction that the robot required was to be turned on or off. Once powered it engaged in moderate interaction with other systems in its environment through its sensors and traction.

For this last case study, the context-exploration stage of the process appeared to be almost the simplest possible. The system context could be defined by a user (to turn it

on/off), and a surface (presumably smooth and level) on which to travel until it bumped into something. In other words, the system did not appear to be complex enough to interact with its environment in any meaningful way. This was an error. The transition from context exploration to function discovery incurred the mode of failure "incomplete context characterization."

During the function-discovery, a dynamic inspection revealed the system behavior to be more complex than expected. For example, in spite of the absence of sensors or in fact of any apparent control mechanism, the system interacts with its surroundings in such a way that it can navigate an obstacle-filled environment. The principal design goal appears to be: to create a system whose size and behavior resemble a small fast beetle or a cockroach.[45] An ability to navigate an obstacle filled environment resulted in a reevaluation of the context to include obstacles, and perhaps other things.[46]

The interface-allocation stage was at once simple and challenging. The system has a single simple interface other than the on/off switch: 12 flexible but inert legs. But in a case like this, the interface-allocation stage does need not (and should not) conclude with mere identification of the interface. What is it about the legs that is responsible for the system's unexpected behavior? Is there something besides the legs?

As the reverse engineer prepared to transition to boundary-breach stage the modes of failure "incomplete characterization of system functions" and "incomplete allocation of function" were almost incurred. The system includes a self-righting function. If it ends up upside down, the geometry and material of the system's carapace interact with the surface in such a way that it is quickly returned to its upright position. This function was not initially discovered, and would have remained so if the boundary breach had not been postponed to conduct more testing.

---

[45] The system is advertised as "the robotic creature that behaves like a real bug" however, this characterization was not available to the reverse engineer who discovered and was surprised by this behavior during the course of this project.

[46] Navigate is used in the sense of to traverse an environment without being effectively stopped by the obstacles encountered. An online search for accessories for the Hexbug Nano yields a considerable amount of options. This suggests the other interactions by design exist which did not become manifest during this case study. The OEM has this to say about the mechanism. "Thirty some iterations, a few years, and dozens of designs later the Nano was born. To give just a taste of how much development went into this product, we engineered and rapid prototyped over 150 variations of legs alone."

An inspection suggested the system would not yield to a breach without undergoing significant destruction. With the prospect of irreversibly losing the system's function, the reverse engineer decided to carry out a series of impromptu experiments. These were aimed at obtaining a fuller characterization of the locomotion mechanism before the breach of system resulted in the permanent loss of its operation. The tests attempted to answer questions like: will the system right itself? how does this locomotion mechanism respond to damage? and could it be steered? These are not questions that fit within the definition or model offered in this dissertation. Figure 51 shows the Hexbug Nano undergoing some testing.

Figure 51. Hexbug Nano operational test

Test included disabling anywhere between one leg (1) and 10 legs (2). The system's performance was only moderately impaired even when only the rear two legs were left operational. On the other hand the system went into a circular path if the front-most two operational legs were not the same number [for example, if the front left leg is disabled, then the front-most operational left leg is #2 and the front-most operational right leg is #1]

After the experiments were complete, the reverse engineer was able to provide a full characterization of the operational principles. A close inspection through the semi-transparent system boundary was used to confirm the accuracy of the system diagram/model drawn. At this point there was no need for a system breach. All of the functions and operational principles were discoverable (and discovered) from the outside. This is a slight departure from the model. In spite of its simplicity, the operational principle at work in this system was the most unexpected and the most enjoyable to uncover. The process of arriving at an adequate characterization of this operational principle involved a great deal of thinking and testing.

A boundary breach and tear-down was ultimately undertaken as part of this project in order to discover any heuristics or other lessons that might be learned from the effort. Figure 52 shows the system in its full tear-down state. Very little damage resulted.

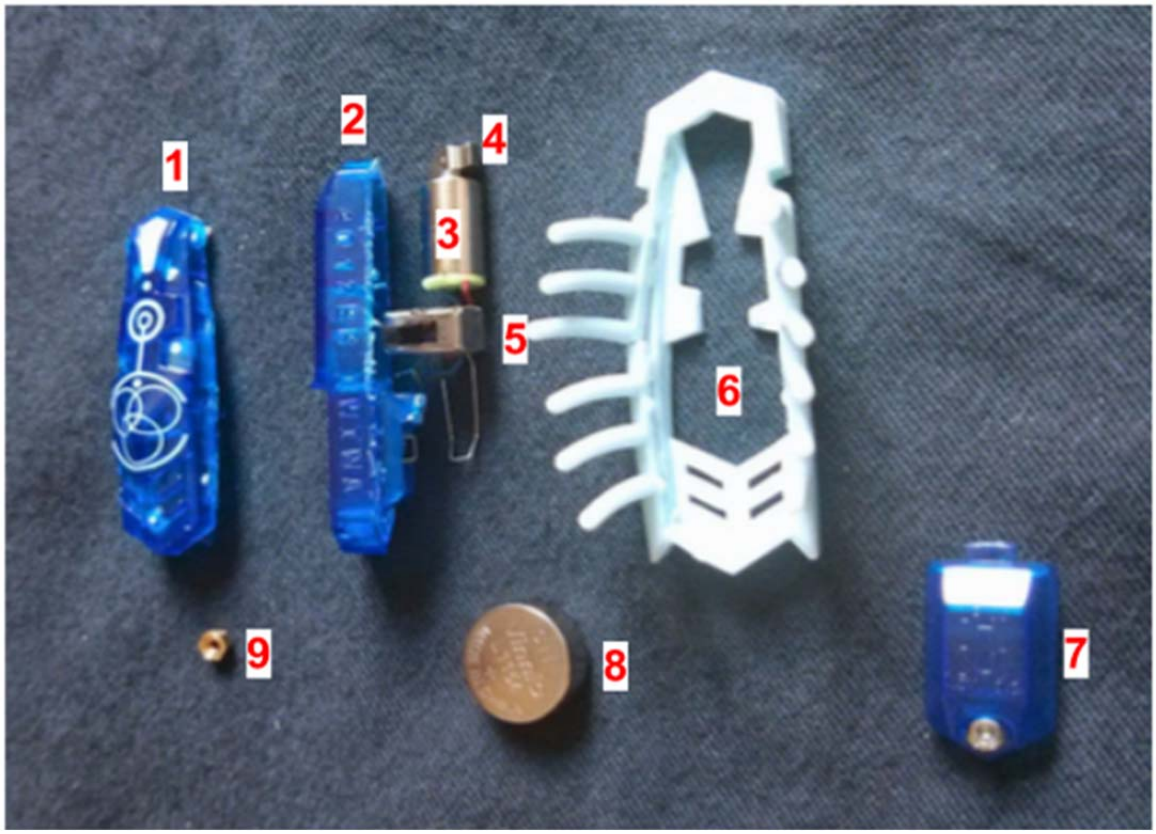

Figure 52. Hexbug Nano fully disassembled

Top section of plastic body (1), bottom section of plastic body/chassis (2), electric motor (3), asymmetric rotating weight (4), switch (5), soft part including bumper function and leg function (6), battery compartment cover (7), battery (8), and nut (9).

An additional interesting observation from this case study is that discoveries about the system continued to be made after the project was officially completed. For example, the author found that the overhanging head of the system has a geometry similar to that of diving board. This observation, which came over a week after the project, was officially over, resulted in an updated working model describing the operation of the system. Figure 53 shows the system diagram that captures all the critical system functions and components as they were understood at the conclusion of the project.

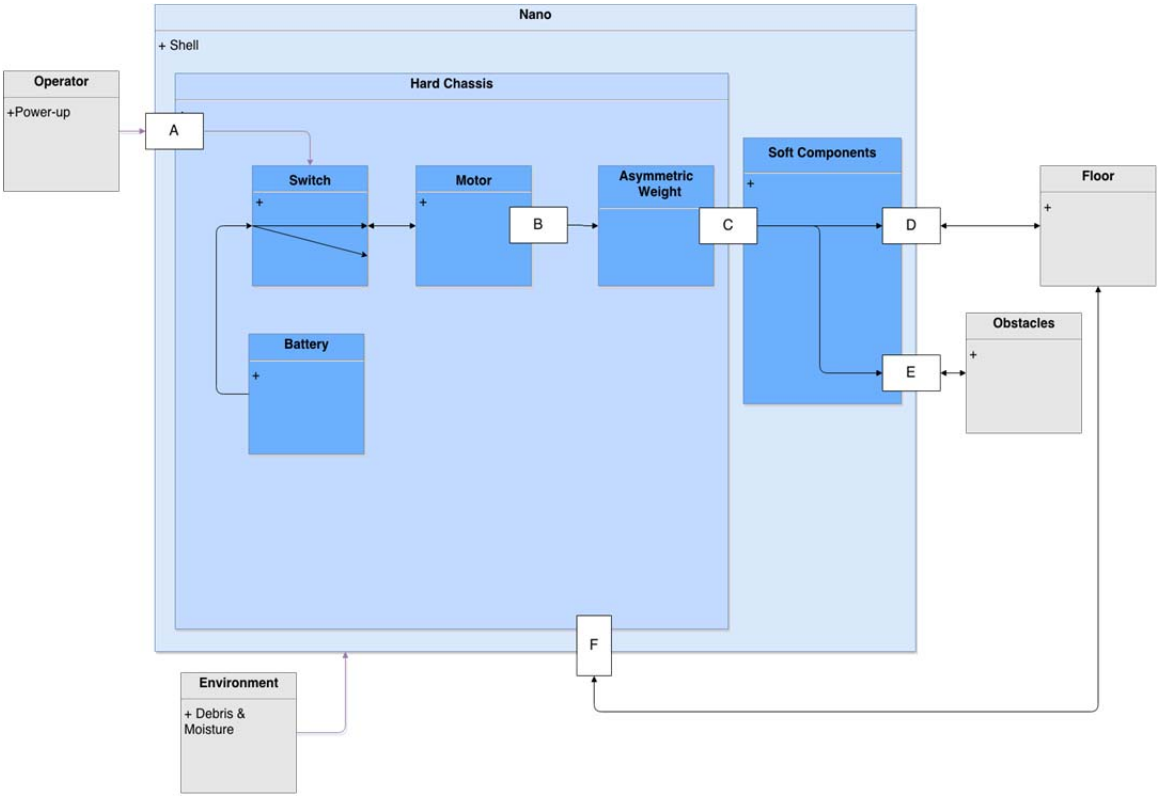

Figure 53. System diagram for the Hexbug Nano

The simplicity of the design [only one moving part] is not a good basis for predicting the behavior of the system. The behavior was remarkably life-like. In suitable conditions (smooth floor) the Nano "aggressively explored" its environment in a way that very much suggested the behavior of a cockroach or some other high-energy small creature. The design is also surprisingly robust: under many circumstances it can lose up to 10 of the 12 legs and suffer only moderate loss of performance. Labeled components: A—Lever; B—Output Shaft; C—Direct coupled vibration; D—Legs; E—Bumper; F—Hump

b. Major Difficulties

The major difficulty in this project consisted in obtaining a clear understanding of the operational principle that in this case was completely unknown. All three of the other case studies involved an operational principle that was unknown because the particular configuration of components was unknown. However, they employed combinations of well-understood components—pistons, gears, linkages, cams, switches, valves, and so forth—to achieve the mystery operational principle. In contrast, this case study employed two components that the problem solver had never encountered (flexible but inert legs, and an off-center rotating weight intended to induce a high frequency vibration). The operational principle of locomotion was also completely new. It was relatively easy to describe at a superficial level (vibration results in forward motion), but more difficult to describe at a deeper level (but how does vibration turn into forward motion?). Finally, the novelty of the system also prompted a third level of exploration (how good is this vibration-turns-into-forward-motion as a form locomotion?)

c. Heuristics

The following heuristics were observed during this case study:

- H: Do not be too eager to jump into the tear-down (Appendix B.IV.21)
- H: Think of and do experiments to see more clearly (Appendix B.IV.14-22)
- H: Think of and do experiments to verify a hypothesis (Appendix B.IV.14-22)
- H: A third kind: Think of and do experiments to see what if? (Appendix B.IV.14-22)
- H: Even the most challenging system breach problem has (so far had) a solution (Appendix B.IV.23-26)
- H: During a tear-down when no productive next step is obvious, an unproductive next step will usually do the job (and expose an productive next step) (Appendix B.IV.23)
- H: Continue to think about the problem, even after the project is over

5. Case Study V (Antikythera Mechanism)

This case study is based on the historical and ongoing effort to discover the purpose, functions, and origin of an ancient artifact that has come to be known as the Antikythera Mechanism. The mechanism was discovered in 1900, and from then until the present it has been the object of intense (though intermittent) study. Given the definition provided in this work, it may be said that the Antikythera mechanism has been the target system in a century-long project of reverse engineering. Appendix B.V provides a synthesis of the accounts presented in four separate sources. (Price 1959; Freeth 2008; Marchant 2010; Jones 2012). As with the other case studies in this work, the purpose of the appendix is to present the events in the sequence in which they occurred in order to make them accessible to examination in light of the model of reverse engineering. Consistent with the other case studies, an effort has been made to identify the various events as discoveries, thoughts, or actions. However, it is explicitly acknowledged that this time-line and classification represent information that is qualitatively different from the similarly presented data in the preceding case studies.

Firstly, the timeline here covers a scale of years and even decades, as compared to minutes in the other case studies. Secondly, based on several different sources which are themselves syntheses of years of research, the classification of the events into a few dozen discoveries, thoughts and actions is necessarily a great oversimplification of a process that has unfolded over 100 years and involved over 50 researchers (some of them consumed by life-long obsession with the puzzles presented by the project). Acknowledging that it misses the majority of the details of personal experiences in reverse engineering, it is nevertheless believed that the overall *shape* of the process can be detected and furthermore considered in light of the same model of reverse engineering as the other case studies. A part of the target system for this case study is shown in Figure 54. The narrative of the reverse engineering process is found in Appendix B.V.

Figure 54. Case study V target system (partial)

Antikythera Mechanism—The largest fragment (known as Fragment A). Source: Antikythera Mechanism Research Project, "Fragment A" 2015. Antikythera-Mechanism.gr, Retrieved from http://www.antikythera-mechanism.gr/data/fragments

a. Model vs. Reality

The context-exploration stage of the reverse engineering process calls for identifying the target system's purpose and the definition of its boundary. It has been argued that the discovery of context information generally precedes the other types of information, except where feedback takes place. In the other case studies, it was difficult to validate this sequence because context information was always known in advance. The reverse engineer of everyday things, one does not pause to ask "Who or what is this for?" One of the factors that make the Antikythera Mechanism so interesting as a case study in reverse engineering is that the target system's context is separated from the reverse engineer's context by two millennia. Accordingly the target system's purpose and boundary were completely unknown.

The model's prediction of the precedence of context information appears to be validated in the case of the Antikythera Mechanism. The first decade of research is largely concerned with the question of "Who used this and what did they use it for?" (Appendix B.V.1-12).

According to the model, function discovery seeks to find out "what does this thing do?" The first systematic attempt to answer that question is the work of Derek De Solla Price. His first answers are published in 1959. There are sufficient clues—Price explains—to understand the overriding operational principle.[47] Fifty years separate the focus on purpose and context from the focus on function and operational principle.

According to the model, the search for functions is supposed to be guided mainly by the characterization of the context that occurs in the context-exploration stage. Feedback from the interface-allocation stage sometimes provides additional information or corrective guidance. In the case of the Antikythera the context-exploration stage resulted in something like this: 1st Century BC, used for predicting the positions of various celestial bodies and events (eclipses), perhaps as a demonstration of contemporary astronomical knowledge or theory, intended user is still unknown, probably not a mariner.

The most complete functional characterization (function-discovery stage) to date goes as follows: Indicates the date; shows the position of the sun for the given date (relative to the "unmoving" zodiac); shows the position of the Moon, taking into account the first anomaly[48]; displays the Moon's phase; indicates the rising and setting of several important stars; shows the Metonic and Callippic calendars[49]; shows the Saros and

---

[47] The mechanism can be described as employing a single operational principle, implemented repeatedly, with variations, in order to achieve a series of related but distinct functions.

[48] *Anomalies* in lunar theory are deviations between the Moon's expected position - were it to travel at a constant orbital velocity - and its actual position. The first anomaly corresponds approximately to Kepler's law of equal orbital area velocities. Since the moon's orbit is slightly elliptic, it moves faster when it is nearer the earth and slower when it is farther.

[49] These are 19 and 76 years respectively, each of these cycles contains an integer number of months (in contrast with a year which contains some non-integer number) these cycles are useful in the calibration of such a mechanism.

Exeligmos cycles[50]; provides for adjustment for leap years; and provides for a manual "prime mover" input on the side that causes all the dials to move at the correct relative rates (like the adjustment knob on a mechanical watch). Additionally, it is speculated that the mechanism may have also shown the positions of up to five planets.

Was the acquisition of all this functional information guided by the context information? Some of it was. The quest to discover and read inscriptions on the surface of the mechanism is evidence of this "main path" line of effort. However, the quest to accurately characterize the gears—tooth counts and relative locations—is evidence of a simultaneous feedback line of effort. The two lines of effort take place in parallel and the information discovered in either informs the interpretation of findings in the other.

In some cases it is difficult to make a distinction whether a bit of functional information owes to an understanding of the context, or of the mechanism's structure. For example, when the discovery of a 223-tooth gear suggested the presence of an eclipse prediction function—the counting of teeth is an interface-allocation stage activity that proceeds from an analysis of physical structure. But the number itself is only recognized as relevant in light of the understanding of the artifact's context.

Maybe a better description of the process is to say that the determination of context and purpose opened the door to function-discovery and interface-allocation information—but the predicted distinction or succession between those two was not observed. On the contrary, progress in this project required a continuous parallel effort and sharing of information between the two types of activity.

Finally, the boundary-breach stage played a role similar to the one predicted by the model. The collection of functional and structural information through external inspection and testing of the target system proceeded until the exterior of the target system "dries up" as a source of information. At that point a breach is undertaken.

There are some obvious differences with the other case studies. A partial physical breach of the mechanism took place early on and more or less unintentionally during the

[50] These are 223 months and 54 years respectively they are periods between eclipses (or more precisely, between identical arrangements of the sun-moon-earth relative positions).

initial restoration and cleaning. Given the frailty and archeological value of the target system, all subsequent "breaches" consisted in the subjection of the mechanism to imaging technologies. According to the model, the iterative process of reverse engineering results in the hierarchical breaching of progressively smaller components. In this case study there is a related but different progression: each subsequent breach gave access to improved resolution. The timing of the breaches was not intentional but simply conformed to the availability of new imaging technologies. Figure 55 shows the quality of information available to Price following the second system breach.

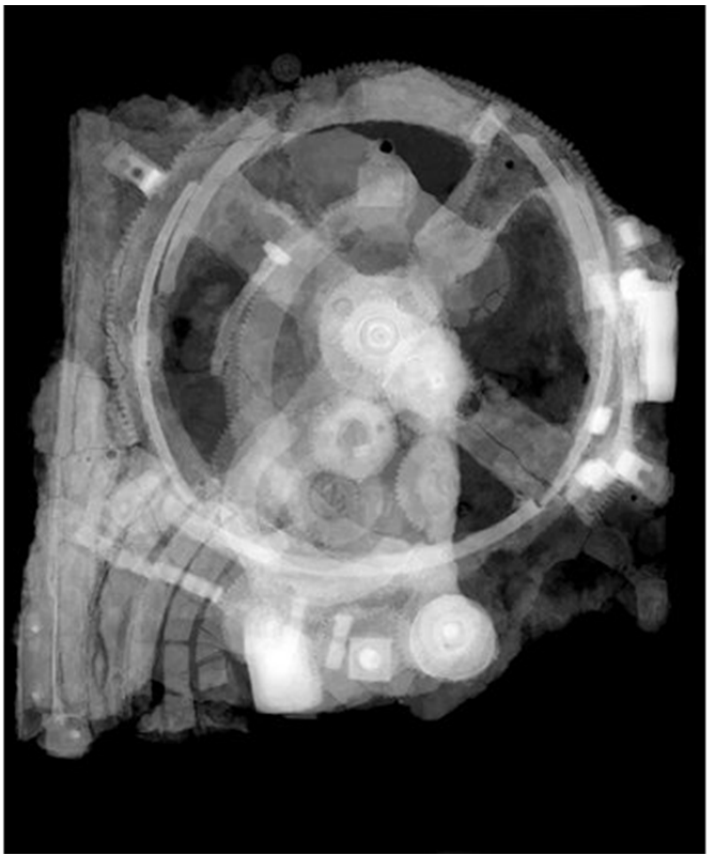

Figure 55. Antikythera mechanism—X-ray of Fragment A

Shown here, the quality of information available to Price. Source: Antikythera Mechanism Research Project, "Fragment A—Radiograph," Antikythera-Mechanism.gr, Retrieved from http://www.antikythera-mechanism.gr/data/radiographs

The breaching efforts were aimed at revealing internal structural and functional information without altering the target system physically. To this end, increasingly penetrating and higher resolution technologies and techniques were used. Figures 56 and 57 show the quality of the data available to the most recent wave of researchers, who are currently working on the problem. With each inspection new physical features were uncovered. These in turn called for revisions of the initial model.

Figure 56. Antikythera mechanism—Fragment 19

Polynomial Texture Mapping (PTM)—A technique available to more recent researchers. Source: Antikythera Mechanism Research Project, "Polynomial Texture Mapping Fragment 19," Antikythera-Mechanism.gr, Retrieved from: http://www.antikythera-mechanism.gr/data/ptm/full-resolution-ptm

In spite of the protracted timeline involving the often-uncoordinated efforts of numerous researchers, the process that emerges from looking at this case is very similar

to the prototypical case study described elsewhere in this work. Researchers after the publication of Gears from The Greeks (Price 1974) often begin their work by claiming that Price's work, while seminal, was fundamentally flawed and they are therefore "starting over." It is contended here that the flaws are a lot less fundamental than these researchers make them out to be. Their "complete revisions" and "fresh looks" are better described as refinements, the overall operational principle having been established early on by Price.

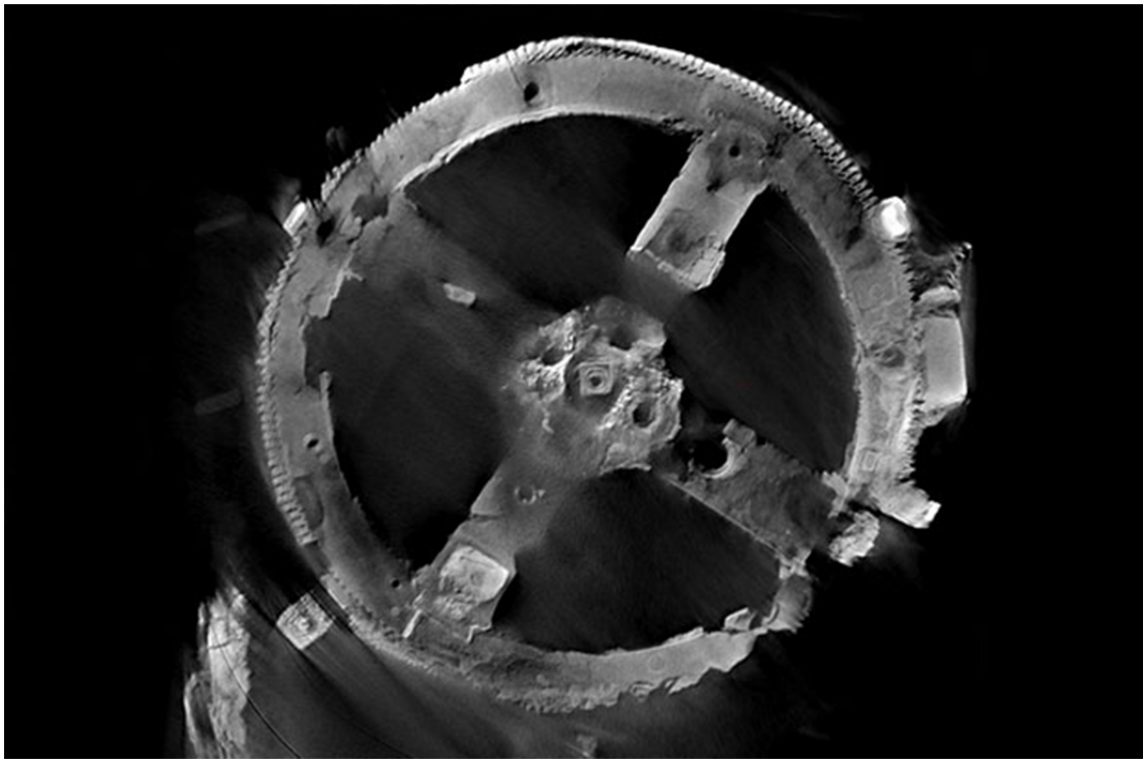

Figure 57. Antikythera mechanism

Fragment A under Micro focus X-ray Computed Tomography. The reader is reminded that this image corresponds to the same object shown in Figure 54. The apparent independence of the single gear is due to the tuning of the X-ray CT to a very specific depth. Source: Antikythera Mechanism Research Project, "Computed Tomography (CT)," AntikytheraMechanism.gr, Retrieved from http://www.antikythera-mechanism.gr/data/ct

b. Major Difficulties

The most obvious difficulty in this case of reverse engineering is presented by the gaps. There are two kinds of gaps. First, gaps in contextual information: due to the historical separation between the reverse engineers and the target system, there is simply

much that they do not know—and cannot know—about the target system's context. Second, physical gaps in the mechanism: if a single part is missing, it may be inferred accurately from the context provided by the rest of the system. If more parts are missing the prospects of ever arriving at a completely accurate characterization quickly shrink (it would be interesting to attempt to quantify the effect of missing parts upon the likelihood of success for a given project). On the other hand, the success in the application of imaging technology to fill in the gaps is noteworthy: 115 years after its discovery, new information is still being revealed as new technologies are brought to bear on the problem.

c. Price on Reverse Engineering

Given his prominent role in the reverse engineering of the Antikythera Mechanism, it was interesting to read Price's thoughts on discovering purpose and function of such a mechanism. Here is what he has to say about the problem and possible avenues to its solution:

> What is it? There are four ways of getting at the answer. First, if we knew the details of the mechanism, we should know what it did [its configuration and operational principle]. Second, if we could read the dials, we could tell what they showed [its function]. Third, if we could understand the inscriptions [equivalent to a user's manual], they might tell us about the mechanism [its purpose]. Fourth, if we knew of any similar mechanism, analogies might be helpful. All these approaches must be used, for none of them is complete. (Price, 1974)

In the case of the Antikythera Mechanism all these approaches must be used, for none of them is sufficient by itself

d. Heuristics

- H: Absence of evidence is not evidence of absence (Appendix B.V.5)
- H: Remove obstructions (Appendix B.V.6)
- H: Unanimous agreement can still be wrong (Appendix B.V.8)
- H: Sometimes "coincidence" is the most accurate explanation for a seemingly interesting circumstance (Appendix B.V.9)

- H: Look for information AROUND the target system (Appendix B.V.10)
- H: Make a drawing or build a model (Various)
- H: Identify the need for and type of expert assistance, and go recruit it (Various)
- H: Write down your theories and subject them to third party criticism (Various)
- H: Take notice of details—write them down (Various)
- H: Read the inscriptions and instruction manual (Various)
- H: Strive to achieve a solid base of knowledge that may be applicable to the target system (Various)
- H: What is this? There are four ways to get to an answer: (1) What does the thing do (externally), (2) What do its parts do (internally); (3) What does the supporting media say about it (inscriptions, manuals, etc.); (4) By analogy. You may need to use them all. (Appendix B.V.28)
- H: Look for an analogy from experience as a way to explain what you see (Appendix B.V.27-28)
- H: Sometimes a discovery that seems too exciting to be true, is (Appendix B.V.34)
- H: Be prepared to adjust your model to accommodate new findings (Various)
- H: But when a finding does not fit a particularly good theory—consider adjusting the finding (Appendix B.V.38-40)
- H: If you do not have the necessary tools or instruments, you may be able to build them (Appendix B.V.48)
- H: Excessive pride in the tools you have built may lead to unwarranted high estimates of their capability (Appendix B.V.49)
- H: Even if you discard an idea, record it (Appendix B.V.52)
- H: Leverage the latest technologies (if you can afford them) (Appendix B.V.59-60)
- H: It is easier to disprove an old theory than to come up with a suitable replacement (which is not intended to suggest that the disproving is not useful, only that it is the easy part) (Appendix B.V.108-110)

6. Case Studies VI-X (Virtual Case Studies Using Monterey Phoenix)

a. Virtual Case Studies—Introduction

A model of reverse engineering as an iterative process of exploration, discovery and synthesis was developed in chapters V and VI . In spite of its apparent simplicity, the proposed model encompasses a vast number of different scenarios. Each scenario is a product of variables introduced at the outset with the target system and context, and choices made by the reverse engineer as the task unfolds. For example, a discoverable interface may (or may not) be present in the first layer of the target system, and if it is present, the reverse engineer may (or may not) discover it while exploring the target system; the reverse engineer may (or may not) discover clues to the target system's function while exploring the context; when in possession of knowledge of an interface and its corresponding function, the reverse engineer may (or may not) see the connection and successfully allocate the interface to the function; and so forth. At each point in the process where there exists a variable, there is a branching of possible scenarios. The reverse engineering process then is something like a tree, each branch a unique scenario. The behavior of the model is the shape of this tree: it is defined as the set of all possible reverse engineering scenarios implicit in the model.

As discussed in Chapter IV, a visual representation of the reverse engineering model (as in Figure 58) offers a number of advantages. Chief among these, is its ability to present a lot of information in a very compact form, summarizing a great deal of information. However, the very compactness of the visual model makes it difficult to say with certainty whether its implications are not in conflict with reality. In other words, the challenge is to validate the model. Validation calls for testing the model in order to expose possible inconsistencies. One means to expose inconsistencies is to test the model in the real world, through physical or historical case study analysis. As described in the previous section and Appendix B, the resulting observations must fit the model, or else the model must be modified and reconciled to the observations.

However given the limited number of real world case studies examined, a second source of validation was needed. One option considered was that of using additional

historic cases, similar to the Antikythera case study. However, while historical cases with abound, most of the information related to these is anecdotal. Historical cases with the requisite amount of detailed information to infer the reverse engineering process were not found. Therefore, virtual case studies were used.

A different kind of supplemental verification of a model can be achieved by testing the logic of the process itself, exposing all possible of outcomes of the model (i.e., making explicit the combinations of events that are only implicit in the visual model of the process). An approach known as virtual experimentation may be used to expose logical inconsistencies hidden within the assumptions that underlie the model. While such a simulation or virtual experiment cannot truly validate a model (i.e., it cannot prove that the model applies in the real world), it can nevertheless provide valuable verification of the model's assumptions and internal logic. This verification can supplement the validation by physical experiment and historical case study analysis.

The modeling language known as Monterey Phoenix (MP) is a language for system and software architecture and business process simulation. In other words, the purpose of MP is to make explicit all the possible scenarios implicit in a given system or process model. To achieve this expression, the model must be specified in terms of the events to which it may give rise. Events can influence each other and can be related temporally (A precedes B) and hierarchically (A contains A.1 and A.2). Events can also be specified as either necessary, or optional (A will happen vs. A may happen). Once a model is fully specified in terms of events, the resulting MP code can be executed to generate an exhaustive list of scenarios in a series of graphical representations known as "event traces."[51] Each event trace represents a different scenario that is "possible" in that it satisfies the relationships and constraints that have been formally defined (Giammarco, 2014; Auguston 2014). Thus, an MP process simulation can validate a model by exposing all possible scenarios at the specified scope, allowing them to be studied and "reality-checked." To this end, the model presented in Chapter VI was *translated* into MP.

[51] The execution is accomplished through the MP analyzer which can be accessed online at http://firebird.nps.edu/

The MP model of reverse engineering yielded 368 event traces. In spite of their large number, the visual nature of the event traces greatly facilitated the inspection of all 368. Specifically, each of the 368 scenarios can be categorized under one of only five different patterns. In turn the patterns (literally, the shapes of the traces) correspond with general types of scenarios. Four of these scenarios involve some form of failure while the fifth represents the success of the reverse engineering endeavor.[52]

Five event traces (one from each pattern or family) are examined in Appendix C. The result provides additional confidence in the model for several reasons described in the following section.

b. Virtual Case Studies—Additional Validation of Model

The presentation of the reverse engineering process as a visual model (as shown in Figures 37, 54, and others) conveys the process in a clear and compact way. However, the visual model on its own offers little assurance of the logic that runs through it, or the soundness of the assumptions on which it rests. In contrast, specifying the reverse engineering process in a formal language called for a fresh look at the assumptions behind the model whilst maintaining a rigorous adherence to logic (otherwise the outcome would be nonsense, or it simply would not compile). Thus, the very act of subjecting the model of reverse engineering to formal specification in MP provided additional confidence that the model is valid.

Executing the MP model of reverse engineering produces 368 event traces. Each event trace is a virtual case study showcasing a scenario that is possible within the assumptions of the model. In theory, some of the 368 virtual case studies might expose critical flaws in the model, either by presenting scenarios that were not logical, or consistent with the real-world, or by failing to generate scenarios that were observed during real-world case studies. All 368 event traces were inspected. None of them presented a scenario that was illogical or inconsistent with reality. Instead, some of the scenario variants generated by MP yielded unexpected insights about possible

[52] In the real world, a persistent reverse engineer may overcome the "failure scenarios," as they result from flawed, but (in the real world) not irreversible actions or choices.

implementations of certain parts of the reverse engineering process. However, these insights do not impact the abstract reverse engineering process model, but bring to light some interesting exception cases. For example, one event trace[53] describes a scenario in which the reverse engineer failed to formulate a true working model in spite of the fact that she had obtained sufficient (accurate and complete) information infer a true working model. This scenario—once exposed—was obvious (of course a human reverse engineer may fail, even when all the precursors for success are present). However, the scenario was never considered until it was exposed as a virtual case study. The result of using Monterey Phoenix was that the model was both validated, and more completely understood. Furthermore, the outcomes of the scenarios generated by the process simulation were consistent with the outcomes suggested by the model. For example, the modes of failure introduced in Chapter VI are made explicit in many of the event traces generated by the process model.

Finally, the code developed to specify the reverse engineering process in MP provides a tool for further scientific exploration into the subject of reverse engineering. This is a key contribution that will enable future researchers to explore the validity of the simplifying assumptions incorporated into this model, among other things. They will also be able to consider case studies observed in the real-world, in light of the formally generated scenarios. Because MP contains no information about the probability of a given scenario, this will prompt the asking of interesting questions seeking to understand why some scenarios (and families of scenarios) take place with regularity while others manifest less often, or not at all.

A detailed description of the MP specification of the model of reverse engineering is provided in Appendix A. This includes a description of the assumptions, the structure of the model, analysis of the findings, and the MP code itself. The event traces for five scenarios (out of 368 generated by MP) as well as a narrative description of each (presenting them in a format consistent with the presentation of the real world case studies) are found in Appendix C.

[53] Event Trace #128 of 368—it is described in further detail in Appendix B.

THIS PAGE INTENTIONALLY LEFT BLANK

VIII. RESULTS AND CONCLUSIONS

> The machine does not isolate man from the great problems of nature but plunges him more deeply into them.
>
> Antoine de Saint-Exupery—*Wind, Sand, and Stars*

A. GENERAL FINDINGS

The proposed model describes reverse engineering as an iterative process composed of four stages. The implied notion that the four stages and their subsequent iterations are clearly sequential and distinct is not consistently validated by the case studies. In the first four case studies, the context-exploration stage was generally tacit. The system's purpose, context and boundary were known before the start of the project. However, in the fifth case study where the context of the target system was distinct from the context of the reverse engineer (the two are separated by 2000 years), the distinction between the context exploration stage and the following stages was clear.

The target systems tackled in the first four case studies were relatively simple. In all five case studies the supposedly iterative aspect of the reverse engineering process was difficult to observe because following the initial system breach, there were usually few if any remaining subsystems that may be subjected to the next iteration of inspections, analysis and tear-down.

The function-discovery and interface-allocation stages were sometimes indistinguishable from each other, they overlapped, or at least they happened in close interdependence. Often the discovery of an interface and the discovery of a flow or interaction through that interface tended to be indistinguishable events.

The reverse engineering process model presented in Figure 30 and subsequent updates, is potentially misleading because it suggests each of the four stages is more or less the same kind of thing. This is not the case. The first three stages are information-focused stages. The second iteration of the context-exploration stage can feed back to the interface-allocation stage from the previous iteration (i.e., the study of how the components relate to each other can provide clues that may lead to the discovery of

system-level functions that previously went unnoticed). The boundary-breach stage is qualitatively different, it is concerned with action.

The proposed model of reverse engineering did not account for the ideas that the reverse engineer brings to the table. It is given that a reverse engineer can—and should—have a broad base of engineering knowledge. However, through the case studies it became evident that the reverse engineer has a specific set of ideas (whether consciously held or otherwise) of how the system works. As the process of reverse engineering develops, this working model both informs and is informed by the discoveries made about the actual target system. A working model is not a passive recipient of updates, but plays an active role in guiding the process. The role of the working model can be positive: as it can prime the reverse engineer's mind to recognize expected functions and objects, thus achieving these goals quicker than otherwise. On the other hand, the working model sometimes plays a negative part, as it blinds the reverse engineer to the presence of the unexpected. A working model introduces new modes of failure. Figure 58 is an attempt to update the model presented earlier in such a way that it highlights or makes explicit the points mentioned in the preceding paragraphs.

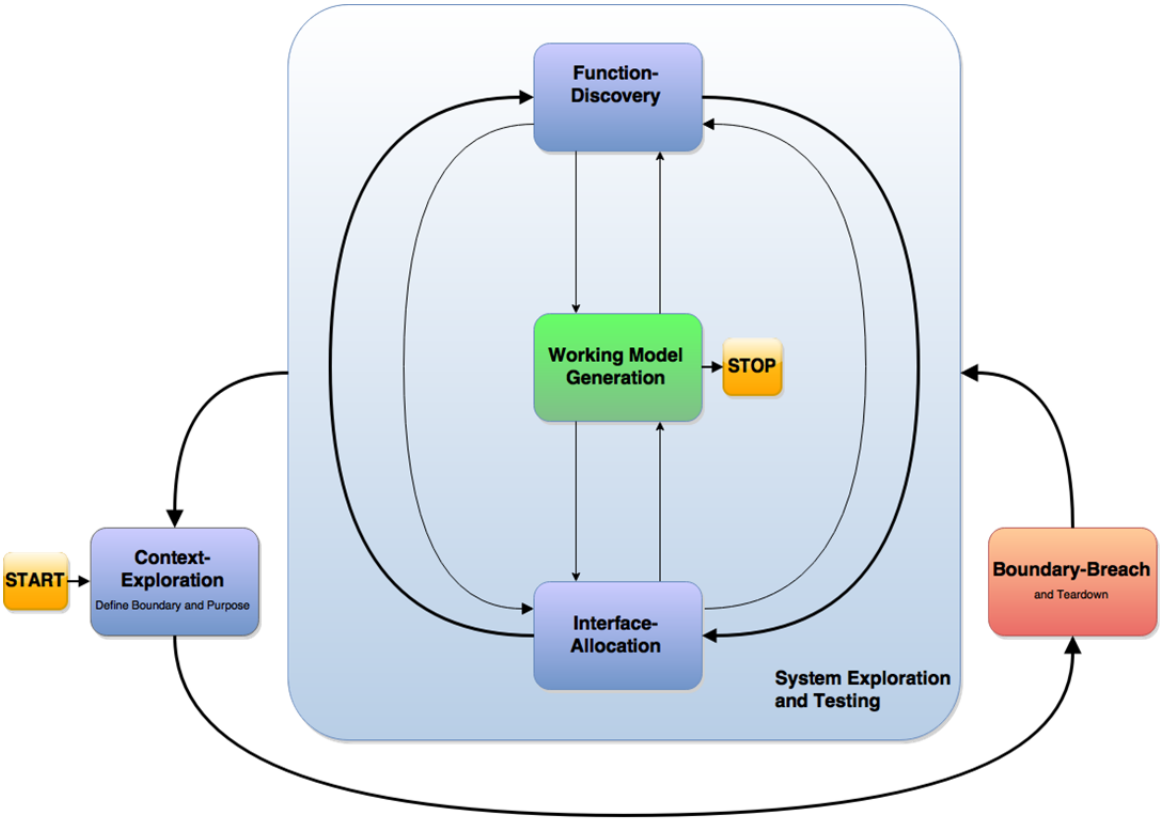

Figure 58. Updated (final) reverse engineering process model

Highlighted: (1) The central role of the reverse engineer's working model; (2) The qualitative distinction between the information-centered stages and the boundary breach; (3) the close relationship between function discovery and interface discovery; (4) The normal starting point (exploration of context) and ending point (following an update to the working model)

A working model will have a number of important characterizations such as: strength (the model may be firmly believed in, or not), clarity (the model may be clear and in the "foreground" of the reverse engineer's mind, or only vague and in the background), and accuracy (the model may reflect reality, or it may be wrong). These characterizations are probably not binary, but exist in different gradations and combinations for a particular reverse engineer and target system. The degree to which these variables affect the performance of the reverse engineer poses an interesting question for further research.

B. A LIST OF REVERSE ENGINEERING HEURISTICS

The following is a list of heuristics applicable to reverse engineering. Each of these heuristics proceeds from one or more of the following sources: Discovery during review of the technical literature and found relevant; Inference from the model of reverse engineering derived in this dissertation; Observation in the course of the case studies. The heuristics have been grouped into 5 categories: preparation, context-exploration, system exploration and testing, breaching and tear-down, and general good sense.

1. Preparation

- H: Bring the right tools (specialized tools may be necessary), instruments, cleaning materials, camera, spudger (Appendix B.III.18) (Appendix B.V.59-60)
- H: Bring the necessary supplementary material—get it (Appendix B.III.2)
- H: Look for information AROUND the target system (Appendix B.V.10)
- H: Identify the need for and type of expert assistance, and go recruit it (Various)
- H: Strive to achieve a solid base of knowledge that may be applicable to the target system (Various)
- H: If you do not have the necessary tools or instruments, consider building them (Appendix B.V.48)
- H: Before opening the system, make a diagram of what you expect to find (Appendix B.III.13)
- H: If practicable, become familiar with the target system as a user or consumer. Barring this, learn what users and consumers value in similar systems

2. Context-Exploration

- H: What is this? There are four ways to get to an answer (and you may need to use them all. (Appendix B.V.28)):
    - What does the thing do (what is its output)?
    - What do its parts do (open it up and look)?

    - What does the supporting media say about it (inscriptions, manuals, etc.)?
    - Are there any analogies that apply? (Appendix B.V.27-28)
- H: Make a hypothesis, write it down, subject it to third party criticism (Various)
- H: Draw a figure (Polya, 1973, 99)... Revise your figure when the information changes. (Appendix B.III.13)
- H: Sometimes you have to gather new questions before you can get any good answers (Appendix B.III.6-15)
- H: Absence of evidence is not evidence of absence—especially when the specimen is in a deteriorated condition (Appendix B.V.5)
- H: Sometimes "coincidence" is the most accurate explanation for a seemingly interesting context (Appendix B.V.9)
- H: Be prepared to adjust your model to accommodate new findings (Various)
- H: But when a discovery does not fit any good theories—consider adjusting the finding (Appendix B.V.38-40)
- H: Be wary of your working model. Think of ways to disprove it. Make your assumptions explicit and clear. Then challenge them. Aka use the scientific method

3. System Exploration and Testing

- H: Remove and Operate to find the function or relevance of a component (Otto and Wood) [B.I.8]
- H: Experiment to see more clearly—You can do this by:
    - Using repetition [B.I.15]
    - Removing opacity [B.I.13]
    - Enhancing contrast [B.I.23]
    - Changing the speed [B.I.15]
- H: Disassemble-reassemble-repeat [B.I.28 through the end]
- H: Apply the principle of *maximum component utilization*: assume all salient features serve a function and need to be explained (Ridder 2007,

241) (But remember that not everything has a function that is relevant to your project)

- H: Use all your senses to see (Appendix B.II.43)
- H: Shift perspective. Continuously turn over the target system. New angles sometimes reveal the true nature of (or way into) a component. Look at it from the point of view of the user. From the point of view of the maintainer.
- H: Certain things may be harder to notice once the system is running: see what you can figure out before you operate the system (Appendix B.II.11)
- H: Name objects for their generic functions (Sickafus, 2004, 83)
- H: Identify each state or mode of operation for the system (Appendix B.III.1)
- H: The, deliberately observe the system during each state or mode of operation (Appendix B.III.6-15)
- H: Think of and do experiments to see more clearly (Appendix B.IV.14-22)
- H: Think of and do experiments to see what if...? (Appendix B.IV.14-22)
- H: Break down a question (what are the functions?) into its parts (Appendix B.I.11)
- H: Make a drawing (or build a model)
- H: Partition the target system into distinct flows and/or events
- H: You do not really understand the system until you can bring it all together (this is easy to forget). If you ca not draw the target system, you probably do not understand it
- H: Ask yourself—what were the use-cases that guided the design of the target system. Explore plausible use-cases experimentally before starting the tear-down

4. Breaching and Tear-Down

- H: Pull, probe and poke to elicit information (Appendix B.II.8 and others)
- H: When attempting to breach a system it is often useful to apply force up to but just below the point of breaking, this helps expose seams. (Appendix B.II.26)

- H: More often than not, the system boundary breach will not proceed as planned. The most common culprits are: first, a missed or hidden fastener. Second, a part held in place by pressure fitting (just pull harder). Third, an inconspicuous part is doubling as a fastener, like a puzzle.
- H: Small details that do not make sense probably reflect a disproportionately large error in the working model (Appendix B.II.44)
- H: Look for seams. If you do not see any, remove something. Look again. (Appendix B.II.23)
- H: Just because some destruction is inevitable it does not mean that the system cannot be reassembled and returned to full operation (Appendix B.II.29 and 45)
- H: Breaching the system can (often does) in the loss of information (i.e., tiny or spring-loaded components). Beware of stored potential energy. Use external cues to anticipate the presence of springs.
- H: Take photographs during disassembly—they may prove useful during assembly (Appendix B.II.45)
- H: Do not be too eager to jump into the tear-down (Appendix B.IV.21)
- H: Even the most challenging system breach problem has (so far had) a solution (Appendix B.IV.23-26)
- H: During a tear-down when no productive next step is obvious, an unproductive next step will often expose a productive next step (Appendix B.IV.23)
- H: Remove obstructions (Appendix B.V.6)
- H: After you are done, reassemble the system and observe it again—you will see new things (Appendix B.II.46)
- H: Irreversible assembly need not result in destructive disassembly: Think sharp knife, chisel, dremel, heat, chemistry
- H: Slow down. When disassembling a system, do not outpace your grasp of how the various parts interact. Hesitate before starting the system breach: have I learned everything I could from the operational system?
- H: Difficult disassembly might be evidence of special boundary function and merits its own efforts at understanding: what is the purpose of making tear-down so difficult?

5. General Good Sense

- H: Excessive pride in the tools you have built may lead to unwarranted high estimates of their capability (Appendix B.V.49)
- H: Even if you discard an idea, record it (Appendix B.V.52)
- H: When you think you are done you usually are not (continuing to think about the problem, even after the project is over will yield new insights)
- H: It is easier to disprove an old theory than to come up with a suitable replacement (which is not intended to suggest that the disproving is not useful, only that it is the easy part) (Appendix B.V.108-110)
- H: Backtrack when necessary. This could mean go back (mentally) to take a closer look at an earlier assumption, or go back (physically) by partially re-assembling the system and re-examining it
- H: Unanimous agreement can still be wrong (Appendix B.V.8)
- H: Work with the end in mind: What are you trying to answer. Of a planned course of action ask: how does this help me achieve my end?
- H: Beware of bias for confirmation: we are "hard-wired" to confirm rather than refute our initial errors (Kahneman 2011, 80–88)

C. FINAL THOUGHTS

From a synthesis of the available technical literature, a clear and general definition of reverse engineering was arrived at. According to this definition, *reverse engineering is the problem-solving activity that ensues when one takes a human-made system, whole or in part, and attempts—through systematic analysis of its physical characteristics and other available evidence—to answer one or more of the following questions: What is this for? What does it do? How does it do it? What is inside it?* A new way to represent systems was developed: The System Diagram for the Analysis of Reverse Engineering or *system diagram* for short. The definition and system diagram were used as the basis for developing a model of the process of reverse engineering. According to this model, reverse engineering is an iterative process of information-gathering composed of four stages that are repeated as the process *drills-down* from system to subsystem, to component, to subcomponent, and so on. In each stage, the reverse engineer's attention and activity are directed toward obtaining a particular type of

information, or solving a particular type of problem. As he or she transitions between stages, the reverse engineer may incur a mode of failure whereby some of the information from the previous stage does not make it across the transition. By suggesting a general way to think about reverse engineering and its possible modes of failure, this model can serve as a tool to infer the incidence of heuristics and a framework to interpret real world reverse engineering activity. The model was applied to four simple case studies one complex historical case study. The model was also encoded in an executable formal language in order to generate additional simulated case studies of which five were studied. The outcomes were: a partial validation of the original model; a collection of reverse engineering heuristics; and an improved model. Finally, the model developed here offers a hitherto nonexistent descriptive language and framework to think about the process of reverse engineering in general (as opposed to its narrow application in software and somewhat less narrow application in computer hardware).

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. FORMALIZING THE MODEL OF REVERSE ENGINEERING USING MONTEREY PHOENIX

A. ASSUMPTIONS

For practical reasons, the reverse engineering model translation into MP contains some simplifying assumptions. The most important of these is the assumption of a *one-layer target system*. An important characteristic of the process of reverse engineering described in preceding chapters is the iteration of the process through increasingly deep layers of system structure and function (like the peeling of an onion). The patterns of exploration, discovery, and synthesis employed at the system-level, are repeated at the subsystem level (for each subsystem), and in each subsequent layer until all the available information is extracted (or until the reverse engineer reaches some terminal point of failure in his or her attempt to do so). Were it to be incorporated into the MP model simulation, this iteration would yield an unmanageably large number of possible scenarios for little expected gain. Thus, the target system as modeled in MP contains a single layer of structure/function.

B. DERIVATION

On the surface, the event-based specification required by MP and the result consisting of 368 event-based visual presentations appear very different from the model of reverse engineering that has been presented thus far. Nevertheless, every aspect of the MP-specified model can be found either in Figure 19 (which introduces the relationship between target system, context, interface and function), Figure 38 (which introduces the variety of possible modes of failure), or Figure 58 (which shows the final version of the model). Some aspects of the model do not make it into the MP specification for reasons that will be explained shortly. The essential aspects that do make it into the formal specification are as follows: (1) There are three fundamental actors (reverse engineer, target system, context); (2) There are two essential forms of information implicit in the target system and context (interfaces and functions). It is one of the reverse engineer's objectives to make both of these explicit through exploration and testing; (3) There are

two ways in which information can "fail" (it can be incomplete or incorrect); (4) Obtaining correct and complete information about interfaces and functions is the essential first step that will be followed (if all goes well) by allocation, and later by the formulation of a working model.

C. THE MP MODEL OF REVERSE ENGINEERING

In an MP implementation of the model, the existence of a fundamental actor is defined as a ***root event***. The first step in specifying the reverse engineering model in MP is the definition of root events. The reverse engineering process presented in this work involves three fundamental actors: a reverse engineer, a target system, and a context. Each root contains a number of events and subevents as shown in Table 4. All events and subevents are *latent* in the MP model. As the reverse engineering process unfolds according to the model, some events remain tacit, while others come to the "surface" and influence others.

Table 4. Inclusion relationships of root events, events, and subevents

Root Event	**Event**	**Subevent**	**Sub-subevent**
Reverse Engineer	Exploration and testing	ALWAYS studies context structure	SOMETIMES discovers a system function in the context
		ALWAYS studies target system structure	SOMETIMES discovers a function (or a false function)
			SOMETIMES discovers an interface (or a false interface)
	Attempts to allocate:	SOMETIMES results in a true allocation	
		SOMETIMES results a false allocation	
		SOMETIMES results in no allocation	
	Attempts to formulate a working model	SOMETIMES results in a true working model	
		SOMETIMES results in a false working model	
		SOMETIMES results in no model	
Target System	SOMETIMES contains a function (or a false function)		
	SOMETIMES contains an interface (or a false interface)		
Context	SOMETIMES contains a system function in the context		

Not shown are the conditional relationships that exist across different root events. For example: if no allocation happens, a subsequent attempt to formulate a working model will result in no working model; if a false allocation happens, it may be followed by either a false model or no model, but not by a true model; an allocation must be preceded as a minimum by discovery of a function and discovery of an interface; however, when the function, the interface, or both function and interface are false, the subsequent allocation must be false.

1. The Reverse Engineer

The events under the reverse engineer root consist of three distinct activities: (1) exploration and testing—this involves physical actions like probing, measuring, pushing buttons, and turning the target system over to consider it from different angles. The goal of exploration and testing is to bring the reverse engineer into contact with discoverable elements in the structure of the target system or context leading to the discovery of functions or interfaces. The outcome of successful exploration and testing is in the form of unconnected information about interfaces and functions; (2) allocation—this takes place when the reverse engineer has collected enough information to hypothesize connections between functions and interfaces. The outcome of successful allocation is in the form of ***this*** *[interface] performs* ***that*** *[function]*; (3) working model generation—this takes place when the reverse engineer appeals to engineering principles and the laws of physics as the basis for a physical explanation. A successful working model expands upon the allocation to provide a hypothesis in the form of *this is* ***how*** *this [interface] performs that [function]*. A working model that is true, represents success of the reverse engineering efforts insofar as the given interface and function for which this hypothesis is formulated.

2. The Target System

This is the technological physical system under consideration by the reverse engineer. The MP version of the model makes explicit the fact that the target system root has three events: (1) It always contains some structure which is the object of the reverse engineer's exploration; (2) In that structure there may be contained discoverable functions—that is, the system may be partly or fully operational, meaning the reverse engineer can sometimes get the system to *do something* (or he can infer what the system does from its physical characteristics). An observed or inferred operation that is in accordance with the target system's design is a function. Otherwise, it is a false function or a red herring; (3) The structure may also contain discoverable interfaces. These are physical features of the target system that deliver energy, material, or information into the context. When this interaction occurs in accordance with the target system's design, the

physical feature is an interface. A physical feature of the target system that does something it was not designed to do, or that appears to be functional but is not (like a skeuomorphism) is a false interface, a different type of red herring.

3. The Context

This is the environment in which the target system functions (or is believed to function). In a sense, the context contains the target system. However, the model was developed assuming that one of the first steps taken by a reverse engineer, is the purposeful excision of the target system from its context in order to conduct careful and controlled "bench" exploration whenever possible. For this reason the target system root event is not modeled in MP as related by inclusion to (contained in) the context.

Yet, the reverse engineering model is also based on the assumption that context remains a potential source of valuable information to a reverse engineer. For example, context may contain other systems (or connections to other systems) that are recipients or suppliers of energy, material, or information flows that may offer important clues as to the target system's purpose or function. Functions flow across interfaces between target system and context. As a simplifying assumption, the model excludes the possibility of interfaces in the context.[54] Also, the analysis of context is part of the iterative process of reverse engineering: each level of exploration terminates in a boundary breach that transforms the system at one layer into the context for the subsystems at the next layer. However, this aspect of the process is not shown in the MP implementation due to the one-layer target system assumption. Thus, the context root event has two events: (1) It always contains some structure that is the object of the reverse engineer's exploration; (2) In that structure there sometimes exist discoverable clues as to the function of the target system.

[54] This is a reasonable assumption because the target system is generally designed to fit the context and not the other way around. Eliminating the possibility of finding interfaces in the context significantly reduces the number of event traces that will be generated and need to be analyzed. A subsequent version of the MP analyzer will incorporate assertion checking capability, making the inspection process semi-automated and obviating the need for this simplification.

4. Event Grammar

MP offers a series of conventions that can be used to implement a sequence of events consistent with constraints of logic. In other words, MP employs a formal event grammar to specify the hierarchical, temporal, and necessity characteristics of each event. Detailed explanations of all the event grammar conventions used in the MP model of reverse engineering are provided in the notes within the MP code in the following section. Key elements are described in the next section.

5. Order and Necessity

The sequential relationship and necessity of each event must be specified. The logic of the reverse engineering process requires some events to manifest in a fixed order, while others may manifest in any order or even simultaneously. For example, the three events under the reverse engineer root are exploration and testing, allocation, and formulation of a working model. Each of these is contingent upon the results of the previous event. Thus, exploration and testing, allocation, and model formulation must take place in that order. On the other hand, the three subevents under exploration and testing are independent of each other. Thus, discovery of a function in the context, discovery of a function in the target system, and discovery of an interface in the target system, may manifest in any order. The may also not manifest. That is, the occurrence of a given event can be necessary or optional. For example, the event wherein the reverse engineer studies target system structure is a necessary event—without it there is no reverse engineering. On the other hand, the event where the reverse engineer discovers an interface in the target system is an optional event. Order and necessity are implemented in MP through the use of different types of parentheses or brackets that delimit the sets, and different types of spacers between (commas or spaces) that separate the events within a set.

6. Coordinate and Share All

Some events in the reverse engineering process are conditionally related to other events. For example, the reverse engineer must possess knowledge of at least one function (deduced either from the exploration of context or target system) and one interface (deducible only from the target system) before she can venture an allocation

(connecting the two). This logic of the type *A and B must exist before C can exist* can be implemented by the **COORDINATE** composition operation which filters out event traces with logically inconsistent events within a root.

Some events have subtly different meanings under different roots. For example, consider the event "Function A found in target system." The inclusion of this event under the Target System root, conveys the fact that the target system may **contain** Function A. On the other hand, the inclusion of this event under the Reverse Engineer root, conveys the fact that the reverse engineer may **discover** Function A in the target system. Only when both meanings of the event "Function A found in target system" come together can the event actually occur. For example, if there were an event trace where "Function A found in target system" appears under the Reverse Engineer root, but not under the Target System root, this would indicate a logically incongruent scenario where the reverse engineer found something in the target system that does not exist in the target system. Conversely if an event trace were to show "Function A found in target system" under the Target System root, but not under the Reverse Engineer, this would indicate a logically incongruent scenario where Function A has become explicit in the process of reverse engineering, but the reverse engineer had nothing to do with it. The **SHARE ALL** composition operation is used to filter out these types of logically inconsistent event traces.

The actual model of reverse engineering implemented in MP, incorporating all the elements described above, is shown in Table 5, along with explanatory commentary.

Table 5. Formal specification of reverse engineering model using Monterey Phoenix

```
SCHEMA REVERSE_ENGINEERING_2

/* The CONTEXT:
1. always contains some structure which is the object of the reverse engineer's exploration
2. sometimes contains discoverable[55] system function.
*/

ROOT CONTEXT: context_structure [function_in_context];

/* The REVERSE ENGINEER:
1. always explores and tests
2. always attempts allocation
3. always attempts to generate a working model
*/

ROOT REVERSE_ENGINEER: explores_and_tests attempts_allocation attempts_to_generate_working_model ;

/* exploration and testing:
1. always studies context structure
1.1 sometimes discovers a system function in the context
2. always studies target system structure
2.1 sometimes discover a function (or a falsefunction)
2.1 sometimes discover an interface (or a falseinterface)
*/

explores_and_tests:
{           (context_structure [function_in_context]),
(system_structure   {           [ (function_in_system | falsefunction_in_system)],
                                  [ (interface | falseinterface)]
                                  }
)
} ;

/* attempts to allocate: can result in a true allocation, a false allocation, or no allocation
*/

attempts_allocation: (true_allocation|false_allocation|no_allocation) ;

/* attempts to generate a working model: result in a true working model, a false working model, or no model
*/

attempts_to_generate_working_model: (model_is_true|model_is_false|no_model);

/*TARGET SYSTEM:
1. always contains structure which is the object of the reverse engineer's exploration and testing
2. sometimes contains discoverable system functions and also false functions
3. sometimes contains discoverable interfaces and also false interfaces
```

[55] Although not part of the model, an "undiscoverable" system function would be one for the discovery of which the reverse engineer lacked the requisite instrumentation. For example if the function involved neutron emission, but the reverse engineer had no instrument capable of observing or measuring such an emission.

```
*/

ROOT TARGET_SYSTEM: system_structure
                                    {       [(function_in_system| falsefunction_in_system)],
                                    [(interface | falseinterface)]
                                    };

/* Allocation:
1. an allocation is essential for the reverse engineer to progress to the model generation stage.
2. an allocation requires two components: a function and an interface
3. a true allocation can take place following the discovery of a real function (from CONTEXT or TARGET SYSTEM) and a real interface
4. a false allocation results where one or both of the components of the allocation are false*/

true_allocation:    ( {function_in_system, interface} |
                      {function_in_context, interface}
                    );
false_allocation:   (       {function_in_system, falseinterface} |
                            {function_in_context, falseinterface} |
                                    {falsefunction_in_system, interface} |
                                    {falsefunction_in_system, falseinterface }
                    );
/* A discovery has two essential parts: a) the thing to be discovered, and b) the discovery event itself. For example, function_in_system (optional) under TARGET_SYSTEM means: target system may contain a function. On the other hand, function_in_system under REVERSE_ENGINEER means: reverse engineer may discover a function. NOTE: the presence of the same event under two roots can cause problems. For example, an event trace showing function_in_system under Reverse Engineer, but not under Target System suggests a scenario where the reverse engineer found something that does not exist. Conversely an event trace showing function in system under Target System, but not under Reverse Engineer, suggests a scenario where something has become explicit, but not to the reverse engineer. The SHARE ALL composition operation eliminates these logically impossible scenarios (and similar ones) by ensuring that only traces that contain the potential information in the target system (or context) AND its potential discovery in the reverse engineer will actually manifest as event traces*/

CONTEXT, REVERSE_ENGINEER SHARE ALL context_structure, function_in_context;
TARGET_SYSTEM, REVERSE_ENGINEER SHARE ALL system_structure, function_in_system, falsefunction_in_system, interface, falseinterface ;

/* A false allocation precedes a false model*/

COORDINATE      $A: false_allocation FROM REVERSE_ENGINEER,
                $B: model_is_false FROM REVERSE_ENGINEER
DO ADD $A PRECEDES $B; OD;

/* No allocation precedes no model*/

COORDINATE      $A: no_allocation FROM REVERSE_ENGINEER,
                $B: no_model FROM REVERSE_ENGINEER
DO ADD $A PRECEDES $B; OD;
```

The colored text is automatically generated by the MP Analyzer to assist in tracking the different grammatical elements. The shaded text is non-executable commentary used by the author to explain the section of the code immediately following. Further justification of MP and explanation of its Grammar can be found in the Monterey Phoenix website (https://wiki.nps.edu/display/MP) and in Behavioral Modeling of Systems Architectures with Monterey Phoenix (Giammarco, Farah-Stapleton, and Auguston)

D. ANALYSIS OF THE MP MODEL

The MP model of reverse engineering results in the generation of 368 event traces. Each event trace is a virtual case study that may be used to help validate the model. The visual presentation of event traces lends itself to efficient visual inspection by comparison of patterns. It is relatively easy to grasp at a glance whether any two event traces have a similar shape or not. Event traces with similar shapes turn out to describe very similar case studies (often it is difficult to spot the difference). Event traces with dissimilar shapes turn out to describe very different scenarios. A visual inspection of the 368 virtual case studies revealed five different general patterns or shapes. Each general shape can be thought of as a family of scenarios. Each of the 368 event traces falls into one of these families. Four of these families correspond closely with specific modes of failure (discussed in Appendix C). The fifth pattern corresponds with a successful reverse engineering project. The five families are as follows:

1. Family # 1—Failure Due to Incomplete Information

The reverse engineer failed to discover all the information necessary to generate a working model (i.e., either a function, an interface, or both have eluded discovery). This situation makes the subsequent allocation impossible. Without an allocation, the reverse engineer cannot proceed to the formulation of a working model. An alternative way of understanding this scenario is by visualizing the requisite information (of interfaces and functions) as forming a path to be traversed by the reverse engineer. The path connects the starting point of the process (ignorance) to the end goal (a true working model). In the case of Family #1, the path is truncated and therefore the end goal is unreachable.

2. Family # 2—Failure Due to False Information

The reverse engineer incorporated some false information among his discoveries. This condition makes a false allocation inevitable (except when the reverse engineer fails to make an allocation altogether, which is always a possibility). A false allocation can only lead to a false working model (or to no model). In the case of Family # 2, a false path is available, however the path to the end goal is truncated, thus the reverse engineer may follow the false path for some time, but the goal remains unreachable.

3. Family # 3—Failure in Spite of Accurate and Complete Information

The reverse engineer discovered all necessary information, but nevertheless failed at some point subsequent to the discovery of information. In the case of Family # 3, the path is complete, but the reverse engineer strayed from it, and failed to reach her goal. There are several variations of this scenario:

A. The reverse engineer failed to make any kind of allocation. She discovered a function and the interface responsible for it, but did not grasp the presence of connection between the two.

B. The reverse engineer made an allocation of the form "interface X performs function F." However, while X is a true interface and F is a true function, the connection is not true: X does not perform F.

C. The reverse engineer failed to make any kind of model. She made a correct allocation but could not produce a model. The state of her understanding at the end of the process is thus "I know that interface X performs function F, but I do not understand how it does it."

D. The reverse engineer failed to make a true model. A correct allocation does not guarantee that a working model based on it will be true. Thus, in this case the reverse engineer offered an explanation of how interface X performs function F. However, while X does perform F, the reverse engineer's explanation of how this is done was not true. A failure of this type could result from an error in logic or from inadequate understanding of the operant physical laws or engineering principles.

4. Family # 4. Failure Due to False Information in Spite of Accurate and Complete Information

This family combines elements of the previous two: the reverse engineer discovered all the necessary information, she also discovered additional false information, a red herring. She went for the false information, consequently failing to reach a true working model.

5. Family # 5. Success

The reverse engineer discovered all the necessary information, used it to arrive at a true allocation, which led him to a true working model. The information path is complete and the reverse engineer navigates it to the end goal. This family contains a branch where the reverse engineer discovers just the right amount of information, another branch where she discovers redundant true information, and a third branch where additional but erroneous information is discovered, but the reverse engineer ignores it in favor of the true information.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. FIVE REAL WORLD CASE STUDIES

General Description: Four projects in reverse engineering were undertaken. The goal in each project was to discover functions of the target system and the parts and operational principles underlying these functions.

Constraints: The projects were to be undertaken without the aid of supplementary material. Supplementary material includes online information supplied by the manufacturer or a third-party, to include product reviews or descriptions of the use, maintenance, or tear-down of the system. In this case, the constraint was also applied to instructions or clues contained in the packaging with the exception of the product name and "subtitle" For example one project was labeled "*Hexbug Original: The Robotic Creature That Reacts to Touch and Sound*" In that case, the functions of sound and obstacle detection were given away by the title. In summary: the rule was: in so far as it is possible, information about the system was to be learned from the system itself.

Project Selection: Each of the four target systems was selected on the following basis (in relation to the author/problem solver):

A. The system standard configuration must be unknown (i.e., must possess no advance acquainted with the nature or arrangement of the internal components).
B. The operational principles underlying the system functions must be unknown (i.e., must be unable to answer "how does it work?")
C. The operational principles must be internal or at least partially hidden. (i.e., providing an answer to a. or b. should require physically opening of the system to look inside)
D. The operational principles should appear to have at least some complexity (measured subjectively).

Generation of the Transcriptions: The problem solver attempted to maintain a log to account for every conscious action, thought, and discovery made in reference to the target system during each project. Doing this required the problem solver to pause frequently to jot down thoughts, provide physical descriptions, and draw occasional pictures. In some instances the disassembly involved a relatively long sequence of straightforward steps. In these cases, all the obvious steps would be completed first before pausing to write down.

After all projects were completed, the notes were subsequently cleaned up to improve legibility. The actions, thoughts, and discoveries (in general all three can be collectively referred to as "events") are presented in the same sequence in which they were generated. Thus, the relative times of all events are preserved. On the other hand, the duration of events and intervals between them were not recorded. This is not considered a problem as the model of reverse engineering provided in this dissertation gives no basis for an analysis of durations and intervals.

A fifth Case Study. The final case study is not from a personal project in reverse engineering. Rather it is a synthesis of several works that describe the historical effort to reverse engineer an ancient artifact known as the Antikythera mechanism. The effort has involved over 50 researchers from across the world and has spanned (it is ongoing) over a century of work. While this is a fundamentally different type of case study, it is believed that the same process of reverse engineering that applies to the small projects is evident in the larger case study. The main reason for incorporating this case study is to offer some validation of the model in terms of scalability.

Organization/Labeling: This appendix is organized into five sections (B.I, B.II, B.III, B.IV, and B.V). Each section consists of a time-table where each line-item approximately corresponds with a subjective event. Each event is numbered. These event numbers will be referred to in the case study analysis and model validation portion of this dissertation.

For example the analysis may contain a description such as this: *"Phase III took place during B.II.12-23, however B.II.17 is a Phase I activity."* Interpretation: The author was engaged in the breaching of the second target system (Phase III) between the 12th and the 23rd events, however the 17th event pertained to ascertaining the purpose or context of the target system (Phase I).

Each event is also categorized as an action, thought, or discovery as follows.

A. Action: The author did something (most likely with his hands) to the target system
B. Thought: The author experienced a new mental state that did not appear to be directly tied to an observation (A thought will look like: I think..., I wonder...., I have a question)

C. Discovery: The author experienced a new mental state that is directly tied to an observation (A discovery will look like: I see..., I have found...., I have discovered)

B.I CASE STUDY I (FOAMING PUMP) [REFER TO FIGURES 39 AND 40]

1. [Thought] What is the purpose? It makes foam (Presumably, this is a way to dispense less soap in more volume… this makes it last longer and give the impression of being greater quantity).

2. [Thought] What is the context/interacting systems? Soap, operator (pushing), air (must somehow be added to the soap to make foam)

3. [Thought] Does the air come from outside or inside the bottle?

4. [Thought] The main mystery is: How are the air and soap mixed to make the foam?

5. [Thought] Hypothesis: the pumping action causes air (from the upper part of the bottle) and soap (from the lower part of the bottle) to be pumped simultaneously and at high speed into a mixing chamber.

6. [Thought] Hypothesis: Maybe air is pumped from outside the bottle into the bottle. This pressurization of the bottle drives air and soap into the mixing chamber.

7. [Action] A drawing: It shows two flow paths one (air) begins “above the waterline” and the other (soap) begins “below the waterline.” Each flow ends in a separate nozzle discharging into a common chamber (where the flows mix).

8. [Thought] If the motive force is the pressurization of the bottle, then the mechanism will not work if it is removed from the bottle. This suggests an experiment.

9. [Action] Removed pump mechanism from the bottle and pumped it several times (used sink water as the source of fluid)

10. [Discovery] As water replaces the soap inside the mechanism it is evident that the mechanism operation is unaffected (except for a gradual loss of foaminess as water replaces soap throughout the system’s internal chambers). The motive force is not the pressurization of the bottle.

11. [Thought] The main mystery is (a better definition) composed of two parts:

12. What is the motive force/mechanism

13. What is the mixing mechanism

14. [Thought] Thinking about the second part… it is clear that the foam requires air: Where is the air coming from? (It is clear that the soap is coming from a suction tube at the bottom of the bottle)

15. [Thought] While the mechanism is mostly transparent, the foam is opaque and it obstructs a clear view of what is going on inside. Flushing all the soap out of the mechanism and replacing it with water (which does not foam-up) should result in a clearer view of what is flowing when and where

16. [Action] The mechanism is flushed with water

17. [Action] Experiment: The mechanism is the operated repeatedly and at different speeds while looking closely at the flow of water and air

18. [Thought] Was it really an experiment? There was nothing that this experiment could prove or disprove… Perhaps there are two general types of experiments:

19. Experiments to test: If X works as I suppose, then doing A will result in B. Experiment: Do A to test my theory of X

20. Experiments to see more clearly: I do not know how X works, but if I do A I will be able to observe more clearly by isolating it, magnifying it, slowing it down, or some other manipulation. Do A to see X more clearly

21. [Discovery] There are distinct events going on during each stage of the pumping (i.e., the up and the down stage). It looks like this:

22. When the pump head is pushed down: Air is pushed from one inner chamber to another. The transfer appears to occur through a small orifice. As the receiving chamber is already filled with soap, the air transfer results in the foaming.

23. When the pump head is released and it moves up: Soap is suctioned up into the chamber. The chamber is primed for making of new foam.

24. [Action] A drawing: It shows the soap chamber relative to the nozzle and the pump's motion. When the pump is pushed, this forces one tube down into another tube of slightly bigger diameter. When the pump is released, the inner tube springs up sucking in new soap.

25. [Action] Close observation of the cycle continues

26. [Discovery] During the push stage there is no flow felt at the suction tube. During the release stage a suction is clearly felt (this complements the previous observation)

27. [Discovery/Thought] During the push stage, air moves from a lower chamber to a higher chamber adjacent to the nozzle (the same chamber that fills up with soap during the release stage). But how does the air get into the lower chamber?

28. [Thought] It is very useful that the whole mechanism is transparent… and yet, I have not figured it out!

29. [Thought] Possible experiment: Flush the mechanism with air. Then let the release stage take place with the bottle submerged (The idea is that this will force water to move through the path normally taken by air thus making it clear what that path is)

30. [Action] Perform the experiment just described. Several times. Alternating the flow of water either through the soap-path or the air-path. This improves understanding...

31. [Action] A drawing: It has 4 separate sections, 2 for air and 2 for soap each shows a the problem solver's current understanding of the path of that fluid through a part of the system during either the push stage or the release stage.

32. [Thought] Each drawing makes sense by itself. It is tempting to say that "I understand how this works." However, the 4 drawings do not fit together into a single a single coherent drawing. Trying to put it all together it becomes evident that there are several transitions not yet understood.

33. [Thought] To learn anything further will require tear-down of the pumping mechanism.

34. [Thought] Testing led to the conclusion that there are 2 different valves inside the mechanism. But the valves are not visible from the outside. It will be necessary to tear-down the mechanism in order to determine where these valves are and how they work.

35. [Thought] An ability to obtain a cleanly cut a cross-section of the pump would be more informative than the piece-by-piece disassembly! Because, as each piece is removed from the whole its operation will be disabled and to be inspected closely, it must be inspected out of context. A cross-section would solve this. Alas I do not have the tools to do a cross-section.

36. [Action] Disassembly begins. After several minutes of unproductive pulling and twisting, pliers are used to pull on a tube. This results in some destruction of the tube, but all the other components are unharmed.

37. [Thought] Expect to find two check-valves (i.e., one-way valves), of the type found in snorkels and other applications to prevent backflow against

small differential pressures (usually a flexible flap that can freely fold away from an orifice but cannot fold into the orifice)

38. [Action] Disassembly continues. Unlike the first component, all the others are yielding to moderate pulling. A ball-and-spring check valve is found at the bottom of the lowermost chamber (the spring pushes the ball against an orifice at the bottom of the chamber. If suction overcomes the spring force, soap can flow in. But nothing can flow out)

39. [Action/Discovery] One compound component contains at least one internal valve. By blowing/sucking on different parts of this component problem solver concludes that there are 2 additional check-valves inside it. One of these can be seen, it is a flexible flap check valve.

40. [Thought] The third valve can be inferred from the experiments, however it remains hidden inside the mechanism. Believe this valve is responsible for alternately allowing soap or air to flow into the mixing chamber.

41. [Discovery] The internal valve is actuated by a stem, however it displays unexpected flow characteristics: specifically, the valve allows flow through during push and blocks flow through during pull. In other words, the valves open/shut position is controlled by the direction in which the stem is moving rather than by the position of the stem (The problem solver does not know of a mechanism for a valve actuator can work based on direction of its movement rather than on its position)

42. [Action/Thought] Reassemble everything to see (once again) if the various interactions between valves can be untangled. The difficulty here might be called "function entanglement"

43. [Action] After several minutes of playing with the assembled mechanism, problem solver once again feels confident that full understanding has been reached. However, again is unable to draw a coherent picture.

44. [Action] A drawing: After several more minutes of playing with the mechanism: two new drawing. One is labeled "Push" the other "Pull." All internal chambers are labeled A thru E. Three valves are labeled x thru z. Adjacent to each diagram there is a description of the flows. For example: "Push Cycle/Air: Atm->A and B->C/Soap: D-> Atmosphere"

45. [Thought] Have full understanding! Wait… the drawing is incomplete. It has less chambers than are in the actual mechanism. Also, the action of the valves has not been shown clearly

46. [Actions] A drawing: Discard it. Draw again, this time using the same labeled chamber/labeled valve format with the flow descriptions on the side of each stage. The valve positions (open or shut) during each stage are also listed (Figure 59)

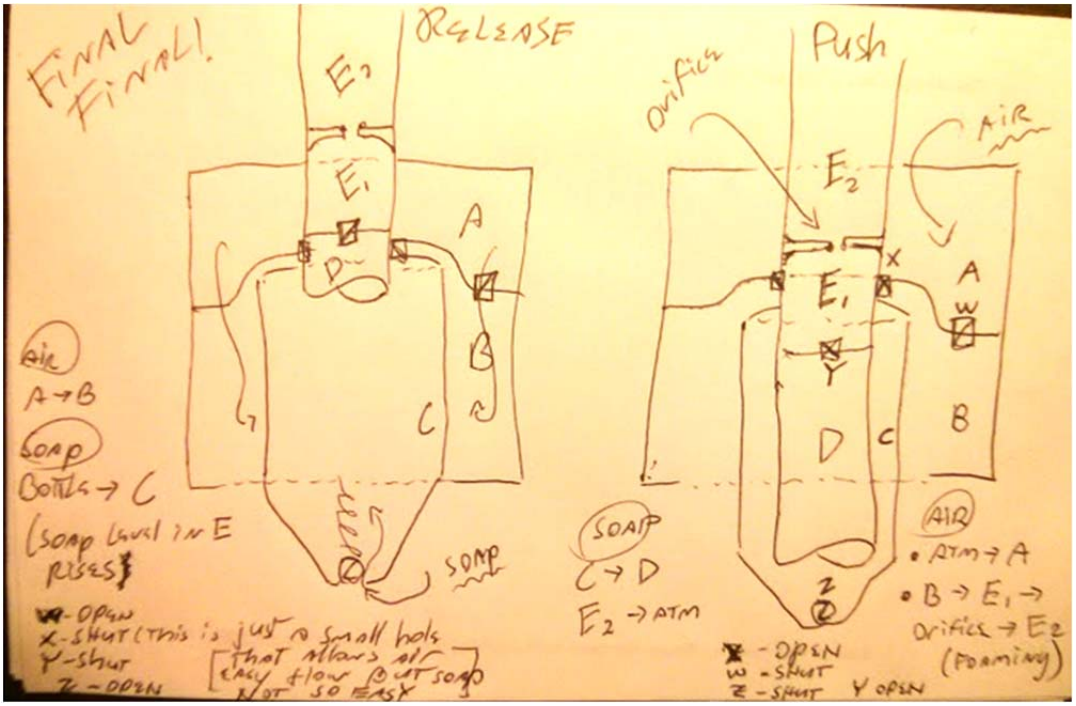

Figure 59. Case study I drawing

Figure taken from the handwritten notes compiled during case study. Shows the operational principle of the foaming pump

1. [Thought/Action] Victory is declared!… jot down some concluding thoughts

[Thought] The process involved many tentative stopping points. But close inspection often revealed the presence of contradictory assumptions. The self-imposed requirement to draw what I understood, forced these contradictions into the open.

[Thought] In spite of initial misgivings, destruction during breach did not prevent my ability to continue to do operational testing (Had an extra bottle just in case, only used it once but it was not really necessary).

[Thought] In spite of initial misgivings, disassembly (as opposed to obtaining a cross-section of the mechanism) was not a bad way to unravel the different functions. As each component was separated from the whole, its intended function was still evident, and more importantly, testable.

[Thought] There is nothing left to learn

B.II CASE STUDY II (MEDIUM-SIZE ROBOT WITH SENSORS) [HEXBUG ORIGINAL. REFER TO FIGURES 42 AND 43]

1. [Thought] First do a pre-operation visual inspection.

2. [Discovery] The shape of the circuit board visible through the carapace seems to be determined by aesthetic purposes. This is unusual because it subordinates the distribution of components to aesthetic considerations.

3. [Discovery] There is a left-right asymmetry in the internal components (gears) visible through the semi-transparent "abdomen."

4. [Discovery] Found On/Off switch and apparent sensor located in rear section.

5. [Thought] Sensor looks like a tiny speaker. It may be a simple microphone for sound detection.

6. [Discovery] Antenna look like possible additional sensors

7. [Discovery] Small circular rubber pads at the end of each leg, probably for improved traction

8. [Action] Move the legs back and forth while looking closely

9. [Discovery] Only the left set of legs permits manual "cranking" through its full range of motion—this is probably tied to the internal asymmetry noted earlier

10. [Action] A drawing: Shows two legs including a cam as the axle of the "power leg" and a linkage between the two legs (Figure 60)

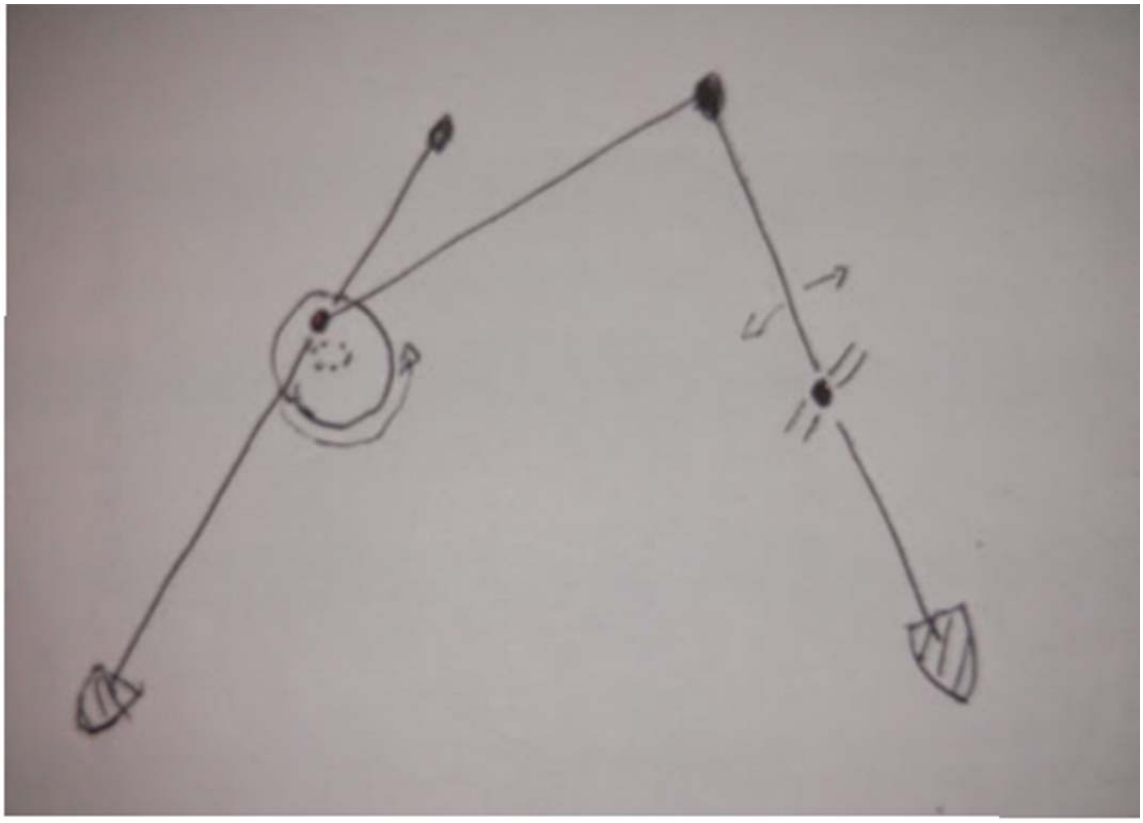

Figure 60. Case study II drawing

Figure taken from the handwritten notes compiled during case study. Shows a partial operational principle of the linked legs (the principle is redundant for missing leg).

1. [Discovery] Soon, the external part of the mechanism for locomotion becomes evident: The center leg (of three) in each side is the only one connected to a prime mover directly. It is connected via a cam (off center) shaft. The cam shaft in results in leg/foot motion of down-and-back or up-and-forward as the shaft rotates through a full cycle—this ensures the center leg comes off the floor when stepping forward. At the same time, linkages transmit force from the center leg to the front and back legs, but the linkages are arranged such that these two legs are out of phase with the center leg (stepping back as the center leg is stepping forward, and vice versa)

2. [Thought] All other external/static features appear to be purely aesthetic

3. [Action] Begin Dynamic inspection—Turn the robot on and watch it walk.

4. [Discovery] Performance is very poor on carpet and soft surfaces (feet slide backward only slightly less than they step forward)

5. [Discovery] Confirmed role of antennae as sensors. The operational principle for the antenna is simple: the base of each antenna is a switch. One terminal the switch is a metal "cylinder" surrounding the base of the antenna, the other terminal is the metal antenna. When the antenna is pushed in any direction (by an obstacle), it is forced out of alignment and into contact with the cylinder. The system then executes an "avoidance routine"

6. [Discovery] Confirmed role of the small component as a sound sensor. In response to a clap or a foot-stomp, the system executes the same "evasive maneuver."

7. [Discovery] The system's avoidance behavior is as follows: Following the trigger "stimulus," the left set of legs stops moving while the right set of legs temporarily reverses its direction, taking approximately 8 steps backward before resuming its normal direction of movement (Note: the behavior actually varies minutely depending on whether the trigger is an obstacle or a loud sound. The obstacle results in about 9 steps backward while the loud sound results in about 8 steps). The end result on a smooth surface is a-turn-while-reversing and an eventual resumption of forward motion in a direction approximately 100 degrees to the right of the original

8. [Thought] There seems to be little left to explain

9. [Action] A drawing: It shows a motor connected to a gearbox via a shaft. The two sensors (antenna and sound sensor) are connected to a box labeled "gear shift actuator." The gear shift actuator provides one of two possible inputs to a gearbox depending on the presence or absence of a signal from a sensor. The gearbox in turn has two possible outputs a) forward on all legs, or b) reverse on left side only.

10. [Thought] The primary remaining questions are: What does the gear shifting actuator actually do? and What arrangement of gears accomplishes the behavior described?

11. [Thought] Nothing further can be learned without opening the system.

12. [Action] Start at most obvious point of entry: Battery Compartment. It is a dead end. After battery is removed the compartment offers no further access options.

13. [Thought] Just remove anything that will come off… maybe then something will become apparent

14. [Action] Only one thing suggests itself: remove all the legs. This is easily accomplished by pulling off their pin-axles

15. [Discovery] It looks like the body of the Hexbug consists of two shells that were glued together.

16. [Action] Close inspection while prying gently with a small screwdriver.

17. [Discovery] The two shells come in contact at 4 points/posts, but only 2 of the 4 have been glued.

18. [Thought] Cutting through the glue may result in only minor destruction and full-function reassembly still possible by using the 2 remaining posts.

19. [Action] Execute above plan. It works nicely.

20. [Discovery] With circuit board exposed, a small screw holding it in place is apparent.

21. [Action] Remove screw on circuit board

22. [Discovery] With the small screw removed, the circuit board is loosened, but it does not separate from the lower body.

23. [Action/Discovery] Continued close inspection reveals the source of the problem: The rigid wire connecting the circuit board to the battery compartment in the lower section is holding the two pieces together.

24. [Action] A small amount of destruction/carving results in the desired release of the rigid wire battery terminal. As soon as the two parts are separated a small gear falls out. The original location is not readily apparent and some time is spent rebuilding the gear set.

25. [Action] Fully assembled, the gears and main axle are available for close inspection and careful manipulation.

26. [Discovery] A small pinion to the right side of the axle couples the motor. A small cylinder couples the left and right sides of the axle.

27. [Thought] This small cylinder may be a clutch

28. [Action] Careful hand-cranking proves that the small cylinder is in fact a clutch. It allows rotation in the forward direction from the motor to be coupled to the left legs while rotation in the reverse direction is decoupled from the left legs.

29. [Thought] The actual operational principle is much simpler design than the proposed actuator and gearbox principle. When the trigger is sent to the circuit board, it simply results in a reversal of the direction of the flow of electricity to the motor. The motor direction of rotation then reverses, causing the left legs to become decoupled, causing the system to execute a turn-while-reversing.

30. [Thought] This theory should be easy to prove. All you need to show is that the motor reverses in response to a trigger.

31. [Action] Do that experiment

32. [Discovery] The tiny size and very high RPM make it impossible to visually determine which way the motor is turning and whether it is reversing.

33. [Thought/Action/Discovery] However,—after some thinking—the reversal of the motor is confirmed by two different methods:

34. Holding the motor between fingers, power it up and listen/feel to the effects of the motor after the response is triggered. You can detect a 4–5 second "hiccup" as the motor reverses, then reverses again.

35. Also the motor can be slowed down using fingertip friction on the pinion, and the direction of rotation (and reversal) can then be directly observed by touch.

36. [Action/Discovery] Trace wiring from the antenna and sensors and to motor. The attempt to obtain a complete trace is not successful given the reverse engineer's limited knowledge of integrated. However, the effort sheds unexpected light on previous observation that had been ignored. The observation (noted as part of event 17) concerned a slight difference between the avoidance routine triggered by the antenna and the one triggered by the sound sensor. The difference was very small (probably just an error in measurement?) and there was no apparent way to explain it so no further effort had been allocated to its explanation. The attempted signal tracing revealed that the analog signals from the antenna and from the sound sensor were each routed to similar but distinct integrated circuit components labeled J476. This suggested that while almost identical, the two avoidance maneuvers were in fact controlled by different components—explaining the small difference.

37. [Action/Thought] The robot is fully reassembled and op-tested. It works. Photographs taken during disassembly prove very useful during re-assembly, especially for the linkages connecting the legs.

38. [Discovery] Some previously unnoticed features are now noticed and explained

39. A small clicking sound (at the same frequency as the rpm) is heard when the robot is backing. This is caused by the slippage of the clutch's single "tooth"

40. The left legs are not actually immobile during backing, but show a small amount of ineffectual motion. This is caused by the slight coupling of the torque across the clutch.

41. [Thought] There is nothing left to learn

B.III CASE STUDY III (TOY GUN) [NERF FIRESTRIKE. REFER TO FIGURES 45 THRU 48]

1. [Action/Thought] External static inspection (this will be followed by dry firing then finally by actual firing)

2. [Discovery/Action] Batteries for sight are not included. Bought and installed required batteries.

3. [Discovery] System comes with 3 darts. The darts are made of a sponge-like material with a higher density rubber tip. The sponge section is cylindrical and hollow.

4. [Discovery] The 3 darts fit in the gun: one in the barrel ready to fire and the other two in a "magazine" (The magazine is only for storage of the darts, transfer from it to the barrel is done by hand).

5. [Action] A drawing. It shows the system's most likely operational principle (a spring-loaded plunger or piston can be charged by pulling back until it engages a sear mechanism or catch. The plunger can then be released by the trigger, pushing against the rear of the dart)

6. [Discovery/Thoughts] Looking inside the barrel reveals some interesting features:

7. The back of the barrel appears to be closed and solid (i.e., there is no apparent plunger or piston-head to propel the dart, and there is also no evident orifice to permit the entrance of air to propel the dart.

8. A small rod extends along the center of the barrel (coaxial with the barrel) from the back almost all the way to the muzzle (front end). It is clear that this rod fits into the hollow part of the loaded dart, but its function is not evident.

9. Three smaller rods can be seen to protrude from the back end of the barrel. These have a blade-like geometry, are shorter than and form a 'y' shape around the central rod. Their function is not evident. (The blade-like profile makes them unlikely to be a plunger, as they would tend to penetrate the soft dart material rather than push it, were they to make high speed contact)

10. [Thought] The inspection of the barrel internals raised more questions than it answered

11. [Action]Test: dry fire (i.e., charging the gun and pulling the trigger without loading ammunition). The gun is charged by pulling on a charging handle connected to a charging rod (similar to the charging of an M4 carbine but unlike the M4 this gun's handle remains locked in the rear position until the weapon is fired).

12. [Discovery/Thoughts] Dry firing the gun results in the release of the externally visible charge mechanism as well as the high speed movement of some heavier internal components—this is apparent from the vibration and sound generated. After a number of dry-fires and close inspections the following is apparent:

13. There is no visible movement inside the barrel during the dry-fire cycle (this is unexpected)

14. There is no discernible flow of air from the barrel during the dry-fire cycle (this is even more unexpected)

15. The internal parts set in motion by the dry firing do not slam at the end of the cycle as might be expected from a spring-powered mechanism, rather they seem to decelerate rapidly but smoothly before reaching the fully discharged position (this, again, is unexpected).

16. [Action] Test: Operate the battery-powered light sight—a second trigger just below the main trigger actuates the sight.

17. [Discovery] The sight picture projected onto a wall is better than expected. The problem solver anticipated the "light beam sight" would be a simple red flashlight. Instead, similar to a laser sight, the system produces a dot that is small and bright. Even in moderate light and at a considerable distance of approximately 25 feet the dot is clearly visible and only about 1" in diameter. The sight picture contains the additional "wow-feature" that it not only projects a laser-like dot, but also a concentric circle around the dot (approximately 3 times the diameter of the dot) suggestive of the sight systems in use with modern firearms (although these do not project the compound picture unto the target but overlay it in the sight picture presented to the shooter)

18. [Action] Test: Actual firing (multiple times at various distances). Once again, the toy gun's performance was far better than the author expected based on his previous experience with similar devices. For example: at 10 feet the dart hits within the small sight picture "painted" on the target, at 25 feet (with no wind) if fired horizontally the dart drops about 10 inches.

19. [Action] A drawing: Explains possible operational principle. The propellant is air from a piston/cylinder mechanism. The drawing shows the theory that the three small rods inside the barrel are part of a spring loaded actuator for a valve. The valve opens when the dart is loaded.

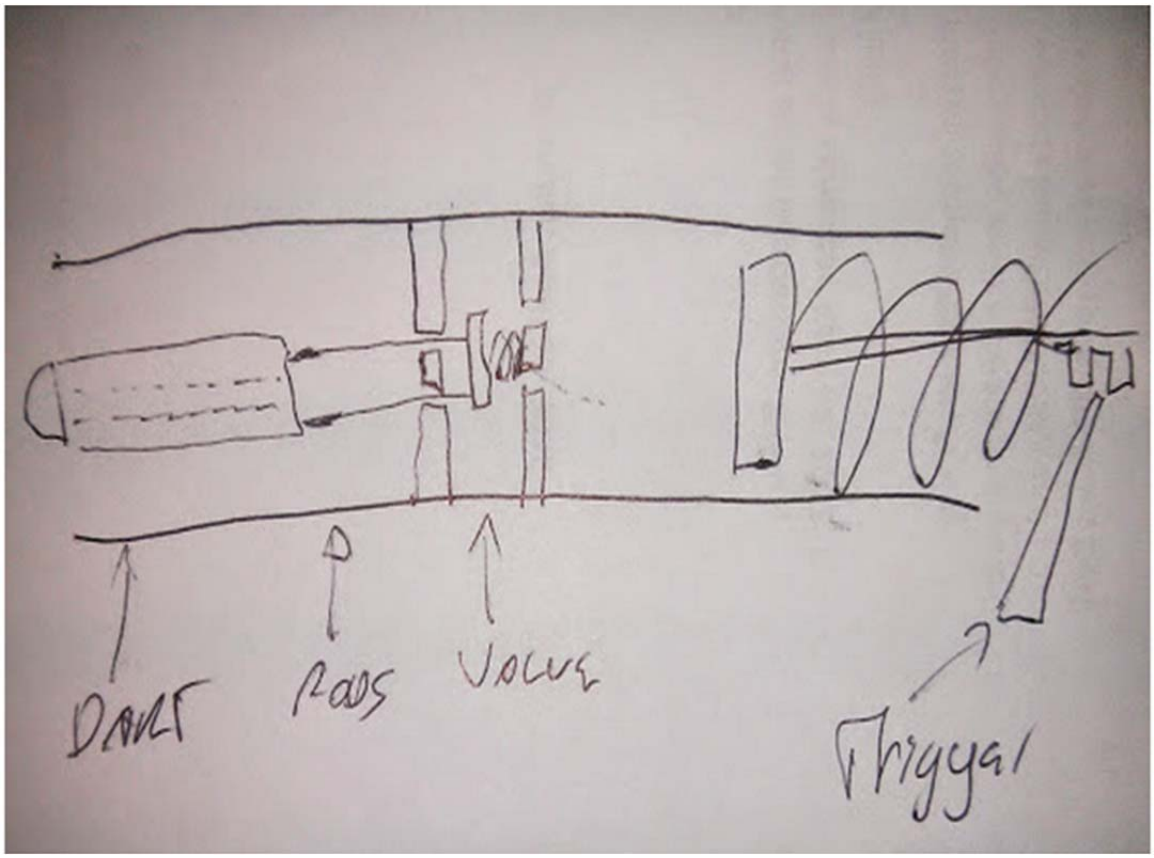

Figure 61. Case study III drawing

Figure taken from the handwritten notes compiled during case study. Shows the operational principle of the firing mechanism. The drawing is based on inference from the observed functions, the system had not been breached yet

20. [Thought] The drawing prompts further reasoning: When the dart is loaded, the valve is pushed open by the back part of the dart. The short length of the actuator means that once the dart flies out (even before it leaves the barrel) the valve is no longer held open. The valve then shuts and the air still in the cylinder causes the observed smooth and quick deceleration of the piston.

21. [Thought] Another interesting question is presented by the concentric dot and circle sight picture. How is this accomplished?

22. [Thought] Nothing further can be learned without opening the system.

23. [Action] Begin disassembly at obvious point, by attempting to remove seven small Phillips screws evenly spaced around the left side of the receiver (although this is a toy, in standard firearms terminology a receiver is like the chassis of a gun—the part onto which all other parts are attached)

24. [Discovery] The deeply recessed screws and relatively small heads result in a failed first attempt (and a near-stripping of one of the heads). Rather than pushing forward, the author decided to pause and go acquire the correct tool.

25. [Thought] There are probably several springs. Some thought suggest at least three: The main spring (for the shooting mechanism), a small trigger spring (for returning the trigger to the forward position after it is pulled), and a similar one for the light actuator (it works the same way as a trigger). The reason for pausing to think about springs is to help anticipate the possibility of one or more of them flying out as the receiver is opened.

26. [Action] Left side of the receiver lifted/removed

27. [Action] Barrel and piston with several attached parts are easily lifted away from the right side of the receiver for further disassembly. Sight and trigger are left in receiver for later disassembly.

28. [Action] Continued disassembly of barrel/piston. Two components attached to the barrel can be removed by lifting tab/pulling. Both components seem to have no function other than aesthetic (perhaps they support the barrel, but this seems unnecessary given the light weight).

29. [Action] Disassembly continues—pulling them apart can easily separate Barrel, piston, and charging handle.

30. [Thought] The gun is almost fully disassembled and yet the exact firing mechanism remains unclear.

31. [Thought/Action] The assembly/joining method for most components is repeated throughout the design: adjacent parts are pressure fit onto each other, additionally held in place and aligned by a tab-and-groove (A drawing made to supplement this statement)

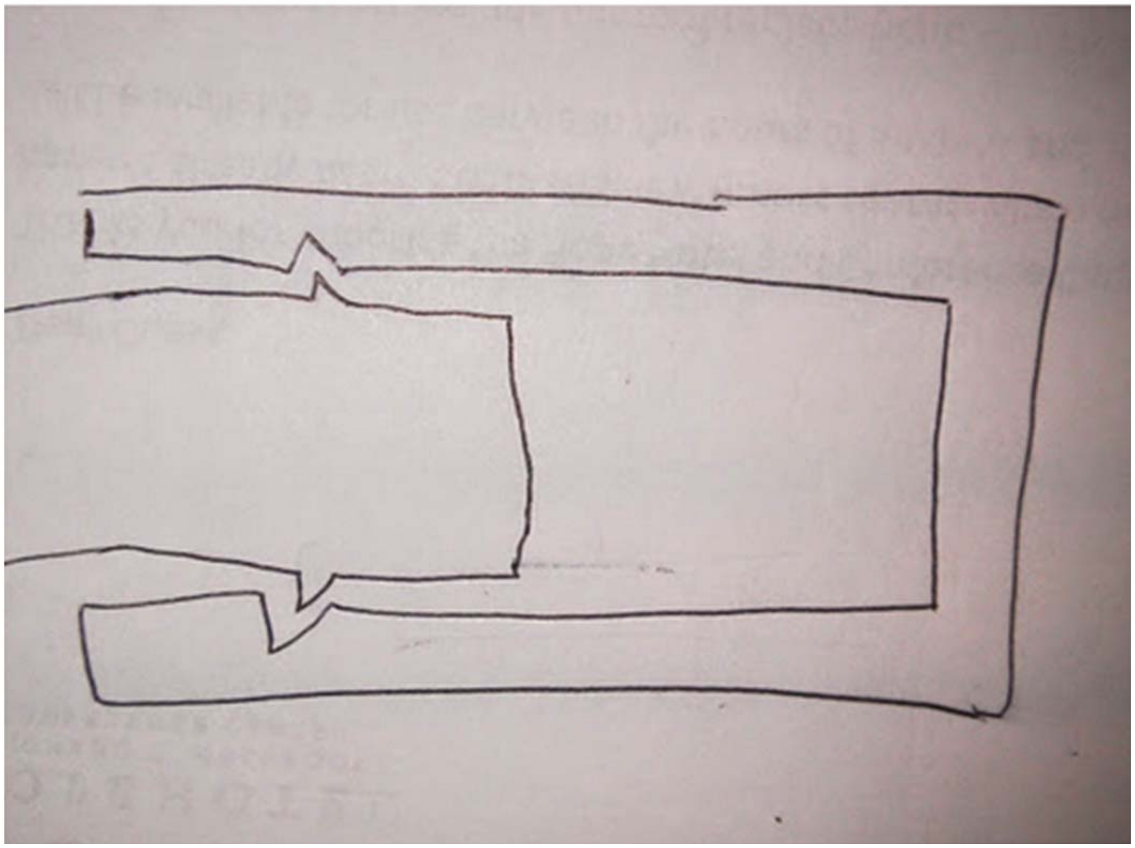

Figure 62. Case study III drawing

Second Figure taken from the handwritten notes compiled during case study. Shows the means for assembly of adjoining parts. This same means was used repeatedly throughout the system. The drawing was added because the reverse engineer lacked the technical terminology to describe it.

1. [Action/Discovery] The rear end of the barrel is removed revealing it to be a valve as earlier speculated. The three rods are in fact an actuator that opens the valve when the dart is in place. When the valve is open air can flow through and around the three rods

2. [Thought] With this function cleared up, it is still not clear why this valve has been included in the design. I.E. The what is now known, but the why remains uncertain. Two characteristics of the valve behavior suggest two possible purposes:

3. It causes the piston to decelerate smoothly rather than slam against the end of the cylinder (Possible purpose based on this: Improved system life expectancy, operator comfort, or ease of firing… perhaps the smooth action results in improved accuracy?)

4. It prevents the weapon from discharging a high-speed burst of air during a dry fire (Possible purpose based on this: Safety. It prevents the accidental discharge of a high speed burst of air directly into the eye of a person—perhaps a person who happens to be engaged in an effort to understand how the gun works.)

5. There is nothing left to learn

B.IV CASE STUDY IV (SMALL-SIZE TOY ROBOT) [HEXBUG NANO. REFER TO FIGURES 50 THRU 52]

1. [Thought] The price (less than $6.00) and size (less than 2 inches) of this system suggest very simple behavior.

2. [Action] External Inspection

3. [Discovery] System contains 12 soft "legs" connected to a single piece of rubber that partially encases a hard plastic body. As the piece of rubber surrounds the body where it may come into contact with obstacles it will henceforth be referred to as the "bumper." The legs and the bumper are in fact a single molded homogeneous part.

4. [Discovery] The 12 legs are identical as follows: they extend vertically from the body; they have a slight curvature or rake to the rear terminating at an angle with the floor of approx. 20 degrees; they have a slight taper and are rounded at the end so they terminate in a "point"; they are evenly spaced; they occupy the rear two thirds of the body so that the front of the body (the head) has no legs directly under it.

5. [Thought] On the appearance: The dimensions and external geometry of the system incorporate a number of details whose role appears to be aesthetic, but nuanced as follows: The system is meant to look like a cockroach AND like a robot. Given its size, it would be very easy to design this system to look like a real cockroach, this would almost certainly reduce the system's appeal (if not with the users, certainly with the user's mothers).

6. [Action] A drawing: A possible operational principle. Shows a back-and-forth motion is somehow transmitted onto the bumper/legs. Given the angle of contact with the floor each leg functions like a barb, the feet are allowed to slide forward on the floor but are prevented from sliding backward, resulting in a net-forward motion.

7. [Discovery] The lower-half geometry of the plastic body includes a bulge in the back that is suggestive of an abdomen. However, the shape is more likely driven by the fact that it houses—and conforms to the dimensions of—the battery (the plastic body is semi-transparent). NOTE: this also

contains the only evident point of entry: a single screw for access to the battery compartment.

8. [Action] Op testing consists of merely turning the system on and "setting it loose." The overall behavior of the system is surprising for several reasons.

9. [Discovery] The mode of propulsion is clearly not the back and forth motion speculated earlier—there is no back and forth movement at all. In fact, there is no movement of any external part relative to another

10. [Discovery] Intense vibration obviously plays an important role in the operational principle behind the locomotion—even before it is placed on the floor, the strength and high frequency of the system's vibration is surprising.

11. [Discovery] As soon as it is placed down, the system moves around much faster than expected

12. [Discovery] The overall characteristic of locomotion is difficult to convey without saying things like "aggressive curiosity." For example if placed in a cluttered drawer, the Nano quickly navigates the available surface. If it encounters a small/light obstacle, it simply pushes through. If it encounters a large obstacle it "selects" a random new direction without slowing down. It is also able to back out of narrow gully-type obstacles.

13. [Discovery] It seems almost unable to get stuck by any of the obstacles with one exception: As it rides onto a sheet of paper that is lying flat on the bottom of the drawer, the system abruptly comes to an almost complete stop. Also, the locomotion mechanism does not work at all in soft or uneven surfaces (carpet, blanket, dirt)

14. [Discovery] If placed on its side or its back, the system vibrates more violently causing it to bounce around until it quickly lands on its feet then resumes its normal movements

15. [Thought] The changes in direction appear to be random thus giving the impression of life and volition. Could such a locomotion mechanism be steered or is it inherently random?

16. [Thought/Discovery] Now that the system has been observed in action, a number of features that went unnoticed during the static inspection become evident through their apparent role (although the features were not small nor hidden):

17. The lower geometry of the plastic body appears to be slightly boat-like. This cause the system to partially ride up small obstacles before "reversing" and changing course

18. The upper geometry of the plastic body includes a roughly pyramid-shaped humped back. This causes the system to be inherently unstable when it is upside down… the vibration quickly returns it to its feet-down orientation.

19. The overall geometry and weight distribution results in a very low center of mass (below the point where the legs join the body): This makes the Nano very stable in spite of its high speed and narrow body.

20. [Thought] At this point, nothing more can be learned without opening up the system

21. [Action/Discovery] Remove battery cover and battery and… there is no clear next step. The body and the bumper are all glued together and the internals appear inaccessible without major destruction.

22. [Thought] The prospect of destroying the system causes some hesitation: I want to see if I can better understand not just the mechanism, but also the capabilities of this mode of propulsion (which is completely unlike any mode of propulsion I have previously encountered). However, it is evident that further breaching of the system will almost certainly result in its destruction: any testing should be done before the breach is attempted. Changed of mind: will do some further testing before proceeding with system breach

23. [Action] Time to do some more creative testing

24. [Discovery] While replacing the battery and cover, one of the legs is accidentally pinched by the cover. Left thus and turned on, the missing leg causes no noticeable degradation in performance. It also suggests a direction for further tests.

25. [Action/Discovery] With the aid of some tape, numerous configurations of "degraded leg inventory" are experimented with. The end result of the testing is that the locomotion mechanism is very robust. For example, the system may lose over 80% (10 out of 12) of its legs and still function with only moderate degradation of performance: the vibration is more violent and the system becomes somewhat less stable, but it still "explores" all traversable surfaces relatively quickly. The system could lose up to 60% of its legs and show almost no noticeable degradation (as long as the original two front legs remain)

26. [Discovery] One exception to the robustness: if the remaining front-most legs are not the same number, then the locomotion path becomes a tight circle. For example, if only one front leg (say, the left one) is disabled, then the remaining front-most left leg is #2 and the remaining front-most right leg is still #1: not the same number.

27. [Action] A (revised) drawing. Explaining the operational principle (Figure 63)

28. [Thought/Discovery] In light of the new drawing, the system's abrupt reduction in performance when it "rides up" on a piece of paper can be understood as a decoupling of the interaction between the legs and the floor

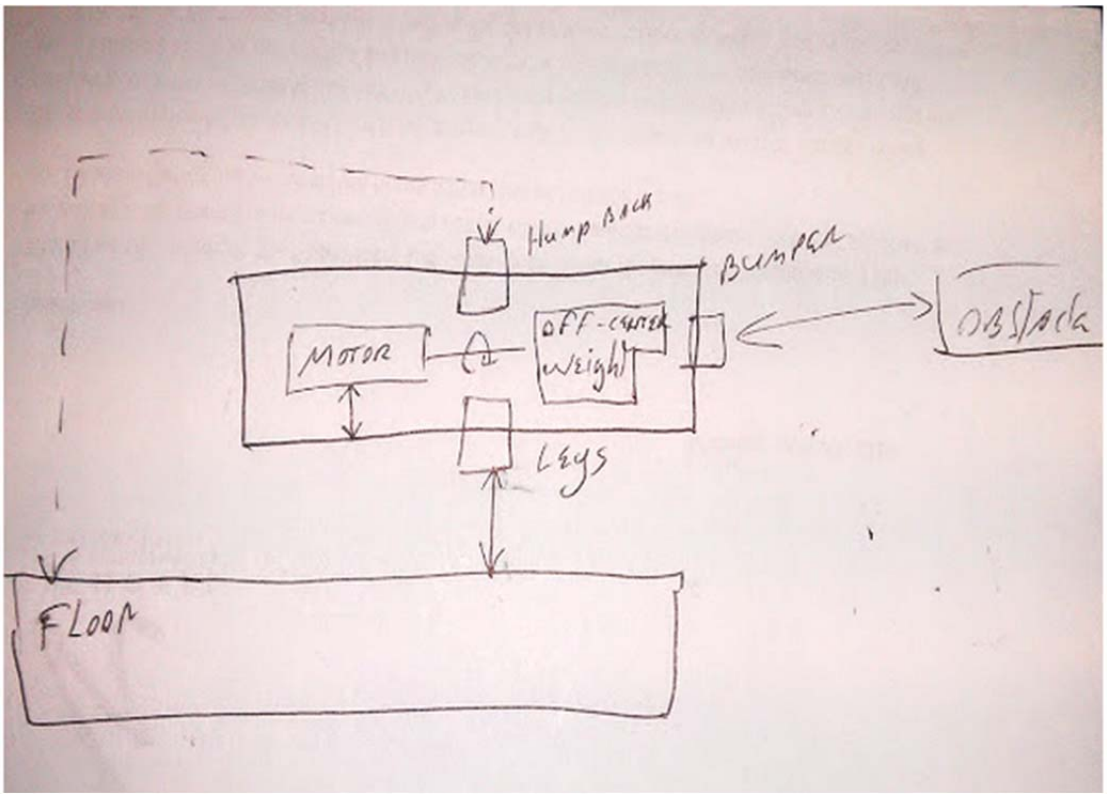

Figure 63. Case study IV drawing

Figure taken from the handwritten notes compiled during case study. Shows the operational principle of the small toy robot. Note that all the information necessary to make this diagram (which is accurate) was obtainable without recourse to breaching the system.

29. [Thought] However, some further consideration of the drawing "reveals" that I still do not understand the locomotion operational principle—The problem is abbreviated as:

30. OP: Legs <—>Floor = ??

31. [Thought] Further testing is required!

32. [Action] Test: Dropping the system from a small height (2-3 inches) results in a slight bounce with a horizontal component (with the power

turned off). The horizontal component of the bounce is usually forward approximately ½ inch, sometimes it is sideways, it is never backward.

33. [Thought] Have a new theory for OP: Legs <—> Floor as follows: The vibration (induced by the rotation of an off-center rotating weight) caused the system to be constantly displaced vertically (the situation may be described as a constant state of micro jumping). Whenever the system lands from one of this jumps, the leg geometry cause a portion of the downward momentum to be transformed into forward momentum

34. [Action] A drawing: Shows a "time series" for the geometry and movement of a single leg in an attempt to visualize the explanation just offered (Figure 64)

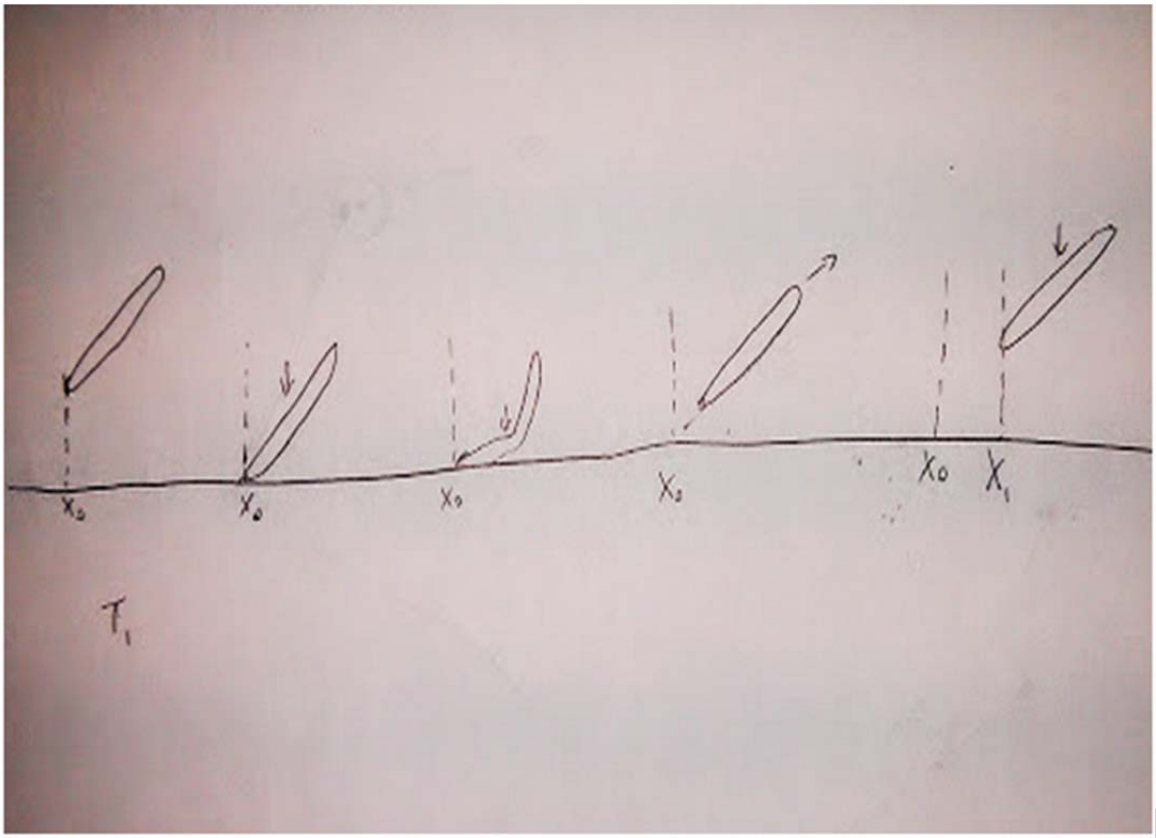

Figure 64. Case study IV drawing

Second Figure taken from the handwritten notes compiled during case study. Shows the operational principle believed to be responsible for the small robot's locomotion. This is based on experiments described in the transcript.

35. [Discovery] The semi-transparent plastic body permits the quick discovery of the theorized off-center weight without opening anything

36. [Thought] At this point, the reverse engineering goal of understanding the operational principle seems to have been fully satisfied without ever opening the system (it helps that it is semi-transparent)

37. [Thought] Decide to proceed with breaching to see if this uncovers any challenges/solutions with a process that is likely to result in serious damage to the system (this is almost sad, given the lively performance)

38. [Action] As mentioned earlier, removal of the battery is a dead-end toward breaching

39. [Action] The only action that suggests itself is to remove the legs/bumper by pulling on it. This is managed with only minor damage to the soft part.

40. [Discovery] Removal of the bumper exposes a seam joining the top and bottom shells of the plastic body. We have a point of entry!

41. [Action] After only a few minutes of working carefully along the seam with a sharp knife a clean separation of the two shells is obtained. The System can still be rebuilt (with the help of a little glue)

42. [Thought] As noted earlier in regard to the battery compartment/abdomen: the external dimensions are largely determined by the need to provide the smallest possible cover around the internal components (i.e., there is no spare room inside)

43. [Thought] There is nothing left to learn

B.V CASE STUDY V (THE ANTIKYTHERA MECHANISM) [REFER TO FIGURES 50 THRU 52][56]

1900–1902 (The find)

1. [Discovery] A group of sponge divers stumble onto a number of bronze statues and other submerged artifacts near the island of Antikythera, Greece. (1900)

2. [Action] Along with the statues, a corroded lump of bronze (about the size of a large dictionary) is extracted from the sea along with the other artifacts. It lies unnoticed in the Athens museum for two years.

3. [Discovery] Spyridon Stais (politician, former schoolteacher) visits museum and notices the lump, which has split open. He observes three fragments containing, clearly containing gears and a few inscriptions on the surface.

[56] The timeline presented is a synthesis of (Freeth 2008), (Jones 2012), (Price 1959), and (Marchant 2012)

1902-1910 (Many archeologists and other scholars)

4. [Thought] Valerio Stais (Archeologist) Publishes a paper suggesting the mechanism is some kind of clock.

5. [Thought] Stais is ridiculed by his peers (They contend that timekeeping of the era in question was accomplished by sundial. The Greeks were known to use gears exclusively in simple devices (1902)

6. [Action] Othon Rousopoulos (chemist) undertakes cleaning and preservation. Originally there were three fragments of mechanism. In the process of restoration additional small fragments become separated from the larger ones (intentionally or unintentionally?). Eventually there are approximately 20 fragments. (1904-1905)

7. [Action, Thought] Several archeologists, a historian, an epigraphist, and a naval officer inspect the mechanism between 1902 and 1910... Public disputes—recorded in a local newspaper—ensue as to the function and purpose.

8. [Thought] All involved agreed on one thing: the mechanism must have a maritime purpose (since it was found in a shipwreck). As to the function, two competing theories emerge:

9. [Thought] It is a navigation instrument, possibly related to a compass. This theory is based on the vocabulary of the inscriptions but makes minimal effort to account for the gears

10. [Thought] It is a marine odometer (for tracking the revolutions of a paddle wheel). This theory is inspired by the gears but disregards the astronomical references in the inscriptions

11. [Thought] Albert Rehm (philologist) publishes a paper claiming the mechanism has an astronomical-calculation function. He is the first to suggest (still the accepted view) that this was not a marine instrument but rather an artifact found on the wreck incidentally, as part of the ship's cargo.

12. [Action] Parallel effort: During the same time frame, other archeologists date the wreck based on the surrounding items in the first century BC (the mechanism itself is of no use for archeological dating, as nothing like it has ever been discovered)

Late 1920s-early 1930s (Ioannis Theofanidis)

13. [Action] Theofanidis (naval officer) Attempts to build a model of the device.

14. [Action] Publishes a paper concluding the mechanism is a navigational instrument related to an astrolabe (merging of Rehm's idea and the navigational instrument theory)

1951-1974 (Derek de Solla Price, George Stamires, and Charalampos Karakalos)

15. [Thought] Derek de Solla Price (Physicist and Mathematician) hears about and becomes interested in the artifact (1951)

16. [Action] Price visits the museum several times, enlists assistance of Stamires (Epigrapher)

17. [Action] Price publishes "An Ancient Greek Computer" (1959). He makes a series of detailed observations for the first time. Among these are the following

18. [Discovery/Thought] "The general pattern of the mechanism is quite clear... an input provided by an axle… leading through an epicyclic turntable and coming eventually to a set of shafts that turned the dial pointers. When the input axle was turned, the pointers all moved at various speeds around their dials."

19. [Discovery/Thought] He suggests all the gears are cut from a single sheet of bronze

20. [Action/Discovery] Measurement of gear teeth yields that they are the same size and shape (triangular with a 60 degree angle) throughout all the gears.

21. [Action/Discovery] Microscope measurement of the inscribed graduations in one of the dials finds an error of 0.25 degrees in 45 degrees.

22. [Discovery/Thought] Finds "signs that the machine was repaired at least twice... indicates that the machine actually worked" (One of these discoveries is now believed to be not a repair but a finely crafted feature critical to one of the functions—See 2000-Present)

23. [Thought] He believes he can say exactly what the front dial did: it was used to track the annual motion of the sun in the zodiac

24. [Find/Thought] The back dial, both more complex and deteriorated, presents a greater challenge. Inscriptions suggest tracking of lunar phenomena (phases, times of rising and setting)

25. [Find/Thought] Based on the apparent setting of several adjustable pieces, concludes (with reservations) that the instrument was made around 82 B.C., used for a short time, then taken onto the ship within 30 years of its construction.

26. [Find] “The main inscriptions are in a sorry state and only short snatches of them can be read.”

27. [Thought] Nevertheless, the presence of a few complete words gives several strong ideas as to function. For example, one line reads “76 years, 19 years.” Price concludes this must be a reference to the Callippic and Metonic cycles. Another inscription reads “223,” and price concludes that it must be a reference to the eclipse cycle of 223 lunar months known as the Saros cycle.

28. [Thought] “The Antikythera mechanism must therefore be an arithmetical counterpart of the much more familiar geometrical models of the solar system”

29. [Thought] We have no way of knowing whether the device was turned automatically or by hand. It might well have been turned by the power from a water clock or some other device.

30. [Thought] “What is it? There are four ways of getting at the answer First, if we knew the details of the mechanism, we should know what it did. Second, if we could read the dials, we could tell what they showed. Third, if we could understand the inscriptions, they might tell us about the mechanism. Fourth, if we knew of any similar mechanism, analogies might be helpful. All these approaches must be used, for none of them is complete.”

31. [Action] 1971 Charalampos Karakalos (Physicist and Radiographer) joins Price and helps with x-rays

32. [Action/Discovery] Although it revealed vast amounts of previously un-guessed at information, Karakolos’ X-rays were 2-dimensional. All the newly discovered gears are shown as overlapping, their relative positions are still a puzzle.

33. [Action] 1974 Price publishes “Gears from the Greeks” The salient points of his work follow:

34. [Discovery] Approximately 30 previously hidden gears are revealed by the X-rays

35. [Discovery/thought] A particular gear arrangement is interpreted as a differential gear mechanism. He believes such a mechanism has a function in predicting the phases of the moon.

36. [Thought] Regarding the differential gears, he remarks that the “next” known instance of such a mechanism occurs 18 centuries later. Therefore, its presence necessarily points to an OEM even more technologically advanced than previously hinted at. (It is now generally believed that

Price's conclusion that the mechanism used differential gears was incorrect)

37. [Thought] The book delves into the scientific/historical significance of his discovery (i.e., the shortcomings of our understanding of ancient science and engineering)

38. [Thought] The book presents a new hypothesis as to the origin and ancestry of the mechanism

39. [Thought] Through this book Price believes he has authored a lasting and significant revision to the accepted history of technology. According to the accepted view, an unprecedented explosion of technology takes place in the middle ages and renaissance. In Price's version of history, there is an unbroken line connecting modern machinery to the Antikythera mechanism. The supposedly new technologies had been well known for centuries before the middle ages, and it is only our incomplete record of history that led us to believe otherwise. (This view of unbroken technological lineage did not become the new accepted view)

40. [Action] Karakolos and his wife count the teeth on gears that seem to form an continuous gear train. They count 65, 38, 48, 24, 128, and 32 teeth. The resulting ratio between the input and the output of the gear train is 260 to 19.

41. [Thought] Price notices the closeness between this ratio and the Metonic cycle. The sun and the moon return to the same exact position relative to each other and the earth every 19 years (or 254 sidereal months).

42. [Action/Discovery] Price decides that changing 65 to 64 and 128 to 127 falls within what he feels to be an acceptable error margin. The tweak yields the exact ratio of the Metonic cycle. Modern measurements confirm Price's guess. Furthermore, the discovery of the Metonic cycle gear train is now considered one of the keys to unlocking the mystery of the Antikythera.

43. [Action] Price builds a model of the mechanism (with several gears added that Price had not seen, but only inferred from his mental model of the mechanism's purpose).

44. [Thought] Price believes the Metonic gear train is used to track the position of the Moon relative to the stars (i.e., the zodiac).

1990s (Michael Wright, Allan Bromley and Frank Percival)

45. [Thought] Wright (Curator of Mechanical Engineering at Science Museum in London) is impressed by Price's work. However, some aspects of it do not make sense to him. For example, why use a complicated differential

gear, when a simpler arrangement could accomplish the same function job?

46. [Thought] Bromley (Historian of computing, considered a world authority on early computers, including the Difference Engine designed but never built by Charles Babbage) is also drawn in and troubled by Price's book. He is motivated to undertake his own research ."..by disquiet felt over some aspects of Price's reconstruction. In particular... the high step-up ratio of nearly 25:l in the gearing from Price's main drive… the absence of any indication for the day—the most obvious of all astronomical phenomena; and the uncertainty about the purpose of [certain other parts]"

47. [Action] Bromley builds a model (using a Meccano construction set) to test some of Price's theories, and confirms that Price's model cannot work.

48. [Action] Bromley teams up with Percival (a clockmaker) and together they endeavor to build a working model.

49. [Thought] Bromley and Wright share the conviction that ."..Price really was mistaken in important respects"

50. [Action] The two decide to combine efforts on a "wholly independent survey of every detail, amassing data by direct examination and measurement." They build an x-ray machine that allows them to use a new technique to achieve improved resolution.

51. [Thought] Wright believes that at this new resolution "the definition of the image appeared to be limited not by the imaging technique but by the ruined state of the artifact." In truth, the technique was still very limited in what it could reveal, compared to other techniques used later. However, the technique did provide a sufficient improvement over Karakolos' X-rays to prove conclusively that Price's model had been flawed.

52. [Action] Wright and Bromley jointly publish a paper describing the effort, the new data (thousands of pages of imagery), and some preliminary finds. Shortly after this, they part ways. Bromley takes the data and refuses to share it with Wright. Wright finally gets access to the data after Bromley dies of cancer a few years later.

53. [Discovery] The Moon phase device was a feature that had been seen since the early work of Svoronos in 1902 but not understood for more than 100 years until Wright suggested its function.

54. [Thought] Wright resurrects some ideas that Price had considered but discarded. According to Wright, price had a habit of discarding some of his best ideas.

55. [Action] Wright builds a model with 40 additional gears. The model tracks the location of 5 planets in addition to those of the sun and moon.

56. [Thought] His peers regard Wright's model as very ingenious, but the significant increase in complexity over and above what has been found in the mechanism raised doubts as to its accuracy

57. Wright continues to work on the project

2000's–Present (Antikythera Mechanism Research Project, Mike Edmunds, Tony Freeth, and others including historians, astronomers, and imaging technology experts)

58. [Action] Edmunds (Astronomer, astrophysicist) approaches Freeth (filmmaker and mathematician) about the possibility of making a documentary film about the mechanism

59. [Action] They gather an international team of scholars and two teams of imaging technology experts

60. [Discovery] Unexpectedly, an archeologist from the museum in Athens contacts the newly formed group to inform them she has found additional Antikythera fragments. The count jumps from around 27 to 82 fragments

61. [Action] A team from Hewlett-Packard explores the surface of the fragments using a new photographic computer-aided technique called Polynomial Texture Mapping (PTM), which enables the bringing out of surface details that are invisible to the eye. A number of inscriptions are discovered and/or clarified.

62. [Action] A team from X-Tek Systems brings an 8 ton machine from the UK to perform Microfocus X-ray Computed Tomography (CT). The machine, which was designed to detect subsurface defects on turbine blades (and is thus known to its users as the Blade Runner) is specifically modified for work on the Antikythera. The machine is used to X-ray a sample while it rotates, the result (after computer aided manipulation) is a high resolution 3-D imagery.

63. [Discovery] The improved technology and combined imaging efforts more than triple the number of text characters identified (now around 3000)

64. [Discovery] Some of the inscriptions can be used confirm hypotheses that have been around for decades. For example, one inscription reads "spiral with 235 segments" about which Freeth writes "I nearly fell off my chair with surprise! Here we had in one short phrase confirmation of both the Metonic Calendar and Michael Wright's proposal of spiral dials."

65. [Action] Freeth and his team develop a computer model of the mechanism based on the combined information (which includes new physical details about the gears as well as many more clues from the inscriptions).

66. [Thought] Freeth claims that all gears have now been accounted for in his model, with the exception of one small gear!

67. [Action] Freeth releases the documentary "The World's First Computer"

68. [Thought] Freeth publishes two articles in Nature. In one of them he offers the following summary:

69. Prices model presented a mechanism that was physically complex and clever, but performed a relatively simple function

70. Wright and Bromley's model was an attempt to correct this situation by reducing the complexity of the mechanism

71. Freeth and his team provide a new correction in the form of a model that is physically complex, and performs highly complex functions

72. [Action] The Antikythera Mechanism Research Project (AMRP) is established as an international collaboration under the auspices of the Hellenic Ministry of Culture and supported by international grants. The group continues to do research on the data gathered, but also looks forward to the possibility of uncovering new pieces of the puzzle from the ocean floor.

73. [Action] The most recent archeological survey of the site was carried out in October of 2014. No information has been published to indicate that anything relevant to the mechanism was unearthed.

APPENDIX C. FIVE VIRTUAL CASE STUDIES

As described in Appendix A, the execution of the reverse engineering model in Monterey Phoenix resulted in 368 event traces. Inspection of the 368 event traces reveals that they can be grouped into five families. In the following section, an event trace from each family is described in detail in order to provide five virtual case studies that can supplement the five real world case studies discussed in Appendix B.

C.I VIRTUAL CASE STUDY I—EVENT TRACE # 20 OF 368.

This case study falls under Family # 1 as described in Chapter VII: Failure due to incomplete information. The trace (Figure 65) shows a scenario in which the reverse engineer discovered a function in the target system. A real world version of such an event could be the reverse engineer's discovery of a function like *it detects metal objects at a distance of 6 inches or less* (in the case of a sensor target system) or *it avoids obstacles before coming into contact with them* (in the case an autonomous roving target system). The virtual reverse engineer then proceeded directly from the discovery of the function to an attempt to allocate it. We might imagine that a movement to allocation while in possession of only incomplete information could be due to the operation of a time constraint. At this point, a reverse engineer in the real world might speculate on a hidden interface, and at least temporarily allocate the function to it. However, in the MP model the attempted allocation must fail because no corresponding interface has been discovered. This is a simplifying assumption. One may imagine a reverse engineer thinking something like "I can see that it detects metal objects… but I cannot see any exterior part that may be the detector." At this point, a real reverse engineer might give up, or he might simply go on with the exploration, either basing his working model on a speculated interface hidden below the surface, or withholding judgment and trusting that an interface will eventually turn up as the process moves on to the target system breach and beyond. The virtual reverse engineer, however, has been deprived of the ability to hope, or speculate about things he does not directly experience. Thus, following the failed allocation the virtual reverse engineer makes a doomed attempt to formulate a working

model. One might imagine him thinking something like “I can see it detects metal objects, but I ca not see any exterior part that may be the detector… therefore try as I may, I cannot come up with any theory as to how it might work.”

The modeled reverse engineer’s apparent “lack of imagination” is an artifact of the simplifying assumption that the target system has a single-level of structure/function. Thus, each scenario unfolds in the static (from the point of view of system integrity) portion of the reverse engineering process punctuated by breach at either end. A more complex (and realistic) MP model could incorporate one or more breaches. In such a model, it would be important (and possible) to describe a reverse engineer whose behavior includes actions based on speculation about what may be found beyond the system breach, and the ability to proceed to system breach and the next layer of exploration and testing even if the exploration and testing of the current layer has yielded no discoveries.

This virtual case study validates the model of reverse engineering and specifically mode of failure #7—”Incomplete characterization of system function due to hidden object or attribute” and mode of failure #10—”Incomplete allocation of system functions (due to hidden object or attribute).” Additionally, as presented in Chapter VI there was no obvious connection between the two modes of failure. However, in the process of formally specifying the reverse engineering process in MP, it became explicit that once mode of failure #7 occurs, it must lead to mode of failure #10 (although mode of failure #10 can also exist on its own).

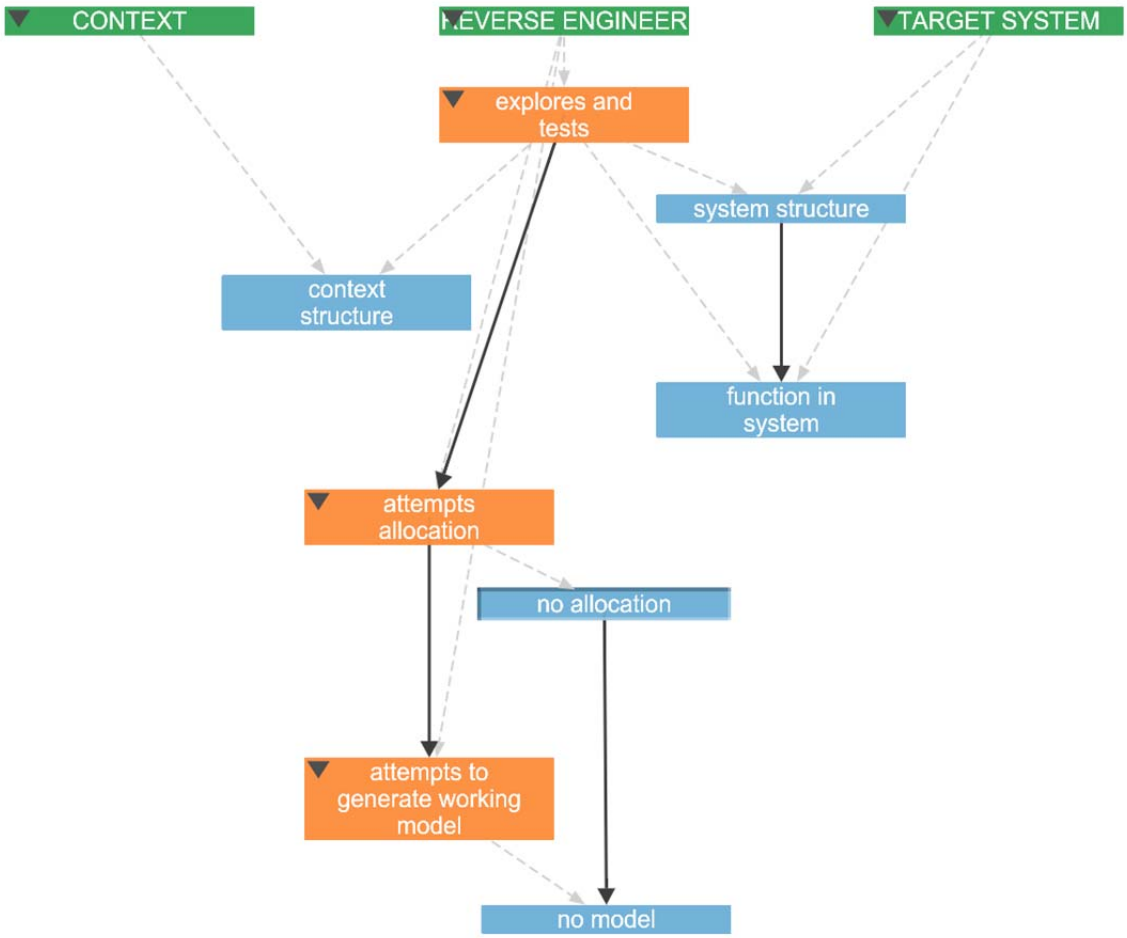

Figure 65. Event trace #20 of 368—Family 1

1. [Action] Reverse Engineer studies the structure of the target system
2. [Discovery] Reverse Engineer discovers a function in the structure of target system
3. [Action] Reverse Engineer studies the structure of the context (nothing found)
4. [Thought] Reverse Engineer attempts to allocate the discovered function to an interface (but none have been discovered, so allocation does not happen)
5. [Thought] Reverse Engineer attempts to formulate a working model (but there is no allocation about which a working model can be made)
6. [Action] Reverse Engineer stops

C.II VIRTUAL CASE STUDY II—EVENT TRACE #45 OF 368

This case study falls under Family # 2 as described in Chapter VII: Failure due to false information. The trace (Figure 66) shows a scenario in which the reverse engineer discovered a true interface in the target system, then discovered a false function. He then allocated the true interface to the false function. The false allocation necessarily led to a false working model. A real world version of such a scenario could have a reverse engineer exploring a metal detector. The exploration soon exposes a conspicuous component. The reverse engineer identifies this component (correctly) as the interface responsible for the detection of metals. The reverse engineer also determines (incorrectly) that the metal detection is accomplished by means of ultrasonic sonar. He therefore performs a false allocation declaring "this conspicuous component (true interface) is an ultrasonic transducer (false function)." Because the allocation is false, any subsequent working model derived from this allocation is also necessarily false. Although this is a simplifying assumption, it is easy to see that it must almost certainly be true. For example, a description of *how Interface X performs Function F* can hardly be accurate if Interface X does not actually perform Function F.

This virtual case study validates the model of reverse engineering and specifically modes of failure #4 and #5 (From Table 3)—"Inaccurate Characterization of System Function." Note that the two modes of failure presented in Chapter VI propose different causes for the inaccurate characterization, while the MP model of reverse engineering does not recognize causes. In the MP model, a function can be: characterized, inaccurately characterized, or incompletely characterized.

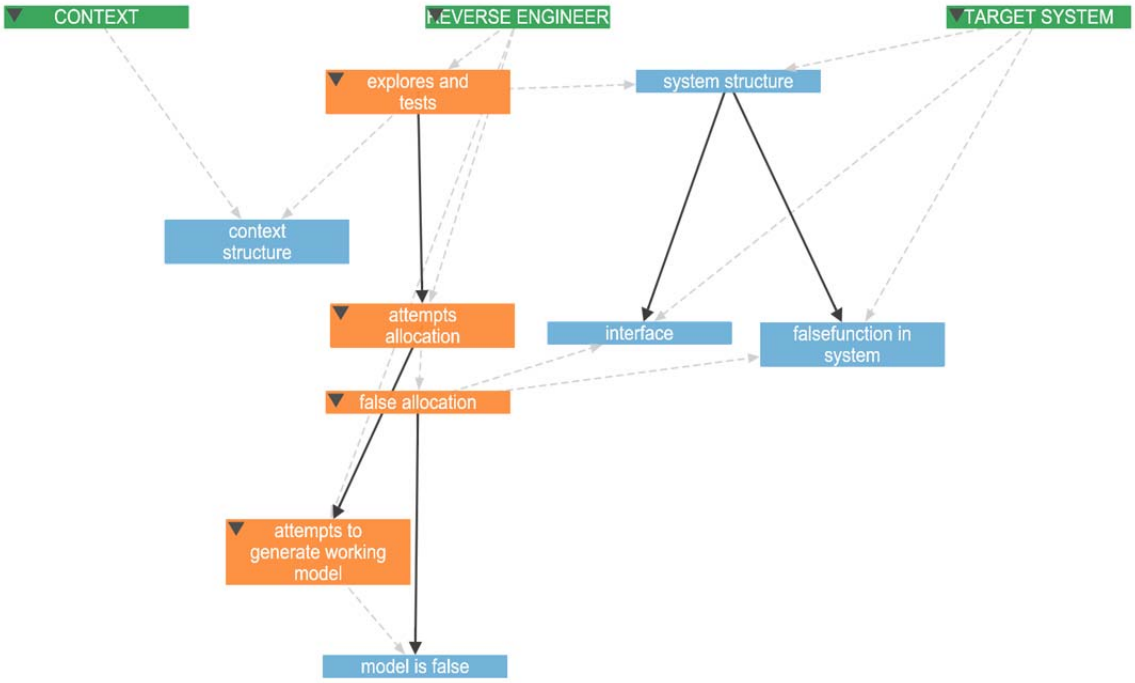

Figure 66. Event trace #45 of 368—Family 2

1. [Action] Reverse Engineer studies the structure of the target system
2. [Discovery] Reverse Engineer discovers an interface in the structure of target system
3. [Discovery] Reverse Engineer discovers a false function in the structure of target system
4. [Action] Reverse Engineer studies the structure of the context (nothing found)
5. [Thought] Reverse Engineer attempts to allocate the discovered function to an interface
6. [Thought] Reverse Engineer allocates the interface found in the target system to the false function found in the target system (false allocation)
7. [Thought] Reverse Engineer attempts to formulate a working model
8. [Thought] Reverse Engineer formulates a false working model (as it is based on misleading information)
9. [Action] Reverse Engineer stops

C.III VIRTUAL CASE STUDY III—EVENT TRACE #128 OF 368

This case study falls under Family # 3 as described in Chapter VII: Failure in spite of accurate and complete information. The trace (Figure 67) shows a scenario in which the reverse engineer discovered an interface and the function for which it is responsible; he then failed to connect the one to the other. It seems unlikely that a reverse engineer in possession of all the right information might nevertheless fail to connect the dots. However, one might imagine a scenario in which perhaps the reverse engineer's previous experience plays a role leading to just such a scenario. It is possible for example, that a reverse engineer is quite experienced with the type system to which the target system seems to belong. In his experience he has become familiar with functions similar to F for which the common technical solution (aka interface) is of the Type S_1. When he encounters function F in the target system, he immediately searches for an interface of type S_1. In the process he notes interface S_2 as an interface for something, but overlooks it as a possible repository for function F.

This virtual case study validates the model of reverse engineering (in that being based on the model it does not present an illogical or inconsistent scenario), but it does not validate any of the modes of failure presented in Chapter VI. Instead, the scenario exposes a new mode of failure: Incomplete allocation of system functions due to failure intrinsic in the reverse engineer such as lack of knowledge or imagination (or perhaps adverse influence of existing knowledge as in the scenario presented in the preceding paragraph).

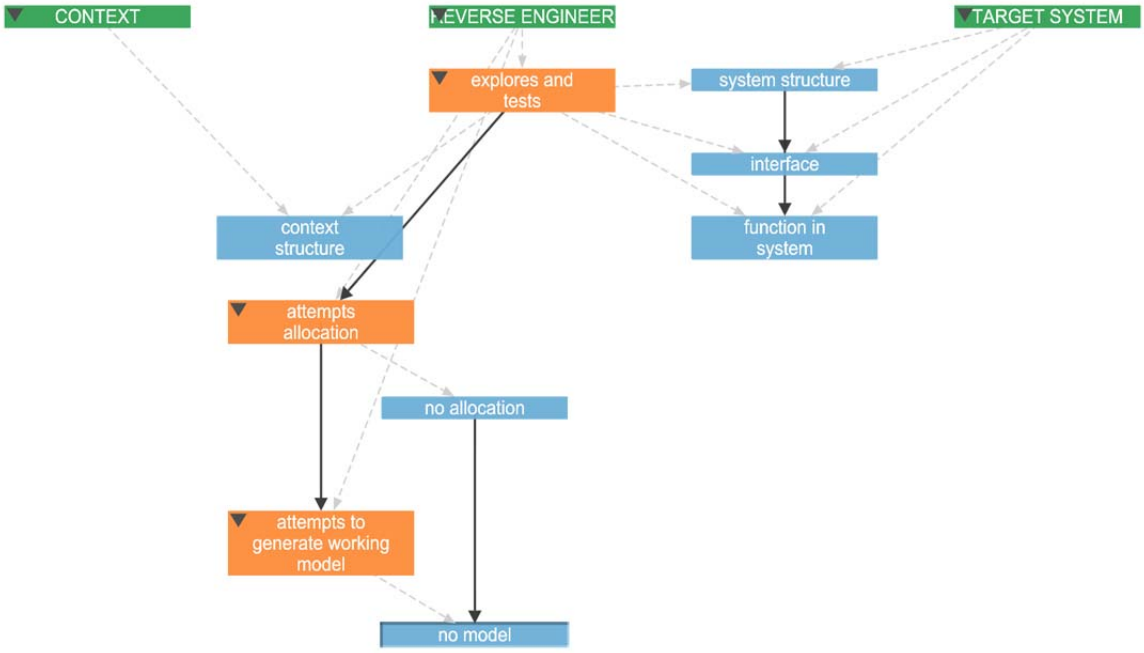

Figure 67. Event trace #128 of 368—Family 3

1. [Action] Reverse Engineer studies the structure of the target system
2. [Discovery] Reverse Engineer discovers an interface in the structure of target system
3. [Discovery] Reverse Engineer discovers a function in the structure of target system
4. [Action] Reverse Engineer studies the structure of the context (nothing found)
5. [Thought] Reverse Engineer attempts to allocate a discovered function an interface (however he is unable to "connect the dots")
6. [Thought] Reverse Engineer attempts to formulate a working model (but there is no allocation about which a working model can be made)
7. [Action] Reverse Engineer stops

C.IV VIRTUAL CASE STUDY IV—EVENT TRACE #211 OF 368

This case study falls under Family # 4 as described in Chapter VII: Failure due to false information in the presence of accurate and complete information. The trace (Figure 68) shows a scenario in which the reverse engineer discovered an interface and a false function (both in the target system); he also discovered a true function (in the context).

He then ignored the true function and went for the red herring false function. The circumstances that may lead to such a scenario can be similar to those described for Virtual Case Study III. The end result is identical to Virtual Case Study II.

This virtual case study validates the model of reverse engineering and introduces a new and nuanced variation to modes of failure #4 and #5 (From Table 3)—"Inaccurate Characterization of System Function." The nuance arises because the accurate characterization is discovered by the reverse engineer, but is subsequently ignored in favor of an inaccurate one. In other words, this case study highlights the potential adverse effects of "red herrings."

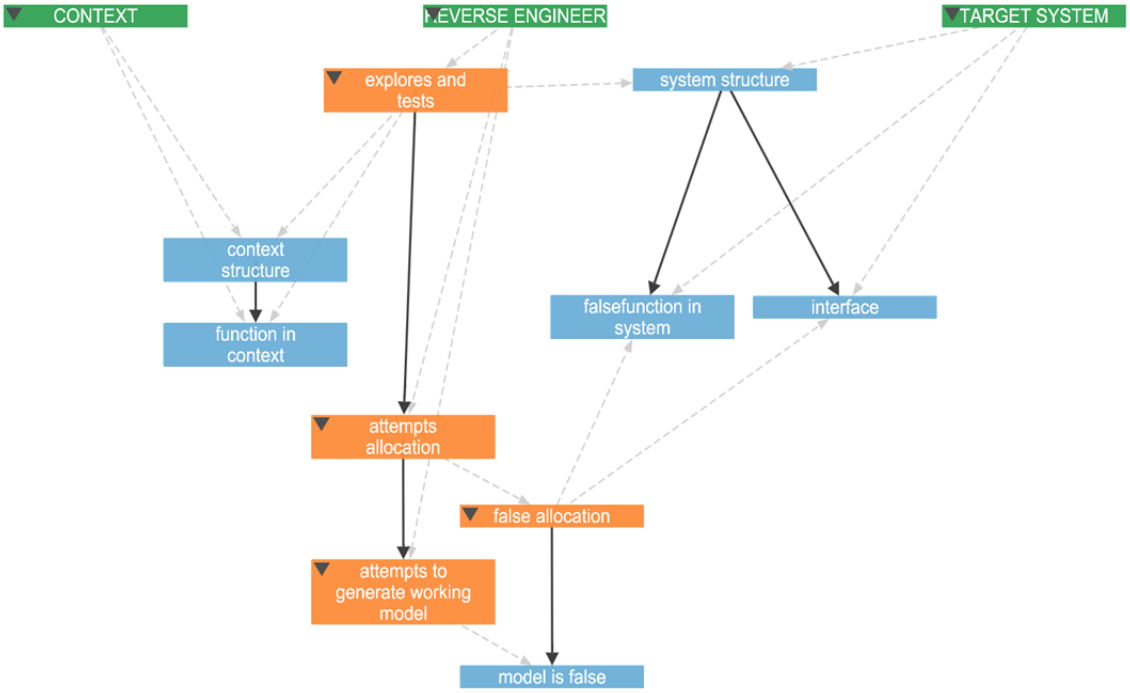

Figure 68. Event trace #211 of 368—Family 4

1. [Action] Reverse Engineer studies the structure of the target system
2. [Discovery] Reverse Engineer discovers a false function in the structure of target system
3. [Discovery] Reverse Engineer discovers an interface in the structure of target system
4. [Action] Reverse Engineer studies the structure of the context

5. [Discovery] Reverse Engineer discovers a function in the structure of context

6. [Thought] Reverse Engineer attempts to allocate a discovered function to an interface

7. [Thought] Reverse Engineer allocates the interface found in the target system to the false function found in the target system (the true function discovered in the context was available to complete a true allocation, but it was ignored)

8. [Thought] Reverse Engineer attempts to formulate a working model

9. [Thought] Reverse Engineer formulates a false working model (as it is based on misleading information)

10. [Action] Reverse Engineer stops

C.V VIRTUAL CASE STUDY V—EVENT TRACE #336 OF 368

This case study falls under Family # 5 as described in Chapter VII: Success

The trace (Figure 69) shows a scenario in which the reverse engineer discovered a function followed by an interface, both in the target system. The reverse engineer then goes on to explore the context and discovers further evidence of the same function. This second discovery is redundant. The reverse engineer's attempt to allocate function to interface is successful (unlike in virtual case study III). The reverse engineer's attempt to formulate a working model based on the true allocation is also successful (suggesting a scenario where the reverse engineer is equipped with adequate knowledge of the applicable laws of physics and engineering principles). The existence of redundant information in the target system and context means that there are several different paths to reach the same goal.

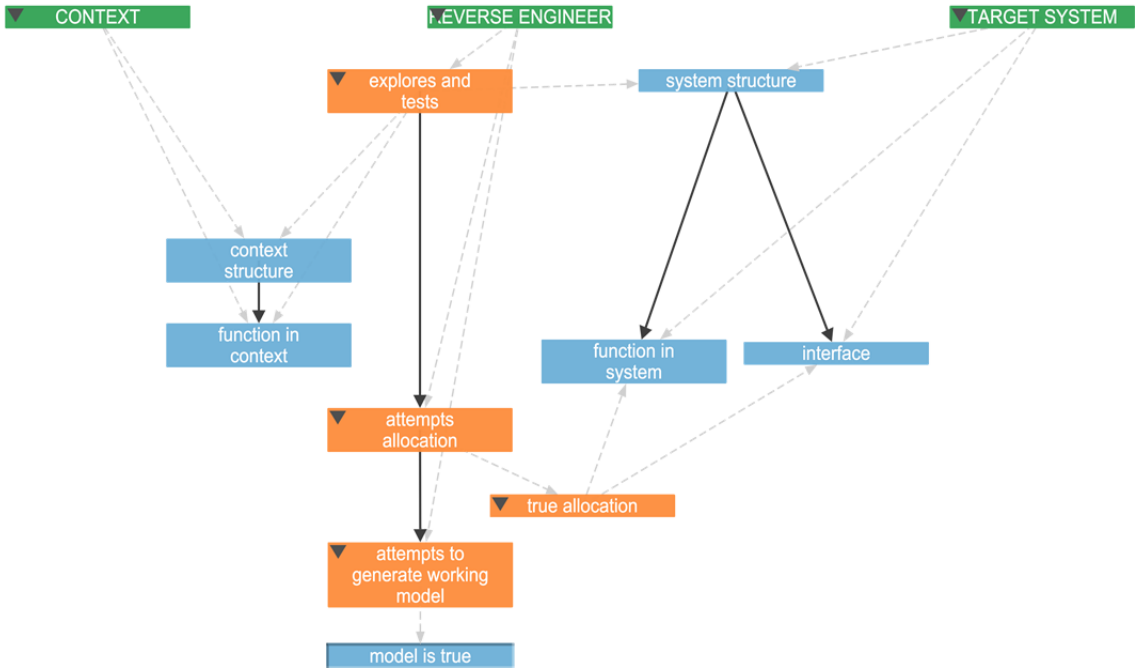

Figure 69. Event trace #336 of 368—Family 5

1. [Action] Reverse Engineer studies the structure of the target system
2. [Discovery] Reverse Engineer discovers a function in the structure of target system
3. [Discovery] Reverse Engineer discovers an interface in the structure of target system
4. [Action] Reverse Engineer studies the structure of the context
5. [Discovery] Reverse Engineer discovers a function in the structure of context
6. [Thought] Reverse Engineer attempts to allocate a discovered function to an interface
7. [Thought] Reverse Engineer allocates the interface found in the target system to the function found in the target system (a "spare" true function discovered in the context was ignored)
8. [Thought] Reverse Engineer attempts to formulate a working model
9. [Thought] Reverse Engineer formulates a true working model
10. [Action] Reverse Engineer stops

APPENDIX D. WHAT ENGINEERS KNOW (A REVIEW OF VINCENTI'S BOOK)

en·gi·neer·ing
noun

> The branch of science and technology concerned with the design, building, and use of engines, machines, and structures.
>
> Oxford English Dictionary

Reverse engineering is unique among other engineering disciplines because, while it begins with "engines, machines and structures" its primary concern is with knowledge. In a sense, it has more in common with Philosophy of Engineering, and Pedagogy of Engineering than with the more traditional engineering-related disciplines.

Engineering knowledge constitutes a large proportion of human knowledge in general. And it is the portion that has the greatest impact on our daily lives. Yet little has been written about engineering knowledge as a subject: What do engineers know?

The study of reverse engineering may well begin with the question: What is engineering knowledge? Then move on to the more specific question: What can engineers learn directly from engines, machines, and structures?

This appendix is an attempt to summarize the answers to those two questions based on a synthesis of the work What Engineers Know and How They Know It (Vincenti 1990). The last column of Table 6 is not part of Vincenti's work but has been added by this author as an educated guess.

Table 6. Types of engineering knowledge

Type of Engineering Knowledge	**Subtype of Engineering Knowledge**	**Example**	**Can It Be Learned Through Reverse Engineering?**
Fundamental Design Concepts	Operational Principle		Yes
	Normal Configurations		Yes
Criteria and Specifications			?
Theoretical Tools	Mathematical Tools	Ranging from pure math (Calculus or Trig) to device-specific or phenomenological theories (like ray theory for lenses, or De Broglie's wave theory for refraction... Models which we know to be wrong... whose value lies in the fact that they work.	?
	Intellectual Concepts	The nouns of engineering like electric current, tensile strength, heat absorption coefficient, and so forth	?
Quantitative Data	Descriptive	Like physical constant (single numbers), or material properties (spectra of values encoded in tables or graphs)	?
	Prescriptive	Safety margins and other kinds of "fudge factors"	?
Practical Considerations		Knowledge of factors that make a design safer, stronger, cheaper, more, well, practical. Like knowing the maximum size a certain component may be built, based on the manufacturing process, or like knowing the ideal size for a telephone screen)... These are derived from actual experience with design... Some practical considerations receive enough attention to be developed and formalized into quantitative data in their own right.	?
Design Instrumentalities	Structured Procedures	For example, doing functional and structural decomposition, hierarchical organization of function, optimization, iteration, integration	?
	Ways of Thinking	"Provide shared ways for apprehending the operation of a device and imagining the effect of alterations in its design" Vincenti p.220—while "ways of thinking" is probably not a form of knowledge transferable through RE, it sounds like a form of knowledge potentially very useful... In fact, a collection of fruitful "ways of thinking" may be precisely what I'm after.... A set of heuristics	?
	Judgment Skills		?

APPENDIX E. A TAXONOMY OF HEURISTIC TYPES

Heuristic: From the Greek *heurisken* (to find) same root as eureka (*I have found it*)

One important outcome of this research has been the exposure of the author—and hopefully of the reader—to the concept of heuristics (in particular, the central role of heuristics in engineering endeavors). Throughout this research, it has also become evident that the single word *heuristic* encompasses a broad spectrum of mental artifacts: from sophisticated techniques, to informal tricks and rules of thumb, from concrete tabular data to vague "approaches" or states of mind. Moreover, the domain of application of the heuristic is also very broad. Almost every author that touches on the subject agrees as to its importance. And almost every author offers a unique, often incompatible, definition of heuristic. This raises the question of whether heuristics can even be approached as a single subject.

This author has not found adequate academic coverage of this question. To some Heuristics are all-important and everywhere—a subject so vast that it even absorbs science and engineering onto itself. To others, it is the subject of intense study—but only in some narrow sense of the concept, and for a particular application. For my part, I have found the subject fascinating and have done my best to synthesize what I learned about the subject in this appendix.

Table 7. Types of heuristics

Type	**Description**	**Applies**	**Exponent**	**A place in Reverse Engineering?**	**Example**
Creative Problem Solving	Suggest courses of action that may uncover a solution—heuristics for "getting unstuck"	Math	Polya	Yes—In reverse engineering the Problem Solver often does not see a way forward either physically (a way to penetrate the system beyond this point) or mentally (a way to understand the system beyond this point).	1
		Design	Sickafus		2

Search	Tell you where (not) to search, when to stop searching for a solution, or how to select from among several answers. Simplify decision making by reducing the amount of information and computation necessary to decide…	Cognitive (aid)	Gigerenzer, Lenat	Yes- These are heuristics that are more powerful but more subtle. The problem solver is generally believed to have no power over them: heuristics ARE how we think (when we think quickly). In Kahneman's view it's a source of systematic error. In Gigerenzer's view, it's powerful practical logic. It is interesting that although these seem like opposite views, they are in fact very similar: Quick thinking/intuition/heuristic is great when we have a dearth of information, and is weakest, when we have an abundance	3
		Cognitive (obstacle)	Tversky and Kahneman		4
Engineering Wisdom	proverbs and "Laws" such as Murphy's laws. Good ones are pithy and whimsical encodings of accumulated experience…. majority do not guide action or search, but rather provide a lens through which the engineer may interpret the context.	Prescribe action	Maier and Rechtin (Chapter 2 and Appendix of The Art…)	Could apply specially if slightly adapted from their use in engineering to use in RE (for example,)	5
		Describe Environment		Hard to find in a simple experiment (a survey of experienced reverse engineers is suggested)a RE version of murphy's Law could read "If something can be designed poorly, it will be designed poorly"	5
Calculation	Tables or equations that incorporate approximations or tabulate empirical information ... for example, material characteristics/parameters. In terms of my research,	Throughout engineering and science	Fisher	Applicable to the Reverse Engineer attempting to extract a different level of information (different from operational principle and standard configuration) Find By...I do not expect to find any as their specificity can emerge only from considerable experience.	6
Everything	Heuristics are—at the fundamental level—the only legitimate type of knowledge as a basis for decision, search, or calculation	Everywhere	Koen	Yes—But too broad an interpretation	7

Examples

1. How to Solve it (Polya 1973) is the most commonly found reference on works pertaining to heuristics. Many authors reviewed for this research (Lenat 1981; Sickafus,

2004; and Wankat, 1992 et al.) have been inspired by Polya. His book contains four principles for problem-solving and several heuristics with examples and applications. The following are three heuristics from Polya's work:

- Solve a related but easier problem first.
- Draw a figure of the problem
- Decompose and recombine the problem

2. Ed Sickafus is a corporate scientist formerly employed by Ford Motor Company Research Laboratory. In Heuristics for Solving Technical Problems (Sickafus, 2004) he describes a systematic approach to invention called USIT (Unified Structured Inventive Thinking).[57] The work presents a method to discover heuristics applicable to technical design and invention using a visual model of the design problem. Sickafus is explicitly concerned with heuristics as a subject. At the end of his work he provides a catalog of heuristics he has found using his method. These are narrower in application than Polya's, but the catalog is more extensive. The following are three examples of heuristics from Sickafus's work:

- Name objects for their generic functions
- Eliminate Unnecessary Objects (from your model of the problem)
- Status quo: for every change considered, consider also not changing it.

3. Fast and Frugal Heuristics that Make Us Smarter (Gigerenzer 1999) makes the case that heuristics are tools of practical rationality. Where impractical rationality (not a term used by Gigerenzer) would require a decision-maker to consider all the pro's and con's and apply to each an appropriate weighing factor before risking a decision between two choices, practical rationality might suggest that we make the decision based on the single factor that seems most important to us at the moment (which is in fact what we do). Gigerenzer argues (citing a number of studies) that practical rationality not only saves time, but also surprisingly often leads to more correct answers. He refers to the tools of practical rationality as "fast and frugal heuristics." *Frugal* is a reference to the

[57] USIT is related to other approaches such as SIT (Systematic Inventive Thinking) and TRIZ (*Teoriya Resheniya Izobretatelskikh Zadatch* or Theory of Inventive Problem Solving).

amount of external information that the decision-maker must consider in order to apply the heuristic. *Fast* is a reference to the amount of computation that must be allocated to processing of the information.

Example: Triage for patients with chest pain "should" take into consideration no less than 19 variables, all of which are considered relevant to a patient's cardiac condition. However, measuring 19 variables on a patient that is writhing with chest pain, and may not survive the next several minutes, fails the practical rationality test.

Therefore an alternate method for triage was developed. It consists of three yes/no questions. At any point, a "yes" answer results in the patient being considered critical and rushed to the operating room. If all answers are "no" the patient is considered not critical. The surprising thing (which Gigerenzer endeavors to explain throughout the remainder of his book) is that the fast and frugal approach is not only faster (thus in this case saving more lives), but it is equally and sometimes even more accurate than the "rational" approach (Gigerenzer 1999).

4. In Thinking Fast and Slow (Kahneman 2011) the Nobel Prize (Economics) winning author presents a contrasting view: heuristics are not always good. They are fast and frugal, but they often result in systematic error (i.e., bias). Example:

> The anchoring heuristic: If asked to estimate an a quantity, a person tends to be influenced by other quantities presented in the question. For example, people asked "Do you think Einstein's IQ was above or below 100?" followed by "What do you think Einstein's IQ was?" Will provide consistently lower answers than people asked "Do you think Einstein's IQ was above or below 200?" followed by "What do you think Einstein's IQ was?"

5. In The Art of Systems Architecting (Maier & Rechtin, 2000, 46–54) the term Heuristic refers to a broad spectrum of engineering wisdom. Heuristic wisdom is characteristically "chunked" and passed down in the form of proverbs or pithy sayings. Interested in new wisdom, the authors suggest the following criteria for screening a good engineering heuristic:

- There must be a strong correlation (if not a clear cause and effect relationship) between the proposed heuristic and the success of the process

- The heuristic should be expressible in terms other than the specific problem it solved… it should be able to be generalized at least slightly
- It should be easily explained (5 minutes or less)
- The opposite of the heuristic should be a clearly absurd or foolish statement
- The underlying principle of the heuristic should be timeless (I do not like this one as it would preclude the discovery of any fundamentally new heuristic)

The authors suggest a taxonomy on the basis of where—along the systems architecting process—a particular heuristic may be applicable. They also broadly divide heuristics into Prescriptive and Descriptive. Finally, they provide a catalog of heuristics (Maier & Rechtin, 2000, 280–291).

6. A majority of engineering heuristics are simple computational shortcuts or aids (rules of thumb, tables of empirically derived values, etc.). This type of heuristic has very specific applications and therefore tends to be powerfully useful within a narrow application. The breadth of these heuristics is as large as engineering itself. *Rules of Thumb for Engineers and Scientists* (Fisher 1991) provides an extensive compilation of such rules of thumb.

7. Koen (1985 and 2003) begins his treatment of heuristics by presenting them as the specialty type of knowledge that engineers (as distinct from both scientists and laymen) must master. However, in both books he soon takes the reader in a different direction: everything we know is a heuristic. In other words, what we perceive as "knowledge" is always pragmatic. Knowledge is not of what is, but of what works. This extends even to knowledge that we apprehend through our senses. Figure 70 provides an excellent example.

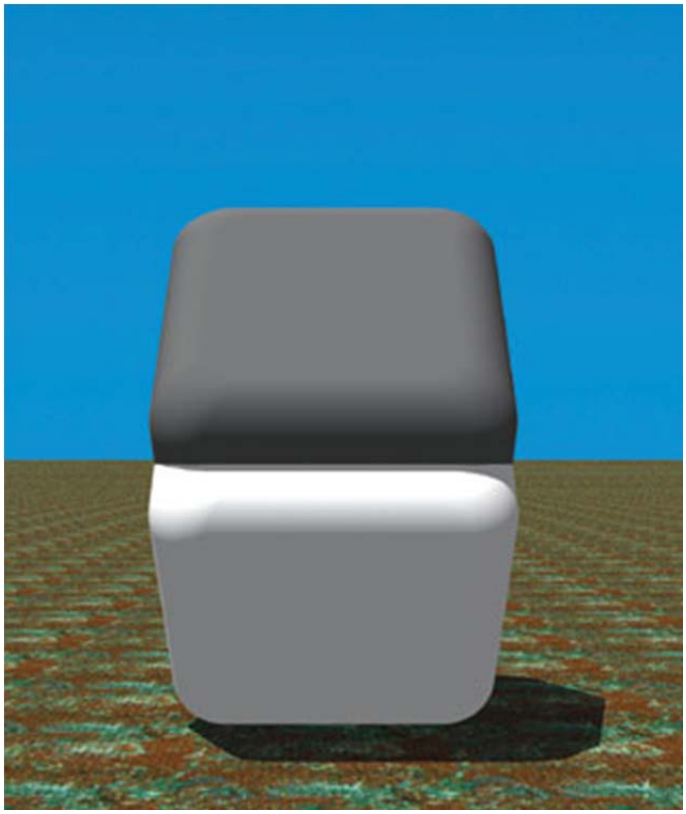

Figure 70. A perception heuristic

The two blocks are perceived as being different colors. They are not (to see this, simply use a pen to cover the boundary between the blocks). This is a heuristic because it derives from experience (loosely speaking, evolution has conferred such experience on our species), because it works, and because it is independent of the question of whether it is true. The heuristic "reads" something like this: H: When shadows are present, use them to adjust your original perceptions of shape and color. Reproduced from (Lotto 2002, 3)

APPENDIX F. THOUGHTS ON FUNCTION, FUNCTIONAL AND NONFUNCTIONAL FEATURES OF SYSTEMS

What is a Function? What is a Purpose? System-Level vs. Component-Level. System-level function or purpose must exist in relation to something external to the system. This can make it easier to ascertain in cases where—for instance—the system's purpose is defined in relation to a human user. For the average person, the purpose of a cell-phone is easier to figure out than the purpose of one of the electronic components inside the cell-phone.

A component, or subassembly need not result from a design process in order to serve a clear purpose in relation to the overall system. For example, there is no controversy in asking: what is the purpose *the spikes that cover the cockroach's legs?* (presumably transmission of vibration, or improved traction, or both)

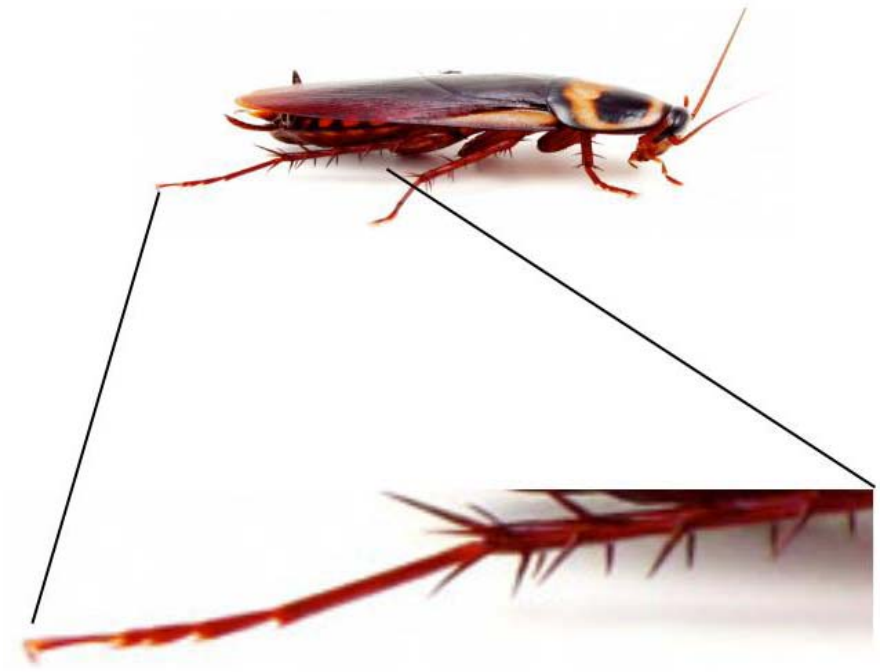

Figure 71. Purpose in design

The figure highlights the fundamental difference between two levels of "purpose": Purpose at the system level and purpose at the component level.

On the other hand the absence of an explicit designer (this point may be contentious) raises some issues when we ask something like *what is the function or purpose of a cockroach?*

A. DIRECT FUNCTIONS

The system-level function of a military fortification is to repel enemy ground forces within certain parameters. Given that function, consider the odd diamond-shaped protuberances in each corner (bastions)—*what is their component-level function? Do they even have a function?* One possible answer is that they are decorative—therefore have no function.

Figure 72. Direct functions

Bastions in a fortification are an example of features supporting a direct function.

This requires clarification. *Being decorative* can be a function. In a church or a museum, this may be the function of many features. But when we ask whether a

component or feature has a function we are asking whether it has a function *in relation to the function of the overall system*. Do bastions help repel the enemy and if so how? They do. In fact, bastions have what may be called a direct function.

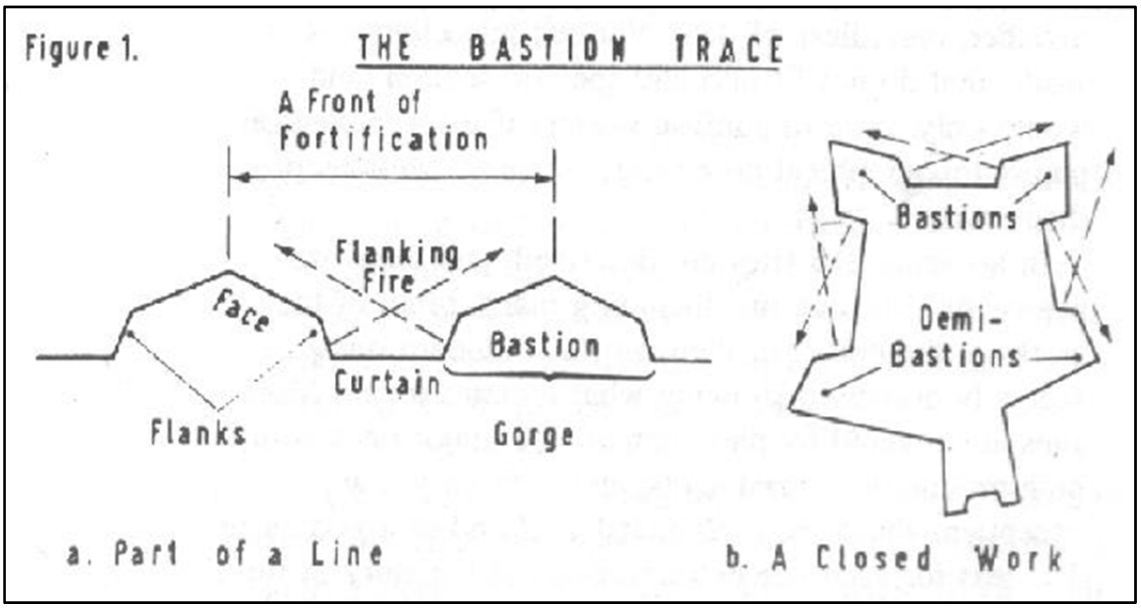

Figure 73. The function of a bastion

The bastion's geometry has a specific function that supports the overall function of the fortification. It allows the placement of guns in such a way that the guns protect the fortification without themselves being exposed.

B. SUPPORT FUNCTIONS

While bastions (ramparts, moats, ravelins, glacis...) have clear function (even if not initially obvious) in relation to the overall system function, other features may have a less clear functions. For example, water and food storage spaces are not obviously about repelling the enemy. Yet without them, the repelling would not succeed for long. A different but related type of function is present in features that are incorporated to aid in disassembly, maintenance, or troubleshooting of a system. This may be called support functions.

C. AFFORDANCE FUNCTIONS

A third type of function stems from the design choice to "help guide the system user" toward the proper use of the system. Figure 74 shows an example: the ring around

the power plug guides the user to the safe placement of his fingers. Design features and components that guide use in the right direction, or help prevent misuse are called affordances (Norman 2002).

Figure 74. Affordance functions

D. ATTRIBUTE FUNCTIONS

We may also speak of a type of function that does not "do anything." This type of function can reside in certain physical attributes of the system. For example, the maximum thickness of the walls serves the function of making the fort buildable in a practical timeframe—otherwise, why not make the walls ten times thicker? 100 times? Likewise why build out of bricks, as opposed to stone or wood. In these cases, the function has more to do with cost and practicality of realizing the system, than with overall system function. But there is nevertheless a reason for a certain material used in a certain thickness of a certain length, and so forth.

E. NON-FUNCTIONS

There may also be features within a system that serve no function. Why discuss these? Because the reverse engineer must be on the lookout for these, as their continued investigation represents wasted resources.

1. Aesthetic Non-functions

Consider the teardrop-shape of the central lawn of the fort pictured in figure 75. It probably serves no function (as discussed earlier, "being decorative" is a non-function in the context of a system designed for a non-aesthetic purpose).

Figure 75. Aesthetic non-functions

2. Skeuomorphic Non-functions

These are features that once had a function but no longer do, yet continue to be incorporated into design after the introduction of materials or context makes the original function obsolete. A common skeuomorphism is shown in figure 76. The feature known

as *dentils* found sometimes in roofs has its origin in wooden rafters, which had a function, the dentils do not.

Figure 76. Skeuomorphic non-functions

3. Manufacturing Defects and Byproducts Non-Function

Sometimes a relatively prominent feature of a system is not there for any system-related reasons (not even aesthetic). Figure 77 shows a manufacturing process that sometimes results in such features. Is this an interface of some sort, or just a manufacturing byproduct?

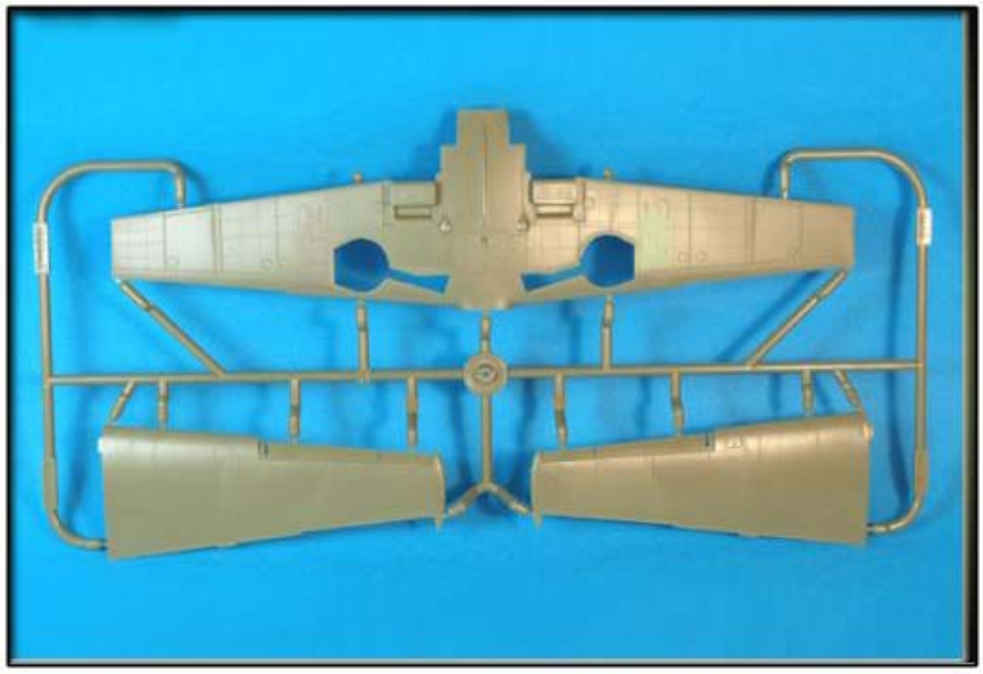

Figure 77. Manufacturing defects and byproducts non-functions

4. Designer Mischief Non-functions

This final category is most prevalent in the realm of software. Here small portions of code are sometimes surreptitiously embedded within larger programs. Figure 78 shows an example.[58] These Easter eggs (as they are commonly referred to) are not purely aesthetic, as they actually perform functions—just not the functions you would expect. They may even be embedded into the overall code in such a way that they cannot be simply removed. And yet, like the other features described in this paragraph, Easter eggs are also a red herring and a potential waste of time and resources for the reverse engineer.

[58] The movie Tron released in 2010 contains this picture of the main character from the original movie released in 1982, it is embedded within the code in the movie's DVD.

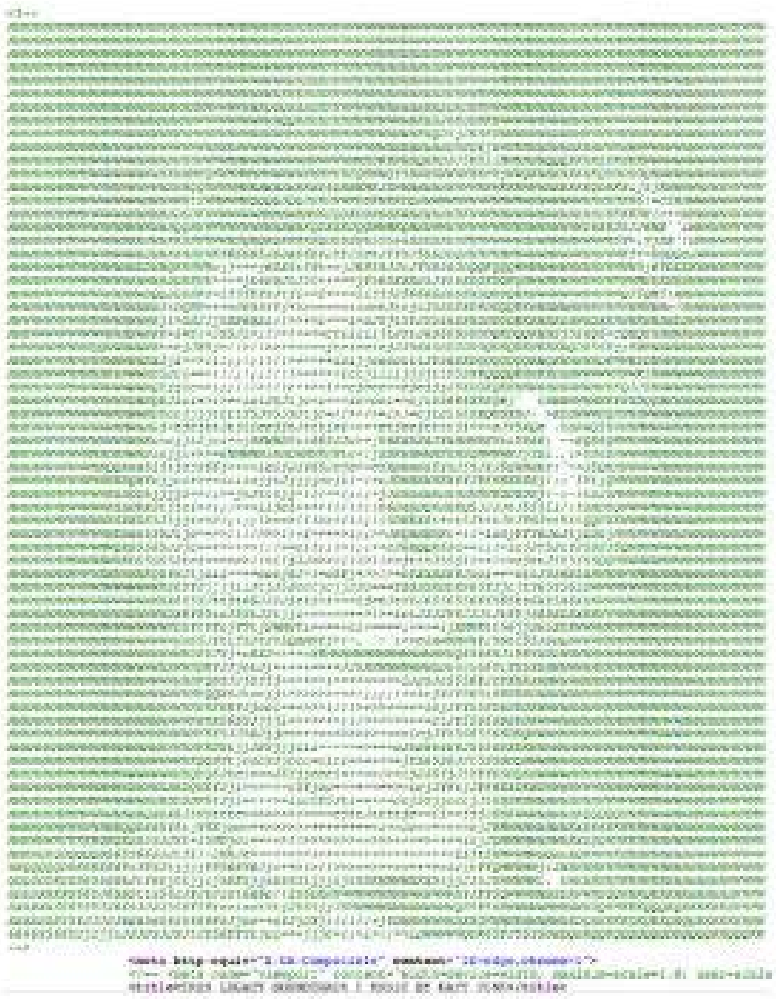

Figure 78. Designer mischief non-functions

LIST OF REFERENCES

Albasini, Marcio D. 2011. "Reverse Engineering the Mechanisms and Dynamical Behavior of Complex Biochemical Pathways." Dissertation, School of Informatics and Computing, Indiana University.

Anderson, Chris. 2012. *Makers: The New Industrial Revolution*. New York, NY: Crown Business.

Arthur, Bryan. 2009. *The Nature of Technology*. New York, NY: Free Press.

Auguston, Mikhail. 2014. "Behavior Models for Software Architecture." Technical report, Naval Postgraduate School.

Axe, David. 2012. "It Won't Be Easy for Iran to Dissect, Copy U.S. Drone." *Wired.* www.wired.com/2011/12/cia-drone-secrets/

Basalla, George. 1988. *The Evolution of Technology*. Cambridge, MA: Cambridge University Press.

Bibanda, Bopaya, S. Motavalli, and K. Harding. 1991. "Reverse Engineering: An Evaluation of Prospective Non-Contact Technologies and Applications in Manufacturing Systems." *Journal of Computer Integrated Manufacturing* 4(3).

Bibanda, Bopaya, and Yasser A. Hosni. 1994. "Reverse Engineering and Its Relevance to Industrial Engineering: A Critical Review." *Journal of Computers and Industrial Engineering* 26(2): 343-348.

Blaze, Matt. 2004. "Safecracking for the Computer Scientist." Technical report, Department of Computer and Information Science, University of Pennsylvania.

Casey, Steven. M. 1998. *Set Phasers on Stun: And Other True Tales of Design, Technology, and Human Error*. Santa Barbara, CA: Aegean Publishing Company.

Chikofsky, Elliot, and James Cross. 1990. "Reverse Engineering and Design Recovery: A Taxonomy." *IEEE Software Journal*, 7(1): 13–17.

Clarke, Arthur. 1984. *Profiles of the Future: An Inquiry Into the Limits of the Possible*. New York, NY: Holt, Reinhart and Winston.

Cste, Marie E., and John C. Doyle. 2007. "Reverse Engineering of Biological Complexity." *Science,* 295(5560): 1664–1669.

DeSolla Price, Derek. 1959. "An Ancient Greek Computer." *Scientific American* 200(6): 60–67.

———. 1984. “Of Sealing Wax and String.” *Natural History* 93(1): 48.

Dewey, John. 1960. *The Quest for Certainty: A Study of the Relation of Knowledge and Action* (Gifford Lectures 1929). London, UK: Allen and Unwin.

Dunbar, Kevin N., Anne L. Fay, and David Klahr. 1993. “Heuristics for Scientific Experimentation: A Developmental Study.” *Cognitive Psychology*. 25(1): 111–146.

Enrong, Pan. 2013. “Object-Oriented Method and the Relationship Between Structure and Function of Technical Artifacts.” In D. Michelfelder Ed. *Philosophy and Engineering: Reflections on Practice, Principles and Process.* New York, NY: Springer.

Ferguson, Eugene S. 1994. *Engineering and the Mind’s Eye*. Cambridge, MA: The MIT Press.

Finance, G. 2010. “SysML Modeling Language Explained.” OMGSysML.com, www.omgsysml.org/SysML_Modelling_Language_explained-finance.pdf

Fisher, David. 1988. *Rules of Thumb for Engineers and Scientists*. Houston, TX: Gulf Publishing Company.

Friendshuh, Luke, and Len Troncale. 2012. “Identifying Fundamental Systems Processes for a General Theory of Systems (GTS),” Proceedings of the 56th Annual Conference, International Society for the Systems Sciences (ISSS), San Jose, CA, July 15–20.

Freeth, Tony. 2008. “The Antikythera Mechanism—Decoding an Ancient Greek Mystery.” Research Project, University of Michigan.

Fuhrman, Stefanie, and Roland Somogyi. 1998. “A General Reverse Engineering Algorithm for Inference of Genetic Network Architectures.” Pacific Symposium on Biocomputing, March18–29.

Giammarco, Kristin, Monica Farah-Stapleton, and Mikhail Auguston. 2014. “Behavioral Modeling of Systems Architectures with Monterey Phoenix.” Presentation, Naval Postgraduate School.

Gigerenzer, Gerd, P. Todd, and the ABC Research Group. 1999. *Simple Heuristics That Make Us Smart*. New York, NY: Oxford University Press.

Gleick, James. 2011. “What Defines a Meme?” *Smithsonian*. www.smithsonianmag.com/arts-culture/what-defines-a-meme-1904778/

Grand, Joe. 2011. “Hardware Reverse Engineering: Access, Analyze, and Defeat.” Presentation. Grand Idea Studio Workshop. Washington, DC, January 20.

Grand, Joe, Ryan Russell, and Kevin D. Mitnick. 2004. *Hardware Hacking: Have Fun While Voiding Your Warranty*. Rockland, MA: Syngress Publishing.

Halsmer, Dominic, Peter W. Odom, Jessica Fitzgerald, and Taylor G. Tryon. 2013. "Implementation and Assessment of a Curricular Module on the History and Philosophy of Reverse Engineering in Biological Systems." ASEE Annual Conference Proceedings, Atlanta, GA, June 23–26.

Halsmer, Dominic, N. Roman, and T. Todd. 2009. "Integrating the Concept of Affordances into Function-based Reverse Engineering with Application to Complex Natural Systems." ASEE Annual Conference Proceedings, Austin, TX, February 3–5.

Harney, Robert. 2015. "Lecture Notes on Combat Systems Engineering." Unpublished document.

Hause, Mathew C. 2006. "The SysML Modeling Language." Fifth European Systems Engineering Conference, Edinburgh, UK, September 18–20.

Hempel, Carl G. and Paul Oppenheim. 1948. "Studies in the Logic of Explanation." *Philosophy of Science* 15(2): 135–175.

Hess, Harry L. 2000. "Teaching Manufacturing Using the Golden Key—Reverse Engineering." ASEE Annual Conference Proceedings, St Louis, MO, June 18–21.

Hoffman, Carl. 2006. "The Teardown Artists." *Wired*. www.wired.com/2006/02/teardown/

Hunkin, Tim. 1994. "Illegal Engineering." Tim Hunkin. www.timhunkin.com/94_illegal_engineering.htm

Ingle, Katheryn A. 1994. *Reverse Engineering*. Chicago, IL: McGraw-Hill Inc.

Jones, Alexander. 2012. "The Antikythera Mechanism and the Public Face of Greek Science." Presented at Conference: From Antikythera to the Square Kilometer Array—Lessons from the Ancients, Kerastari, Greece, June 11–16.

Haskins, C. 2006. *Systems Engineering Handbook—A Guide for Systems Life Cycle Processes and Activities*. Las Vegas, NV: INCOSE and Wiley.

Jahan, Kauser and Ralph A. Dusseau. 1999. "Reverse Engineering of Water Filters." ASEE Annual Conference Proceedings, Charlotte, NC, June 20–23.

James, Dick. 2009. "Design-for-Manufacture Features in Nanometer Logic Processes—A Reverse Engineering Perspective." IEEE Custom Integrated Circuits Conference, San Jose, CA, September 13–16.

Jenkins, Reese V. 1984. “Elements of Style: Continuities in Edison’s Thinking.” *Annals of the New York Academy of Sciences*. Volume 424, Bridge to the Future: A Centennial Celebration of the Brooklyn Bridge pages 149–162.

Johnson, Bobbie. 2010. “Shanzai!” *Wired UK*. www.wired.co.uk/magazine/archive/2011/01/features/shanzai

Johnson, Scott D. and Shih-Ping Chung. 1999. “The Effect of Thinking Aloud Pair Problem Solving (TAPPS) on the Troubleshooting Ability of Aviation Technician Students.” *Journal of Industrial Teacher Education*. 37(1): 7-25.

Kelly, Kevin. 2010 *What Technology Wants*. London, UK: Penguin Books.

Kahneman, Daniel. 2011. *Thinking, Fast and Slow*. New York, NY: Farrar, Straus and Giroux.

Kim, Linsu. 1997. *Imitation to Innovation: The Dynamics of Korea’s Technological Learning*. Boston, MA: Harvard Business Press.

Kline, Stephen J. 1985. “Innovation Is Not a Linear Process.” *Research Management* 28(4): 36–45.

Koen, Bill V. 1985. *Definition of the Engineering Method*. Washington, DC: American Society for Engineering Education.

———. 2003. *Discussion of the Method: Conducting the Engineer’s Approach to Problem Solving*. New York, NY: Oxford University Press.

Kroes, Peter. 1998. “Technological Explanations: The Relation Between Structure and Function in Technological Objects.” *Society for Philosophy and Technology* 3(3):18-34.

———. 2002. “Design Methodology and the Nature of Technical Artifacts.” *Design Studies* 23(3):287-301.

Kurzweil, Ray. 2006. *The Singularity is Near: When Humans Transcend Biology*. New York, NY: Penguin Books.

Lazebnik, Yuri. 2004. “Can a Biologist Fix a Radio?—Or, What I Learned While Studying Apoptosis.” *Biochemistry* (Moscow) 69(12): 1403-1406.

Lee, Hakjin, Hyunsang Youn, and Eunseok Lee. 2007. “Automatic Detection of Design Pattern for Reverse Engineering.” 5th International Conference on Software Engineering Research, Management, and Applications, Busan, South Korea, August 20–22.

Lefever, Douglas D., and Kristin L. Wood. 1996. “Design for Assembly Techniques in Reverse Engineering and Redesign.” ASM Design Theory Methodology Conference, Montreal, Canada, August 15–18.

Lenat, Douglas B. 1981. “The Nature of Heuristics.” Research Paper. Palo Alto Research Center—Cognitive and Instructional Sciences Group.

Little, Aaron D., and Kristin L. Wood. 1997. “Functional Analysis: A Fundamental Empirical Study for Reverse Engineering, Benchmarking, and Redesign.” Proceedings of the Design Engineering Technical Conference, Sacramento, CA, September 17–20.

Lochhead, Jack, and Arthur Whimbey. 1987. “Teaching Analytical Reasoning Through Thinking Aloud Pair Problem-Solving.” In J. E. Stice, Ed. *Developing Critical Thinking and Problem-Solving Abilities—New Directions For Teaching and Learning*. San Francisco, CA: Jossey-Bass.

Long, David, and Zane Scott. 2011. “A Primer for Model Based Systems Engineering.” Vitech.

Lotto, R. Beau, Dale Purves, and Surajit Nundy. 2002. “Why We See What We Do: A Probabilistic Strategy Based on Past Experience Explains the Remarkable Difference Between What We See and Physical Reality.” *American Scientist.*

Maier, Mark W., and Eberhardt Rechtin. 2000. *The Art of Systems Architecting*. 2nd Edition. Boca Raton, FL: CRC Press.

Marchant, Jo. 2010. *Decoding the Heavens*. Cambridge, MA: Da Capo Press.

Martinez, Sylvia L. 2013. *Invent To Learn: Making, Tinkering, and Engineering in the Classroom*. Torrance, CA: Constructing Modern Knowledge Press.

Messler, Robert. 2013. *Reverse Engineering: Mechanisms, Structures, Systems and Materials*. New York, NY: McGraw-Hill Professional.

Michelfelder, Daniel P., Natasha McCarthy, and David E. Goldberg. 2013. *Philosophy and Engineering: Reflections on Practice, Principles and Process*. New York, NY: Springer.

Mitcham, Carl. 1994. *Thinking Through Technology—The Path between Engineering and Philosophy*. Chicago, IL: University Of Chicago Press.

Miyake, Naomi. 1986. “Constructive Interaction and the Iterative Process of Understanding.” *Journal of Cognitive Science* 10(2):151–177.

Morison, Elting. 1966. *Men, Machines, and Modern Times*. Cambridge, MA: MIT Press.

Newberry, Byron. 2013. “Engineered Artifacts.” In *Philosophy and Engineering: Reflections on Practice, Principles and Process*. Chapter 13. New York, NY: Springer.

Niosi, Jorge E. 2012. Innovation and Development Through Imitation (In Praise of Imitation). Presented to the meeting of the International Schumpeter Society, Brisbane, Australia, July 2–5.

Norman, Donald. 2002. *Design of Everyday Things*. New York, NY: Basic Books.

O’brien, Shannon. and Patrick Abulencia. 2010. “Learning Through Reverse Engineering.” ASEE Annual Conference Proceedings, Louisville, KY, June 20–23.

Ogot, Madara M. 2006. “Developing A Framework For Disassemble/Assemble/Analyze (DAA) Activities In Engineering Education.” ASEE Annual Conference Proceedings, Chicago, IL, June 18–21.

Otto, Kevin and Kristin L. Wood. 2000. *Product Design: Techniques in Reverse Engineering and New Product Development*. Upper Saddle River, NJ: Prentice Hall.

Penev, Kiril D. 1996. “Design of Disassembly Systems: A Systematic Approach.” Dissertation, Eindhoven University of Technology.

Plants, H., R. Dean, J. Sears, and W. Venable. 1980. “A Taxonomy of Problem-Solving Activities and Its Implications for Teaching.” In Lubkin, J. L., Ed. *The Teaching of Elementary Problem Solving in Engineering and Related Fields*. Chapter 3. American Society for Engineering Education, Washington, DC.

Polanyi, Michael. 1958. *Personal Knowledge—Towards a Post-critical Philosophy*. New York, NY: Harper.

Polya, George. 1973. *How to Solve It—A New Aspect of Mathematical Method*. Princeton, NJ: Princeton University Press.

Rad, Hamid. 2012. “Reverse Engineering as a Learning Tool in Design Process.” ASEE Annual Conference Proceedings, Louisville, KY, June 20-23.

Raja, Vinesh. 2008. *Reverse Engineering: An Industrial Perspective*. London, UK: Springer.

Ramirez, Borja. 1997. “Redesign Supported by Data Models with Particular Reference to Reverse Engineering.” Dissertation, Loughborough University.

Rekoff, Michael G. 1985. “On Reverse Engineering.” *IEEE Transactions on Systems, Man, and Cybernetics* 15(2):244–252.

Reynolds, Glenn. 2014. "Should We Be Afraid of the 3D Printed Gun?" *Popular Mechanics*. www.popularmechanics.com/technology/gadgets/a12935/ should-we-be-afraid-of-the-3d-printed-gun-16700086/

Ridder, Jeroen de. 2007. "Reconstructing Design, Explaining Artifacts: Philosophical Reflections on the Design and Explanation of Technical Artifacts." Delft University of Technology.

Rosen, William. 2010. *The Most Powerful Idea in the World*. New York, NY: Random House.

Rosenblueth, Arturo, Norbert Wiener, and Julian Bigelow. 1943. "Behavior, Purpose and Teleology." *Philosophy of Science* 10: 18–24.

Sagan, Carl. 1989. "Why We Need to Understand Science." *Parade*, September 10.

Shelley, Cameron. 1996. "Visual Abductive Reasoning in Archaeology." *Philosophy of Science* 63(2): 278–301.

Shenkar, Oded. 2010. *Copycats: How Smart Companies Use Imitation to Gain a Strategic Edge*. Boston, MA: Harvard Business Press.

Shooter, Steven B. 2008. "Reverse Engineering to Design Forward: An Introduction to an Engineering Experiential Module with Video Podcasts." ASEE Annual Conference Proceedings, Pittsburgh, PA, June 22–25.

Sheppard, Sheri D. 1992. "Mechanical Dissection: An Experience In How Things Work." Academic Proposal Paper, Department of Mechanical Engineering Design Division, Stanford University.

Shipman, Matt. 2013. "Iron Man, Reverse Engineering and the Future of Materials Science." NC State University. https://news.ncsu.edu/2013/04/iron-man-science/

Sickafus, Edward. 2005. "Heuristics for Solving Technical Problems—Theory, Derivation, Application." TRIZ-Journal.com. http://www.triz-journal.com/heuristics-solving-technical-problems-theory-derivation-application/

Siegele, Ludwig. 2014. "A Cambrian Moment." Special Report. *Economist*.

Simon, Herbert. 1996. *Sciences of the Artificial*. Cambridge, MA: MIT Press.

Sirinterlikci Arif and John Mativo. 2010. "Teaching Reverse Engineering For Non-Industrial Applications." ASEE Annual Conference Proceedings, Louisville, KY, June 20–23.

Siuru, Bill. 1990. "From Scrap Heap to Showroom." *Journal of the American Society of Mechanical Engineers* 112(11): 66–68.

Slocum, Jerry. 2001. “The Art of the Puzzle: Astounding and Confounding.” Brochure. Katonah Museum of Art.

Stratton, Roy, and Darrell Mann. 2000. “The Theory of Inventive Problem Solving and Systematic Innovation—A Missing Link in Engineering Education.” *Theory of Inventive Problem Solving.*

Tamarez, Frank. 2007. “A Reverse Engineering Process for Mechanical Engineering Systems.” Dissertation, Department of Mechanical Engineering, Kate Gleason College of Engineering.

Torrance, Randy, and Dick James. 2011. “The State of the Art in IC Reverse Engineering.” 48th ACM/EDAC/IEEE Design Automation Conference, San Diego, CA, June 5–9.

Treisman, Anne. 1986. “Features and Objects in Visual Processing.” *Scientific American.*

Tufte, Edward. 1997. *Visual Explanations—Images and Quantities, Evidence and Narrative*. Cheshire, CT: Graphics Press.

Tversky, Amos and Daniel Kahneman. 1974. “Judgment Under Uncertainty: Heuristics and Biases.” *Science*. 185(4157): 1124–1131.

U.S. Army. 1987. *Reverse Engineering Handbook*. Redstone, AL: Department of Defense.

U.S. Supreme Court. 1974. “Kewanee Oil Co. v. Bicron Corp.” 416 U.S. 470.

U.S. Supreme Court. 1989. “Bonito Boats v. Thunder Craft Boats Inc.” 489 U.S. 141.

Vaesen, Krist. 2011. “The Functional Bias of The Dual Nature of Technical Artifacts Program.” *Studies in History and Philosophy of Science* 42(1): 190–197.

Van Cleave, Michelle. 2013. “Espionage Threats at Federal Laboratories: Balancing Scientific Cooperation While Protecting Critical Information.” Statement before the House Committee on Science, Space, and Technology.

Villaverde, Alejandro F., and Julio Banga. 2013. “Reverse Engineering and Identification in Systems Biology: Strategies, Perspectives and Challenges.” *Journal of the Royal Society* 11(91).

Vincenti, Walter G. 1993. *What Engineers Know and How They Know It*. Baltimore, MD: Johns Hopkins University Press.

Wang, Wego. 2011. *Reverse Engineering—Technology of Reinvention*. Boca Raton, FL: CRC Press.

Wang, Wen-Xu, and Jie Ren. 2012. “Reverse Engineering of Complex Dynamical Networks in the Presence of Time Delayed Interactions Based on Noisy Time Series.” *Chaos* 22(3).

Wankat, Phillip C., and Frank S. Oreovicz. 1992. *Teaching Engineering*. New York, NY: McGraw-Hill Inc.

Weilkiens, Tim. 2007. *Systems Engineering with SysML/UML—Modeling, Analysis, Design*. Boston, MA: Morgan Kaufmann OMG Press.

Wymore, Wayne A. 1993. *Model Based Systems Engineering*. Boca Raton, FL: CRC Press.

Zhou, Kevin Z. 2006. “Innovation, Imitation, and New Product Performance: The Case of China.” *Industrial Marketing Management* 35(3): 394-402.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California